

Technical Reference Manual

Release 11i

CAUTION!

This document contains proprietary, confidential information that is the exclusive property of Oracle Corporation.

See back cover for important information.

**KEEP
UNDER
LOCK &
KEY**

Oracle[®] Activity Management Gateway Technical Reference Manual

RELEASE 11i
June 2001

ORACLE[®]

To order this book, ask for Part No. A83510-02

Copyright © 1997, 2000, 2001. Oracle Corporation. All rights reserved.

Major Contributors: Christopher Andrews, Janet Buchbinder, Laura Garrison, Guriqpal S. Gill, Ramesh Krishnamurthy, Tom Phillips, Jeff White

Contributors: Sakhthivel Balasubramanian, Peter Budelov, Danielle Hafdal, Kalika Kekkar, Cedric Ng

This Technical Reference Manual (TRM) in any form, software or printed matter, contains proprietary information of Oracle Corporation; it is provided under an Oracle Corporation agreement containing restrictions on use and disclosure and is also protected by copyright, patent, and other intellectual property law. Restrictions applicable to this TRM include, but are not limited to: (a) exercising either the same degree of care to safeguard the confidentiality of this TRM as you exercise to safeguard the confidentiality of your own most important Confidential Information or a reasonable degree of care, whichever is greater; (b) maintaining agreements with your employees and agents that protect the Confidential Information of third parties such as Oracle Corporation and instructing such employees and agents of these requirements for this TRM; (c) restricting disclosure of this TRM to those of your employees who have a "need to know" consistent with the purposes for which this TRM was disclosed to you; (d) maintaining this TRM at all times on your premises; (e) not removing or destroying any proprietary or confidential legends or markings placed upon this TRM in whatever form, software or printed matter; and (f) not reproducing or transmitting this TRM in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation. You should not use this TRM in any form, software or printed matter, to create software that performs the same or similar functions as any Oracle Corporation products.

The information in this TRM is subject to change without notice. If you find any problems in the TRM in any form, software or printed matter, please report them to us in writing. Oracle Corporation does not warrant that this TRM is error-free. This TRM is provided to customer "as-is" with no warranty of any kind. This TRM does not constitute Documentation as that term is defined in Oracle's agreements.

Reverse engineering of the Programs (software and documentation) associated with this TRM are prohibited. The Programs associated with this TRM are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back-up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

Restricted Rights Legend

This TRM and the Programs associated with this TRM delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the TRM and the Programs associated with this TRM shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, this TRM and the Programs associated with this TRM delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the TRM and the Programs associated with this TRM shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle is a registered trademark, and CASE*Exchange, DDE Manager, Hyper*SQL, NLS*Workbench, Oracle7, Oracle8, Oracle Access, Oracle Alert, Oracle Alliance, Oracle Application Object Library, Oracle Discoverer, Oracle Financials, Oracle Quality, Oracle Work in Process, Oracle Workflow, PL/SQL, Pro*Ada, Pro*C, Pro*COBOL, Pro*FORTRAN, Pro*Pascal, Pro*PL/I, SmartClient, SQL*Connect, SQL*Forms, SQL*Loader, SQL*Menu, SQL*Net, SQL*Plus, SQL*Report, Oracle Web Employees, and SQL*TextRetrieval are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.



CAUTION

This Technical Reference Manual in any form -- software or printed matter -- contains proprietary, confidential information that is the exclusive property of Oracle Corporation. If you do not have a valid contract with Oracle for the use of this Technical Reference Manual or have not signed a non-disclosure agreement with Oracle covering this Technical Reference Manual, then you received this document in an unauthorized manner and are not legally entitled to possess or read it.

Use, duplication, and disclosure are subject to restrictions stated in your contract with Oracle Corporation.



Contents

- Chapter 1** **Introduction** **1 – 1**

- Chapter 2** **High-Level Design** **2 – 1**
 - Overview of the Activity Management Gateway 2 – 2
 - Integrating Your External System with Oracle Projects 2 – 5
 - Security Requirements 2 – 8
 - Handling Error Messages 2 – 11
 - Standard API Parameters 2 – 13
 - Common APIs 2 – 18

- Chapter 3** **How to Use Activity Management Gateway APIs** **3 – 1**
 - Using Project APIs 3 – 2
 - Using Budget APIs 3 – 10
 - Using Agreement and Funding APIs 3 – 20

- Chapter 4** **Detailed Design -- Overview** **4 – 1**

- Chapter 5** **Detailed Design -- Project APIs** **5 – 1**
 - Project APIs 5 – 2
 - View Definitions 5 – 4
 - Procedure Definitions 5 – 22

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Chapter 6	Detailed Design -- Budget APIs	6 - 1
	Budget APIs	6 - 2
	View Definitions	6 - 4
	Procedure Definitions	6 - 9
Chapter 7	Detailed Design -- Resource APIs	7 - 1
	Resource APIs	7 - 2
	View Definitions	7 - 4
	Procedures	7 - 15
Chapter 8	Detailed Design -- Status APIs	8 - 1
	Status APIs	8 - 2
	View Definitions	8 - 4
	Procedure Definitions	8 - 29
Chapter 9	Detailed Design -- Agreement and Funding APIs	9 - 1
	Agreement and Funding APIs	9 - 2
	View Definitions	9 - 4
	Procedure Definitions	9 - 6
Appendix A	Creating a Project Using Load-Execute-Fetch APIs	A - 1
Appendix B	Creating a Budget Using Load-Execute-Fetch APIs	B - 1
Appendix C	Creating a Budget Using a Composite Datatype API	C - 1
Appendix D	Creating an Agreement Using the Load-Execute-Fetch APIs	D - 1
Appendix E	Creating an Agreement Using a Composite Datatype API	E - 1
Appendix F	Controlling Actions in Oracle Projects	F - 1
Appendix G	Project and Task Date Client Extension API	G - 1

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Appendix H	Alphabetical List of APIs	H - 1
	Alphabetical List of APIs	H - 2
	Alphabetical List of Views	H - 8
Appendix I	List of AMG Messages	I - 1
	Index	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Introduction

The *Oracle Activity Management Gateway Technical Reference Manual* provides you with the information you need to understand the structure and processing of Application Programming Interfaces (APIs) that make up the Activity Management Gateway (AMG) for Oracle Projects.

You can use the APIs described in this book to integrate your system (for example, a project management system) with Oracle Projects.

This chapter explains how the manual is organized and provides you with tips on using the manual effectively.



Attention: This technical reference manual (TRM) describes the Activity Management Gateway APIs available with Oracle Projects Release 11.5.4, which includes Oracle Project Costing and Oracle Project Billing.

Do not use this TRM in conjunction with previous versions of Oracle Projects or Oracle Project Accounting.

Introduction

At Oracle, we design and build applications using Designer/2000, our systems design technology that provides a complete environment to support developers through all stages of a system's life cycle. Because we use a repository-based design toolset, we have available to us online all of the information about the underlying structure and processing of our applications. Using Designer/2000, we can present this information to you in the form of a technical reference manual.

This Technical Reference Manual for the Oracle AMG contains all the information you need about the APIs for the product. As we design and build new releases of Oracle Projects and these APIs, we update our Designer/2000 repository to reflect the latest enhancements. As a result, we can always provide you with an *Oracle Activity Management Gateway Technical Reference Manual* that contains the latest technical information.

Audience

The *Oracle Activity Management Gateway Technical Reference Manual* provides information for:

- Technical end users
- Consultants
- Systems analysts
- System administrators
- Other MIS professionals

This manual assumes that you have a basic understanding of structured analysis and design, and of relational databases. It also assumes that you are familiar with Oracle Application Object Library and Oracle Projects. If you are not familiar with these products, we suggest that you attend one or more of the training classes available through Oracle Education Services (see Other Information Sources: page 1 – 6) or Oracle University.

Note: You can see an updated version of the view definitions in Oracle's electronic Technical Reference Manual web page.

How This Manual is Organized

This manual contains the following major sections:

Chapter 2: High-Level Design

Describes the function and use of the AMG, implementation steps, security requirements, standard API parameters, and common APIs that you can use in all modules

Chapter 3: How to Use the Activity Management Gateway APIs

Provides details for using the APIs to perform various complex tasks

Chapter 4: Detailed Design – Overview

Introduces the view and procedure categories that are described in detail in the remaining chapters of this manual

Chapter 5: Detailed Design – Project APIs

Describes the project APIs, including information about each project API view and procedure

Chapter 6: Detailed Design – Budget APIs

Describes the budget APIs, including information about each budget API view and procedure

Chapter 7: Detailed Design – Resource APIs

Describes the resource APIs, including information about each resource API view and procedure

Chapter 8: Detailed Design – Status APIs

Describes the status APIs, including information about each status API view and procedure

Chapter 9: Detailed Design – Agreement and Funding APIs

Describes the agreement and funding APIs, including information about each agreement and funding API view and procedure

Appendix A: Creating a Project Using Load–Execute–Fetch APIs

Provides sample PL/SQL code for creating a project

Appendix B: Creating a Budget Using Load–Execute–Fetch APIs

Provides sample PL/SQL code for creating a budget

Appendix C: Creating a Budget Using a Composite Datatype API

Provides sample PL/SQL code to use as a guide for writing a script to create a budget using CREATE_DRAFT_BUDGET, which uses composite datatypes

Appendix D: Controlling Actions in Oracle Projects

Lists the actions that you can prevent Oracle Projects users from performing on projects that originated in external systems

Appendix E: Project and Task Date Client Extension API

Describes an API that substitutes dates used by external systems for the standard Oracle Projects project, task start, and task completion dates

Appendix F: Alphabetical List of APIs

Lists a page reference and a brief description for each API used in the Activity Management Gateway

Appendix G: AMG Messages

Lists the messages used in the Activity Management Gateway. Includes message code, length, description, and tokens.

How to Use This Manual

This manual provides information about the underlying structure and processing of the Activity Management Gateway APIs. Use this

manual and the AMG APIs when you need to create an interface between Oracle Projects and another system. For example, you can:

- Share project, budget, resource, actuals, and progress information while enforcing business rules, permission, security, and workflow across system boundaries
- Migrate data from a legacy system to Oracle Projects
- Download data (for example, valid class categories and class codes) from Oracle Projects so that you can classify the data within your external system

You need not read this manual cover to cover. Use the table of contents and index to locate the information you need.

How Not To Use This Manual

Do not use this manual to plan modifications

Do not modify the Activity Management Gateway APIs. Modifying the APIs limits your ability to upgrade to future releases and interferes with our ability to give you high-quality support.

We have constructed the Activity Management Gateway APIs to fit your needs with minimal programming, and you can use them to integrate your existing systems with Oracle Projects through interface tables. However, should you require program modifications, contact our support team (see Other Information Sources: page 1 – 6). They can put you in touch with the Oracle Consulting Group, experienced applications professionals who can make the modifications you need while ensuring upward compatibility with future product releases.

Do not write data into non-interface tables

Oracle reserves the right to change the structure of Oracle Applications tables, and to change the meaning of, add, or delete lookup codes and data in future releases. Do not write data directly into tables that are not identified as interface tables or change data in non-interface tables using SQL*Plus or other programming tools. If you do, you may corrupt your database and interfere with our ability to support you.

Moreover, this version of the *Oracle Activity Management Gateway Technical Reference Manual* does not contain complete information about the dependencies between Oracle Projects applications tables.

Therefore, you should write data into only those tables we identify as

Oracle Proprietary, Confidential Information—Use Restricted by Contract

interface tables. If you write data into non-interface tables, you risk violating your data integrity since you might not fulfill all the data dependencies in your Oracle Projects application.

Do not rely on upward compatibility of the data model

As stated above, Oracle reserves the right to change the structure of Oracle Projects tables, lookup codes, and other data in future releases. We do not guarantee the upward compatibility of the Oracle Projects data model. For example, if you write a report that identifies concurrent requests that end in error status by selecting directly from Oracle Application Object Library tables, we do not guarantee that your report will work properly after an upgrade.

About Oracle Application Object Library

This manual may contain references to tables and modules that belong to Oracle Application Object Library, a collection of application components and facilities for building extensions to Oracle Applications. The Oracle Application Object Library contains the shared components -- forms, subroutines, concurrent programs, and tables -- we used to build Oracle Applications.

Oracle does not support *any* customization of Oracle Application Object Library tables or modules, not even by Oracle consultants. Accordingly, this manual does not contain detailed information about most Oracle Application Object Library tables and modules used by the Activity Management Gateway APIs.

Other Information Sources

You can use additional information sources, including other documentation, training, and support services, to increase your knowledge and understanding of Designer/2000, Oracle Application Object Library, and Oracle Projects.

Most Oracle Applications documentation is available in Adobe Acrobat format on the *Oracle Applications Documentation Library* CD. We supply this CD with every Oracle Applications software shipment.

If this manual refers you to other Oracle Applications documentation, use only the Release 11.5.4 versions of those manuals unless we specify otherwise.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Oracle Projects User Guide

Organized by function and task, the user guide provides you with all the information you need to use your Oracle Projects application.

Oracle Projects Technical Reference Manual

The *Oracle Projects Technical Reference Manual* contains database diagrams and a detailed description of Oracle Projects and related applications database tables, forms, reports, and programs. This information helps you convert data from your existing applications, integrate Oracle Projects with non-Oracle applications, and write custom reports for Oracle Projects.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Applications Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards*. It also provides information to help you build your custom Developer/2000 forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards

This manual contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms 6.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup and reference information for the Oracle Projects implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Oracle Applications System Administrator's Guide

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

Oracle Applications Product Update Notes

This book contains a summary of each new feature we added since Release 11, as well as information about database changes and seed data changes that may affect your operations or any custom reports you have written. If you are upgrading from Release 10.7 you also need to read *Oracle Applications Product Update Notes Release 11*.

Oracle Applications Installation Release Notes

This manual contains a road map to the components of the release, including instructions about where to access the Release 11i documentation set.

Oracle Applications Concepts

Designed to be the first book the user reads to prepare for an installation of Oracle Applications. It explains the technology stack, architecture, features and terminology for Oracle Applications Release 11i. This book also introduces the concepts behind and major uses of Applications-wide features such as MRC, BIS, languages and character sets (NLS, MLS), BIS, Self-Service Web Applications and so on.

Installing Oracle Applications

Describes the One-Hour Install process, the method by which Release 11i will be installed. This manual includes all how-to steps, screen shots and information about Applications-wide post-install tasks.

Upgrading Oracle Applications

This manual contains all the product specific pre- and post-upgrade steps that are required to upgrade products from Release 10.7 (NCA, SC and character-mode) or Release 11 of Oracle Applications. This manual also contains an overview chapter that describes all the tasks necessary to prepare and complete a upgrade of Oracle Applications.

Multiple Organizations in Oracle Applications

This book describes the Oracle Applications organization model, which defines business units and the relationships between them in an arbitrarily complex enterprise. The book provides functional and technical overviews of multiple organizations, as well as information about how to set up and implement multiple organizations in the relevant Oracle Applications products.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Training and Support

Training

Oracle Education offers a complete set of training courses to help you and your staff master Oracle Applications. We can help you develop a training plan that provides thorough training for both your project team and your end users. We will work with you to organize courses appropriate to your job or area of responsibility.

Training professionals can show you how to plan your training throughout the implementation process so that the right amount of information is delivered to key people when they need it the most. You can attend courses at any one of our many Educational Centers, or you can arrange for our trainers to teach at your facility. In addition, we can tailor standard courses or develop custom courses to meet your needs.

Support

From on-site support to central support, our team of experienced professionals continually provides you with whatever help and information you need to keep Oracle Projects working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in accounting, Oracle8 database management, and your particular hardware and software environment.

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 45 software modules for financial management, supply chain management, manufacturing, project systems, human resources, and sales and service 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 Proprietary, Confidential Information—Use Restricted by Contract

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 140 countries around the world.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. 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 is actively engaged with other market leader 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/>.

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing PL/SQL 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.

Thank You

Thank you for using Oracle Activity Management Gateway and this technical reference manual.

We value your comments and feedback. This manual contains a Reader's Comment Form you can use to explain what you like or dislike about AMG or this manual. Mail your comments to the following address or call us directly at (650) 506-7000.

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

Or, send electronic mail to appsdoc@us.oracle.com.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

High-Level Design

This chapter contains the following sections:

- Overview of the Activity Management Gateway (AMG). This section describes some of the ways that you can use the AMG to integrate Oracle Projects and external management systems.
- Integrating an External System with Oracle Projects. Follow the steps in this section carefully. A properly integrated system ensures that your external system can access the Oracle Projects database and that your Oracle Applications users can obtain the privileges necessary to use the application programming interfaces (APIs) discussed in this manual.
- Security Requirements. Follow the steps in this section to ensure proper security when users access Oracle Projects data from an external system.
- Handling Error Messages. This section describes how the AMG creates error messages, and how to display them in an external application.
- Standard Activity Management Gateway (AMG) Parameters. This section describes the standard input and output parameters shared by most of the APIs.
- Common APIs. This section provides details about APIs (GET_MESSAGES, GET_DEFAULTS, and GET_ACCUM_PERIOD_INFO) that are available for use in all project, budget, resource, and status APIs.

Overview of the Activity Management Gateway

The Oracle Activity Management Gateway (AMG) allows you to integrate Oracle Projects and third-party systems to build a complete management tool. You can combine the functionality of your preferred system with the features of Oracle Projects, and then safely share data and exchange information.

The AMG contains more than 150 application programming interfaces (APIs) that:

- Perform real-time or batch sharing of data between your system and Oracle Projects, thereby eliminating duplicate data entry
- Share business rules and workflow from one system to the other
- Share setup, project planning, resource planning, budgeting, actuals, and progress data

Refer to the Detailed Design chapters for detailed descriptions of the APIs.

Applications of the Activity Management Gateway

The AMG is a generic tool you can use to integrate Oracle Projects with many types of external or third-party systems, including:

- **Collaborative project planning and scheduling systems.** Integrate your enterprise business systems with team-oriented project planning and scheduling tools to provide communication links throughout your company.
- **Sales management systems.** Enter your sales order using a sales management system and call APIs to create a project in Oracle Projects based on the order information.
- **Work management systems.** Use the AMG to tailor a comprehensive solution that includes your work management system. Companies in the utilities industry commonly use this type of system.
- **Customer asset management and plant maintenance systems.** Share information about work orders, tasks, assets, crew labor charges, and inventory transactions charged to a project.
- **Project manufacturing systems.** Join inventory, manufacturing, and financial applications using the AMG, as Oracle's project manufacturing solution does.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Where Information Originates

The AMG APIs make two-way communication possible between Oracle Projects and a third-party external system. For example, if a purchase order issued against a task is being processed within your enterprise, you can restrict that project's task so it can't be deleted from a desktop project management system. (For more information about restricting certain actions, see: Controlling Actions in Oracle Projects; page F - 2.)

Table 2 - 1 illustrates the types of information that originates in Oracle Projects:

<i>Information that originates in Oracle Projects</i>	<i>Comments</i>
Project templates with Quick Entry (overridable) fields	You can override some of the template's default values when you create a project.
Resources	
Organizations	
Calendars (both GL and PA periods)	
Estimate to Complete (planned for a future release)	
Actuals: - Cost amounts (raw and burdened) - Commitments (raw and burdened) - Quantities - Revenue - PA or GL period - Inception-to-date - Period-to-date	Oracle Projects acts as the central repository of all project actuals, maintains common business rules (such as transaction controls), and collects a wide variety of transactions. Such transactions include phone usage records, labor, depreciation, commitments, usages, and expenses. Oracle Projects also performs complex cost burdening, generates revenue, and sends summarized information to external systems.

Table 2 - 1 Information that originates in Oracle Projects (Page 1 of 1)

Table 2 – 2 illustrates the types of information that originates in an external system (in this case, a project management system).

<i>Information that originates in your project management system</i>	<i>Comments</i>
Projects and tasks of the work breakdown structure (WBS)	
Budgets – Types – Time-phased – Amounts – Quantities – Baseline	Project managers can enter and baseline budgets from their preferred project management system or from Oracle Projects. Accounting personnel can enter budgets directly into Oracle Projects. Both types of employees can draft and update their own budget versions. Budgets created using project management systems integrate with Oracle Projects' budget calculation extensions.
Schedules and schedule changes	
Task parent reassignment	You can reassign a task to a different parent task as long the reassigned task remains under the same top task.
Percent complete – Project level – WBS (any level)	Once you send this information to Oracle Projects, you can use billing extensions to produce progress billings. You can view this information in Oracle Projects using the project status inquiry (PSI) client extension.
Earned value progress reporting – Budgeted Cost of Work Scheduled – Budgeted Cost of Work Performed – Actual Cost of Work Performed – Budget at Completion	You can use earned value reporting to determine cost variance, schedule variance, and variance at completion. To view this information in Oracle Projects, use the PSI client extension.

Table 2 – 2 Information that originates in your project management system (Page 1 of 1)

Integrating Your External System with Oracle Projects

After you install and implement Oracle Projects, you can integrate your external system with Oracle Projects. Follow the steps below to ensure that your external system can access the Oracle Projects database and that your Oracle Applications users can obtain the privileges necessary to use the APIs discussed throughout this manual.

Step 1 Create a database role

Create a special database role and assign it to anyone who will use the AMG APIs. You need to perform this step only once for each database, regardless of the number of users. Users can define their own role names. Oracle Projects provides the script **pacrrole.sql** to create and assign these database roles. The script resides in the `$PA_TOP/patchsc/107/sql` directory on the server and creates an output file called **pacrrole.lst**. Run the script from any directory in which you have write privileges. You run the script as any user with a Create Role privilege, such as SYSTEM or SYS. The script requires the following arguments:

- Activity Management Gateway role name, such as PMXFACE
- Username for the Oracle Applications user account, such as APPS
- Password for the Oracle Applications user account, such as APPS
- Username for the Oracle Projects user account, such as PA
- Password for the Oracle Projects user account, such as PA

From a SQL*Plus session, use the following syntax to run the script:

```
start $PA_TOP/patchsc/107/sql/pacrrole.sql &role &un_apps  
&pw_apps &un_pa &pw_pa
```

For example, to create the role PMXFACE in the APPS account, enter:

```
start $PA_TOP/patchsc/107/sql/pacrrole.sql PMXFACE APPS  
APPS PA PA
```

The script creates the role and grants the necessary privileges on the required database objects. Check the file **pacrrole.lst** to ensure that the script completed successfully.

Step 2 Create an Oracle Applications user

All API users must first be defined as Oracle Applications users. To define Oracle Applications users and their required responsibilities, use

the Oracle Applications Release 11.5.4 Users window. See: *Oracle Applications System Administrator's Guide*.

Step 3 **Create a database user**

After you have defined an Oracle Applications user with the required responsibilities, you must create a database user. The Oracle Applications username and the database username must be identical. Oracle Projects provides the script **pacruser.sql** to create database users. The script resides in the \$PA_TOP/patchsc/107/sql directory on the server. The script creates an output file called **pacruser.lst**. Run the script from any directory in which you have write privileges. You should run the script as any user with a Create User privilege, such as SYSTEM or SYS. The script requires the following arguments:

- Activity Management Gateway role name, such as PMXFACE. You must use the same role name that you created in Step 1.
- System username, such as SYSTEM
- Password for the system user, such as MANAGER
- Proposed username
- Proposed password
- Username for the Oracle Applications user account, such as APPS
- Username for the Oracle Projects user account, such as PA

From a SQL*Plus session, use the following syntax to run the script:

```
start $PA_TOP/patchsc/107/sql/pacruser.sql &role &sys_un
&sys_pwd &uname &pwd &un_apps &un_pa
```

To create the user JCLARK with a password of WELCOME, for example, enter:

```
start $PA_TOP/patchsc/107/sql/pacruser.sql PMXFACE SYSTEM
MANAGER JCLARK WELCOME APPS PA
```

Check the file **pacruser.lst** to ensure that the script completed successfully.

Note: Oracle Projects provides a template script, **patempus.sql**, which facilitates the processing of large amounts of data. This script generates an output file, **pagenus.sql**, while creating a large number of database users from existing Oracle Applications users. You can add WHERE conditions to narrow the criteria. Run the script from any directory in which you have write privileges. You need to run this script only for users who require access to the AMG APIs.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Caution: The Oracle Applications user is different from the database user, even if they share the same username. Each Oracle Applications user and database user has a distinct password, which you must maintain individually. Changing an Oracle Application user's password does not automatically change the database user's password. Users can choose different passwords for their Oracle Applications and database usernames.

Step 4 Set up your product in Oracle Projects

Set up your external system as a source product in Oracle Projects using the Source Products window.

Note: If the database has been exported and then imported **and** you performed Steps 1, 2, and 3 before the export/import, some or all of the grants may not work properly after the import. Use the script **pacrgran.sql** (located in \$PA_TOP/admin/sql) to restore the grants. From a SQL*Plus session, use the following syntax to run the script:

```
start $PA_TOP/admin/sql/pacrgran.sql &role &un_apps  
&pw_apps &un_pa &pw_pa
```

For example:

```
start $PA_TOP/admin/sql/pacrgran.sql PMXFACE APPS  
APPS PA PA
```

Step 5 Enable your AMG site license

The Activity Management Gateway and Oracle Projects are separate products, and each must be purchased and licensed separately. To enable the site license, set the profile option **PA: Licensed to use AMG** to Yes in the Oracle Applications System Profile Options window. For more information about updating profile options, see Profile Options in the *Oracle Projects User Guide*.



Attention: When you set the license profile option to Yes, you are consenting to the terms of the license and confirming that you have purchased AMG.

Security Requirements

Each interface or application that you develop using the AMG APIs must prompt users for identifying information and then set up global variables. Follow the steps below to ensure that proper security is enforced when users access Oracle Projects data from an external system.

Step 1 Authenticate the user

Your external system should prompt users for their Oracle Projects username and password and then use this login information to establish a connection to the Oracle Projects database. After three unsuccessful attempts to establish a connection, the external system should abort and display an error message.

Step 2 Choose a responsibility

Because Oracle Applications responsibilities control users' access to Oracle Projects data, Oracle Applications users must choose a specific responsibility from the list of their valid responsibilities. Oracle Projects provides this information in the view PA_USER_RESP_V.

Column descriptions for PA_USER_RESP_V are shown here:

Column Descriptions

Name	Null?	Type	Description
USER_ID	NOT NULL	NUMBER	The reference code that uniquely identifies the user
USER_NAME	NOT NULL	VARCHAR2(100)	Username used to log into Oracle Applications
RESPONSIBILITY_ID	NOT NULL	NUMBER	Identification code of the responsibility
RESPONSIBILITY_NAME	NOT NULL	VARCHAR2(100)	Name of the responsibility

The login username entered in Step 1 controls the Oracle Applications responsibilities retrieved by this view. Once a user chooses a responsibility, the external system also stores the corresponding USER_ID and RESPONSIBILITY_ID. The RESPONSIBILITY_NAME field is for display purposes only and need not be stored.

Note: Because Oracle Applications store user names in uppercase letters, you should convert login user names to uppercase letters before using them as keys. Database connection strings are case insensitive. For example, a login username entered as "scott" is stored as "SCOTT". Typical PL/SQL code to display the responsibilities reads as follows:

Oracle Proprietary, Confidential Information--Use Restricted by Contract

```

Login Name is stored in l_login_name
l_upper_login_name = UPPER(l_login_name)
Select RESPONSIBILITY_NAME, USER_ID,
RESPONSIBILITY_ID
from PA_USER_RESP_V where
USER_NAME = l_upper_login_name

```

Caution: Do not use UPPER(USER_NAME) in the WHERE clause. Expressions used in WHERE clauses disable the index and impair performance. Always convert a value to uppercase in your code and use the converted string in the WHERE clause.

Step 3 Set up global variables

Access to Oracle Projects is controlled not only by a user's responsibility, but also by the user's organization for that responsibility. To ensure that the level of access to data matches a user's organization, use the API SET_GLOBAL_INFO to set up global variables. This API is located in the public API package PA_INTERFACE_UTILS_PUB.

SET_GLOBAL_INFO is a PL/SQL procedure that sets the global variables necessary to access data in a multi-org implemented environment.

The arguments P_RESPONSIBILITY_ID and P_USER_ID must have valid values. If the arguments contain null or invalid values, SET_GLOBAL_INFO returns an error status.

Parameters for SET_GLOBAL_INFO are shown below:

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RESPONSIBILITY_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the chosen responsibility (refer to Step 2: page 2 - 8)
P_USER_ID	IN	NUMBER	Yes	The identification code of the corresponding user returned by the view (refer to Step 2: page 2 - 8)
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		Return status. Valid values are: 'S' - Success 'E' - Error 'U' - Unexpected error

After completing these steps, external systems call the remaining Oracle Projects APIs necessary to complete the task, such as CREATE_PROJECT, UPDATE_PROJECT, SELECT_RESOURCE_LIST, or CREATE_DRAFT_BUDGET.

Handling Error Messages

The APIs in Oracle Activity Management Gateway return all applicable error messages for all updates, changes, or additions to a work breakdown structure or budget.

NOTE: Upgrading Error Message Handling

In the original release of Oracle Activity Management Gateway Release 10.7 and Release 11.0, the APIs returned a single error message and terminated processing if they encountered a violation of any Oracle Projects business rule. If you have upgraded from one of these releases, you may want to update your code to take advantage of the enhanced messaging.

For example, you might want to modify a single-error dialog box to display all the error messages, or write all of the error messages to a file and then launch an editor or window to display the file. Instructions are included in this section and under GET_MESSAGES: page 2 – 18.

How Error Messages Are Created

The APIs do not stop processing when an error is encountered. Processing continues until all items are validated and error messages generated. However, if any errors are encountered during one of these processes, no records are saved to the Oracle Projects database.

AMG messages contain all the information necessary to identify the data element related to each error. This information includes:

For WBS data:

- project reference
- task reference

For budget data:

- project reference
- task reference
- budget type
- budget start date

Displaying Error Messages

Because AMG can be used to develop both real-time and batch integrations with external systems, display of error messages must be handled in the external application.

Use the API `PA_INTERFACE_UTILS_PUB.get_messages` to retrieve the error messages. For details on this API and an example of PL/SQL code to retrieve the error messages, see: `GET_MESSAGES`: page 2 – 18.

See Also

Appendix I: List of AMG Messages

Standard API Parameters

All Activity Management Gateway APIs have a set of standard input and output parameters that are used in most of the public procedures. The table below describes each of these standard API parameters.

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Req?</u>	<u>Description</u>
P_COMMIT	IN	VARCHAR2(1)	Yes	Set this parameter to 'T' (True) if you want the APIs themselves to issue the commit to the database. Default = 'F' (False)
P_INIT_MSG_LIST	IN	VARCHAR2(1)	Yes	Set this parameter to 'T' (True) if you want to initialize the global message table. Default = 'F' (False)
P_API_VERSION_NUMBER	IN	NUMBER	Yes	For the current version of the APIs, this parameter must be set to 1.0. This may change in future versions of the APIs.
P_RETURN_STATUS	OUT	VARCHAR2(1)		The return status of the APIs. Possible values are: 'S' - The API completed successfully 'E' - Business rule violation error 'U' - Unexpected error (like an Oracle error)
P_MSG_COUNT	OUT	NUMBER		Holds the number of messages in the global message table. Calling programs should use this as the basis to fetch all the stored messages. If the value for this parameter = 1, then the message code is available in P_MSG_DATA. If the value of this parameter > 1, you must use the GET_MESSAGES API to retrieve the messages.
P_MSG_DATA	OUT	VARCHAR2(2000)		Holds the message code, if the API returned only one error/warning message. Otherwise, the column is left blank.

APIs That Use Composite Datatypes

Read this section if you use PL/SQL 2.3 or higher to call AMG APIs that use composite datatypes, such as an array of records

If you assign a value to a subset of variables in a PL/SQL array, first assign the values to a PL/SQL record and then add the record to the PL/SQL array. It is important to perform the steps in this order due to the way PL/SQL handles assignments to an array.

The following sample PL/SQL code shows how to assign values to the P_BUDGET_LINES_IN PL/SQL table in the CREATE_DRAFT_BUDGET API.

```

DECLARE
--variables needed for API standard parameters
l_api_version_number    NUMBER :=1.0;
l_commit                VARCHAR2(1) := 'F';
l_return_status         VARCHAR2(1);
l_init_msg_list         VARCHAR2(1);
l_msg_count             NUMBER;
l_msg_data              VARCHAR2(2000);
l_data                 VARCHAR2(2000);
l_msg_entity            VARCHAR2(100);
l_msg_entity_index     NUMBER;
l_msg_index             NUMBER;
l_msg_index_out        NUMBER;
l_encoded               VARCHAR2(1);

--variables needed for Oracle Project specific parameters
l_pm_product_code       VARCHAR2(10);
l_pa_project_id         NUMBER;
l_pm_project_reference  VARCHAR2(25);
l_budget_type_code     VARCHAR2(30);
l_change_reason_code   VARCHAR2(30);
l_description           VARCHAR2(255);
l_entry_method_code    VARCHAR2(30);
l_resource_list_name   VARCHAR2(60);
l_resource_list_id     NUMBER;
l_budget_lines_in
    PA_BUDGET_PUB.budget_line_in_tbl_type;
l_budget_lines_in_rec
    PA_BUDGET_PUB.budget_line_in_rec_type;
l_budget_lines_out
    PA_BUDGET_PUB.budget_line_out_tbl_type;
l_line_index           NUMBER;
l_line_return_status   VARCHAR2(1);

API_ERROR              EXCEPTION;

BEGIN
--PRODUCT RELATED DATA
l_pm_product_code := 'SOMETHING';

```

```

--BUDGET DATA
l_pm_project_reference := 'TEST';
l_budget_type_code := 'AC';
l_change_reason_code := 'ESTIMATING ERROR';
l_description := 'New description -> 2';
l_entry_method_code := 'PA_LOWEST_TASK_BY_PA_PERIOD';
l_resource_list_id := 1001;

```

The previous example shows how to assign values to a subset of the PL/SQL table. To assign values only to PA_TASK_ID and RESOURCE_LIST_MEMBER_ID in the P_BUDGET_LINES_IN table, first assign these values to BUDGET_LINES_IN_REC and then add BUDGET_LINES_IN_REC to the BUDGET_LINES_IN PL/SQL table, as illustrated in the following example.

```

--BUDGET LINES DATA
a := 5;
FOR i IN 1..a LOOP
  if i = 1 THEN
    l_budget_lines_in_rec.pa_task_id :=1496;
    l_budget_lines_in_rec.resource_list_member_id:=1731;
  elsif i = 2 THEN
    l_budget_lines_in_rec.resource_list_member_id:=1732;
    l_budget_lines_in_rec.pa_task_id := 1495;
  elsif i = 3 THEN
    l_budget_lines_in_rec.resource_list_member_id:=1733;
    l_budget_lines_in_rec.pa_task_id := 1494;
  elsif i = 4 THEN
    l_budget_lines_in_rec.resource_list_member_id:=1734;
    l_budget_lines_in_rec.pa_task_id := 1492;
  elsif i = 5 THEN
    l_budget_lines_in_rec.resource_list_member_id:=1735;
    l_budget_lines_in_rec.pa_task_id := 1491;
  end if;
  l_budget_lines_in_rec.quantity:=97;
  l_budget_lines_in_rec.period_name:= 'P06-03-95';
  l_budget_lines_in_rec.raw_cost:=300;
  l_budget_lines_in(i) := l_budget_lines_in_rec;
END LOOP;
pa_budget_pub.create_draft_budget
( p_api_version_number => l_api_version_number
, p_msg_count => l_msg_count
, p_msg_data => l_msg_data
, p_return_status => l_return_status
, p_pm_product_code => l_pm_product_code
, p_pa_project_id=> l_pa_project_id
, p_pm_project_reference => l_pm_project_reference
, p_budget_type_code=> l_budget_type_code

```

```
,p_change_reason_code => l_change_reason_code
,p_description => l_description
,p_entry_method_code => l_entry_method_code
,p_resource_list_name => l_resource_list_name
,p_resource_list_id=> l_resource_list_id
,p_budget_lines_in => l_budget_lines_in
,p_budget_lines_out => l_budget_lines_out );
```

Other Parameters

The APIs for AMG typically allow you to reference Oracle Projects entities by either identification codes or reference codes. For example, you can refer to a project using either the `PROJECT_ID` or the `PM_PROJECT_REFERENCE`.

Identification codes are usually system-generated numbers assigned to the entity by Oracle Projects. The reference code is usually a character name or description for the entity.

If a project already exists in Oracle Projects, you can reduce your processing time by passing identification codes instead of reference codes to the AMG APIs. The APIs read identification codes and convert passed reference codes to their corresponding identification codes before execution.

If an API requires a given entity for processing, you must pass either the entity's reference code parameter or the entity's identification code parameter, but not both. If the API cannot find or derive a reference code for the required identification code parameter, the API will generate an error message and abort processing.

When passing parameters to an Activity Management Gateway API, you should use *named notation* (see the following example), which enables you to pass only the parameters required by a particular API. Using named notation can significantly improve the processing of update APIs.



Attention: If you pass an API parameter as NULL, the API updates the column in the database with a NULL value. If you do not want to update a column, do not pass the corresponding parameter.

Example of Named Notation

Using the API `DELETE_PROJECT`, you can pass either the `PROJECT_ID` or the `PM_PROJECT_REFERENCE` for the project. The

following example passes the project identification code P_PA_PROJECT_ID. The SQL statement below omits optional parameters, such as P_INIT_MSG_LIST and P_COMMIT, so that they will not be updated in the table.

```
Delete_Project(p_api_version_number => 1.0
, p_msg_count => l_msg_count
, p_msg_data => l_msg_data
, p_return_status => l_return_status
, p_pm_product_code => l_product_code
, p_pa_project_id => 1043
);
```

Valid Data for Other Parameters

The AMG APIs often use identification code and reference code parameters for many Oracle Projects entities. To facilitate the retrieval of valid parameter data, the AMG supplies views to Oracle Projects data. These views are described in the Detailed Design chapters.

Common APIs

The following APIs are available for use in all modules and are located in the public API package PA_INTERFACE_UTILS_PUB.

GET_MESSAGES

GET_MESSAGES is a PL/SQL procedure that retrieves messages from the message stack. If an API detects only one error during execution, the API returns the error text via the standard API output parameter P_MSG_DATA. If the API detects multiple errors, you must use the GET_MESSAGES API to retrieve the messages.

The following table shows the parameters in GET_MESSAGES.

Name	Usage	Type	Req?	Description
P_ENCODED	IN	VARCHAR2(1)	No	Passes 'T' (True) if you want only the message code to be returned in the p_data parameter. Default = 'F' (False)
P_MSG_COUNT	IN	NUMBER	No	Passes the P_MSG_COUNT value returned by the API that raised the error. If P_MSG_COUNT = 1, this API returns the error text. Otherwise, this API calls the message handling package FND_MSG_PUB.
P_MSG_DATA	IN	VARCHAR2(80)	Yes	Passes the P_MSG_DATA value returned by the API that raised the error
P_DATA	OUT	VARCHAR2(2000)		The message code (if P_ENCODED = 'T') or the message text (if P_ENCODED = 'F')
P_MSG_INDEX_OUT	OUT	NUMBER		The index (cell) of the message in the global message stack

Sample Code for Handling Multiple Messages

The following sample PL/SQL code shows how you can use GET_MESSAGES to handle multiple messages in an external application.

This example uses the procedure PA_PROJECT_PUB.create_project. You can initialize the message stack at the beginning of the session, as in this example, or for each project.

All messages are held in PL/SQL memory. For a large installation where there may be a lot of error messages, you can store all messages related to a project in a file or in the database, and initialize the

message stack frequently. You use FND_MSG_PUB.initialize to initialize the message stack.

You can temporarily insert the messages into a table, as shown in the example. Or, if you are running a 'C' program or using PL/SQL file I/O utilities, you can write the messages to a log file. If you write the messages to a log file, you may want to create header information in the log file. You can then launch a text editor to instantly display the error messages.

Note: The parameter *p_msg_index_out* in this code sample was added as a workaround to a known bug in Oracle AOL. This parameter may be removed in subsequent releases of AMG. If your AMG code stops working after applying patches later than 754949, values sent as this parameter would be a likely cause.

Following is the sample code:

```
-- Initialize the message stack
FND_MSG_PUB.initialize;
pa_project_pub.create_project
  (p_api_version_number => l_api_version_number
  ,p_commit => l_commit
  ,p_init_msg_list => 'F'
  ,p_msg_count => l_msg_count
  ,p_msg_data => l_msg_data
  ,p_return_status => l_return_status
  ,p_pm_product_code => l_pm_product_code
  ,p_project_in => l_project_in_rec
  ,p_project_out => l_project_out_rec
  ,p_key_members => l_key_member_tbl
  ,p_class_categories => l_class_category_tbl
  ,p_tasks_in => l_tasks_in
  ,p_tasks_out => l_tasks_out);
IF l_return_status != 'S'
THEN
  if l_msg_count > 0 THEN
    for i in 1..l_msg_count loop
      pa_interface_utils_pub.get_messages (
        ,p_encoded => 'F'
        ,p_msg_count => l_msg_count
        ,p_msg_data => l_msg_data
        ,p_data => l_data
        ,p_msg_index_out => l_msg_index_out
      );
    end loop;
  end if;
end if;
```

```

-- Insert the messages from l_data into error_table
Insert into error_table (error_msg) values (l_data);
    end loop;
    end if;
END IF;

```

GET_DEFAULTS

GET_DEFAULTS is a PL/SQL procedure that returns the default values required to initialize the VARCHAR2, NUMBER, and DATE variables in your programs. This API has no input parameters.

The following table shows the parameters in GET_DEFAULTS.

Name	Usage	Type	Req?	Description
P_DEF_CHAR	OUT	VARCHAR2(3)		Returns the default value for character variables
P_DEF_NUM	OUT	NUMBER		Returns the default value for number variables
P_DEF_DATE	OUT	DATE		Returns the default value for date variables
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

Default values are useful when you conditionally set a value for a variable. For example, while updating a project, you may conditionally set the value for the variable L_DISTRIBUTION_RULE, depending on whether you want to update the distribution rule in Oracle Projects. To accomplish this, you would use a PL/SQL statement similar to this:

```

Pa_interface_utils.get_defaults (p_def_char => l_def_char,
                                p_def_num  => l_def_num,
                                p_def_date => l_def_date,
                                p_return_status => l_return_status,
                                p_msg_count => l_msg_count,
                                p_msg_data => l_msg_data );
l_distribution_rule := l_def_char;
l_customer_id      := l_def_num;
l_end_date         := l_def_date;

```

GET_ACCUM_PERIOD_INFO

GET_ACCUM_PERIOD_INFO is a PL/SQL procedure that returns information about the last period through which the project is summarized in Oracle Projects, as well as the current reporting period. Use this API to see if the actuals in your external system are current with those in Oracle Projects.

The following table shows the parameters in GET_ACCUM_PERIOD_INFO.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Y	API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PROJECT_ID	IN	NUMBER	Y	Unique identifier of the project
P_LAST_ACCUM_PERIOD	OUT	VARCHAR2		The period up to which the project has been summarized
P_LAST_ACCUM_START_DATE	OUT	DATE		The start date of the last summarized period
P_LAST_ACCUM_END_DATE	OUT	DATE		The end date of the last summarized period
P_CURRENT_REPORTING_PERIOD	OUT	VARCHAR2		The PA period that is defined in the current reporting period
P_CURRENT_PERIOD_START_DATE	OUT	DATE		The start date of the current reporting period
P_CURRENT_PERIOD_END_DATE	OUT	DATE		The end date of the current reporting period

This PL/SQL example demonstrates a typical use of GET_ACCUM_PERIOD_INFO:

```
Pa_interface_utils.get_accum_period_info
(p_api_version_number => l_api_version_number
 l_msg_count => l_msg_count,
 p_msg_data => l_msg_data,
 p_return_status => l_return_status,
 p_project_id => l_project_id,
 p_last_accum_period => l_last_accum_period,
 p_last_accum_start_date => l_last_accum_start_date,
 p_last_accum_end_date => l_last_accum_end_date,
 p_current_reporting_period => l_current_reporting_period,
 p_period_start_date => l_period_start_date,
 p_period_end_date => l_period_end_date);
```

Oracle Proprietary, Confidential Information--Use Restricted by Contract

How to Use Activity Management Gateway APIs

The examples in this chapter show how to use the public project and budget APIs to share information between an external system and Oracle Projects.

After you read this chapter, use the sample PL/SQL code in Appendices A, B, and C as a guide when creating your own integration package.



Attention: These details and specifications are subject to change without notice.

Using Project APIs

The following example describes how to create an interface between Oracle Projects and the project and task information entered in your system. Depending on your company's business needs, your implementation of the project APIs may be more or less complex than the scenario shown here. As you work through the example, you may want to refer to information elsewhere in the manual:

- For a detailed description of the project APIs, see Project APIs: page 5 – 2.
- Most of the AMG APIs use a standard set of input and output parameters. For a description of these parameters, see Standard API Parameters: page 2 – 13.
- For an example of PL/SQL code for creating a project without using composite datatypes, see Creating a Project Using the Load–Execute–Fetch APIs: page A – 2.

Step 1 **Connect to an Oracle database**

To ensure that proper security is enforced while accessing Oracle Projects data, follow the steps in Security Requirements: page 2 – 8.

Step 2 **Select a source template or project**

When using the AMG APIs to create a new project in Oracle Projects, first select a project template from which to create the new project. Oracle Projects will not create a new project unless you perform this step. Use the API view PA_SELECT_TEMPLATE_V to select a valid Oracle Projects source template.

Alternatively, you can choose a source project. The only difference between templates and projects is that the field TEMPLATE_FLAG for templates is set to Y. All projects originate from templates, and the originating template determines which Quick Entry fields appear in your new project. In this section, all instructions involving source templates also apply to source projects.

Step 3 **Get the Quick Entry fields of the source template**

After you select a source template, use the PA_SOURCE_TEMPLATE_ID to retrieve the Quick Entry fields associated with the template. You assign Quick Entry fields to a template when you create the template in Oracle Projects. For more information about Quick Entry fields, see

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_PROJECT_COPY_OVERRIDES in the *Oracle Projects Technical Reference Manual*.

The view PA_OVERRIDE_FIELDS_V displays all the Quick Entry fields associated with a particular template. The user interface you design should display at least the Display Name, Value, and Mandatory fields and should allow users to enter information only into the Value field.

An example of a user interface that meets these requirements is shown below:

Field	Value	Mandatory
Project Name	Build Castle	<input checked="" type="checkbox"/>
Distribution Rule	COST/COST	<input type="checkbox"/>
Project Manager	7	<input type="checkbox"/>
Funding Source	Federal	<input type="checkbox"/>

Buttons: OK, Cancel, Retrieve Valid Values

Figure 3-3: Example of a Quick Entry (Overrideable Fields) Window

Step 4 Enter valid data for the Quick Entry fields

Lists of values (LOVs) validate most of the Quick Entry fields. The view PA_OVERRIDE_FIELDS_V retrieves the name of the view that contains the valid data for the active row and returns this name in the field LOV_VIEW_NAME. Your project management tool can use this information to dynamically access the appropriate view.

For example, if you place your cursor in the Funding Source field and choose Retrieve Valid Values, your project management tool will display a screen with two columns, Code and Description. Values retrieved

Oracle Proprietary, Confidential Information--Use Restricted by Contract

from the database view PA_CLASS_CATEGORIES_LOV_V will appear under these two column headings.

You can also use the following views to retrieve lists of values for a project's Quick Entry fields:

- PA_PROJECT_STATUS_CODES_LOV_V
- PA_DISTRIBUTION_RULES_LOV_V
- PA_KEY_MEMBERS_LOV_V
- PA_ORGANIZATIONS_LOV_V
- PA_CUSTOMERS_LOV_V

Step 5 Interface project information to the server

Not all tools can call the AMG APIs that use composite datatypes. Tools that do not support composite datatypes must call the supplementary Load-Execute-Fetch APIs, also provided in the AMG. The Load-Execute-Fetch APIs include procedures to initialize, load, execute, fetch, and clear data. For more information, see View Definitions: page 4 – 2.

Use these APIs only if you use a tool that does not support composite datatype parameters. If the tool (for example, Oracle PL/SQL Version 2.3 or higher) supports composite datatype parameters, you can call the CREATE_PROJECT and UPDATE_PROJECT APIs directly.

Figure 3-2 illustrates the flow of the Load-Execute-Fetch project procedures.

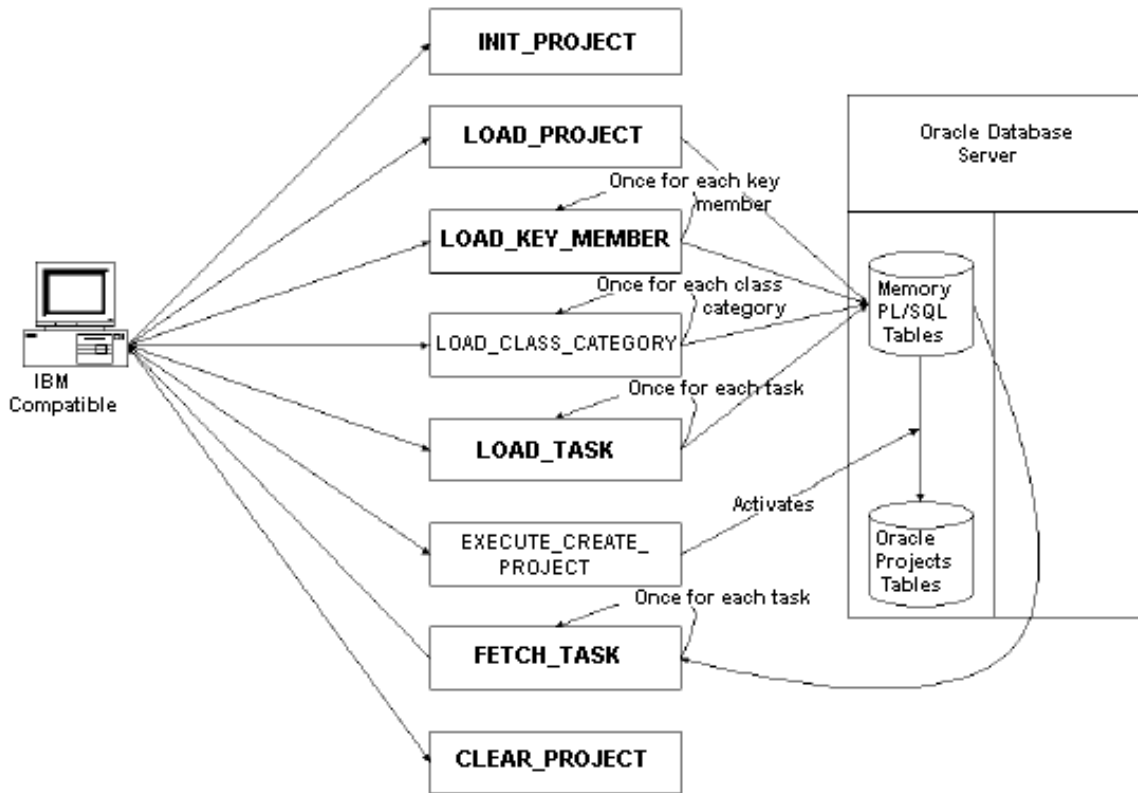


Figure 3-3: Load-Execute-Fetch Procedures for Project APIs

In the example above, INIT_PROJECT resets the server-side global PL/SQL tables that temporarily store the project and task data.

Once you set up these tables, you can use LOAD_PROJECT to move the project data to the Oracle Projects database.

The following table illustrates the relationship between the information in the user interface and LOAD_PROJECT:

<i>Quick Entry Field Value</i>	<i>LOAD_PROJECT Parameter</i>
NAME	P_PROJECT_NAME
DESCRIPTION	P_DESCRIPTION
START_DATE	P_START_DATE
COMPLETION_DATE	P_COMPLETION_DATE
PROJECT_STATUS_CODE	P_PROJECT_STATUS_CODE
PUBLIC_SECTOR_FLAG	P_PUBLIC_SECTOR_FLAG
DISTRIBUTION_RULE	P_DISTRIBUTION_RULE
CARRYING_OUT_ORGANIZATION_ID	P_CARRYING_OUT_ORGANIZATION_ID
CUSTOMER_NAME	P_CUSTOMER_ID

Table 3 - 1 How the User Interface Relates to LOAD_PROJECT (Page 1 of 1)

LOAD_PROJECT passes the values entered into the Quick Entry value field to their corresponding parameters. LOAD_PROJECT passes additional parameters, depending on whether you are updating an existing project or creating a new one. When you create a new project, this procedure must also pass the following parameters:

- P_PM_PROJECT_REFERENCE passes the unique reference code that identifies the project in the external system.
- P_CREATED_FROM_PROJECT_ID passes the unique reference code that identifies the source template in Oracle Projects (PA_SOURCE_TEMPLATE_ID).

If your project has multiple key members or class categories, you must call the APIs LOAD_KEY_MEMBER and LOAD_CLASS_CATEGORY for every key member and class category associated with your project.

During project creation, the Quick Entry fields Key Members and Class Category are related to the input parameters shown in the two tables that follow.

Table 3 - 2 shows the input parameters for the key member quick entry field.

<i>Key Member Quick Entry Field</i>		<i>Input Parameter for LOAD_KEY_MEMBERS</i>	
KEY_MEMBER (value)		P_PERSON_ID	
KEY_MEMBER (display_name)		P_PROJECT_ROLE_TYPE	

Table 3 – 2 Key Member Quick Entry Field (Page 1 of 1)

Table 3 – 3 shows the input parameters for the class category quick entry field.

<i>Class Category Quick Entry Field</i>		<i>Input Parameter for LOAD_CLASS_CATEGORY</i>	
CLASS_CATEGORY (value)		P_CLASS_CODE	
CLASS_CATEGORY (display_name)		P_CLASS_CATEGORY	

Table 3 – 3 Class Category Quick Entry Field (Page 1 of 1)

Step 6 Interface task information to the server

After you interface the project-related data to the server, you can call LOAD_TASK to interface task-related data to the server-side global PL/SQL tables. Call LOAD_TASK once for every task in the project.



Attention: You must load parent tasks before you can load their subtasks.

Each task must specify at least the following information:

- P_PM_TASK_REFERENCE. The unique reference code that identifies the task in the external system.
- P_PM_PARENT_TASK_REFERENCE. The unique reference code that identifies the task's parent task. This parameter is left blank for top tasks.
- P_TASK_NAME. The name of the task.

For the names and descriptions of other parameters that LOAD_TASK can pass, see LOAD_TASK: page 5 – 46.

Step 7 **Start the server-side process**

Once the Load procedures have successfully moved project and task data to the Oracle Projects global PL/SQL tables, call the procedure `EXECUTE_CREATE_PROJECT` to process the project and task data that you interfaced to the global PL/SQL tables. In addition to the standard input and output parameters, this Execute procedure requires the following parameters:

- Input parameter: `P_PM_PRODUCT_CODE`, the identification code of the product exporting the project. For information about setting up your product (external system) as a source, refer to *Setting Up Your Product in Oracle Projects: page 2 – 7*.
- Output parameters:
 - `P_PA_PROJECT_ID`, the unique Oracle Projects identification code for the new project.
 - `P_PA_PROJECT_NUMBER`, the unique Oracle Projects number for the new project. If you have set up Oracle Projects to support manual project numbering, `P_PA_PROJECT_NUMBER` should be identical to the `P_PM_PROJECT_REFERENCE`. If you have implemented automatic numbering, this parameter returns an automatically generated number.

Step 8 **Get return values for tasks**

After the Load and Execute procedures create your project and tasks in Oracle Projects, use `FETCH_TASK` to return each unique task identification code from Oracle Projects. The key parameters for this procedure are the input parameter `P_TASK_INDEX`, which points to a single task, and the output parameters `P_PA_TASK_ID` and `P_PM_TASK_REFERENCE`.

To call the procedure for each task, you can write a simple program to call `FETCH_TASK` in a loop with `P_TASK_INDEX` as the stepping variable (1 through the total number of tasks). The output parameter `P_TASK_RETURN_STATUS` indicates whether the API handled the specific task successfully ('S'). If the parameter returns an 'E' or 'U', the task caused an error, and you must stop the Fetch procedure to retrieve the related error message. Fetch APIs do not return error message data. Instead, use `GET_MESSAGES` to retrieve the error text, as described in the next step.

Step 9 Retrieve error messages

Every AMG API includes two standard output parameters: P_RETURN_STATUS indicates whether the API was executed successfully, and P_MSG_COUNT shows the number of errors detected during the execution of the API.

If the API detects one error, the API returns the error message text. If the API detects multiple errors, use GET_MESSAGES to retrieve the error messages. See GET_MESSAGES: page 2 – 18.

Step 10 Finish the Load–Execute–Fetch process

After executing the Fetch procedures and retrieving any error messages, finish the Load–Execute–Fetch process by calling the API CLEAR_PROJECT and either save or rollback your changes to the database.

Using Budget APIs

The following example describes how to create an interface between Oracle Projects and the budget and budget line information in your external system. Depending on your company's business needs, your own implementation of budget APIs may be more or less complex than the scenario shown here.

As you work through this example, you may want to refer to information elsewhere in this manual:

- For a detailed description of the budget APIs, see Budget APIs: page 6 – 2.
- Most of the AMG APIs use a standard set of input and output parameters. See Standard API Parameters: page 2 – 13.
- For an example of PL/SQL code that creates a budget using Load–Execute–Fetch APIs, see Creating a Budget Using the Load–Execute–Fetch APIs: page B – 2.
- For an example of PL/SQL code that creates a budget using APIs that use composite datatypes, see Creating a Budget Using a Composite Datatype API: page C – 2.

Step 1 **Connect to an Oracle database**

To ensure that proper security is enforced while accessing Oracle Projects data, follow the steps in Security Requirements: page 2 – 8.

Step 2 **Get the budget data**

Before you send budget lines to the Oracle Projects database, you must first make some decisions that affect how the budget and budget lines are linked to other Oracle Projects data. This section provides sample PL/SQL select statements upon which you can model your own. The following pages describe the relationship between the selected values and budget or budget line information. Understanding this relationship helps you to determine which parameter values to pass to the budget and budget line APIs.

► **Select the budget type:**

Select a valid budget type. Oracle Projects predefines the budget types shown in the following table:

<i>Budget Type Code</i>	<i>Budget Type</i>
AC	Approved Cost Budget
AR	Approved Revenue Budget
FC	Forecast Cost Budget
FR	Forecast Revenue Budget

Table 3 – 4 Budget types predefined by Oracle Projects (Page 1 of 1)

The following PL/SQL statement retrieves the budget type information:

```
SELECT code,
       name
FROM   pa_budget_types_v
```

The selected value &CODE is related to the budget parameter P_BUDGET_TYPE_CODE.

Because cost and revenue budgets can contain different budget amounts, you must retrieve the budget amount code for the budget type. The following PL/SQL statement retrieves the appropriate budget amount code:

```
SELECT budget_amount_code
FROM   pa_budget_types
WHERE  budget_type_code = &code
```

The statement returns C if you have chosen a cost budget, and R if you have chosen a revenue budget. Table 3 – 5 illustrates the amounts each budget type can hold and their relation to the parameters of LOAD_BUDGET_LINE:

<i>Amount</i>	<i>LOAD_BUDGET_LINE Parameter</i>
Raw Cost	P_RAW_COST
Burdened Cost	P_BURDENED_COST
Cost Quantity	P_QUANTITY

Table 3 – 5 How budget amounts relate to API parameters (Page 1 of 2)

<i>Amount</i>	<i>LOAD_BUDGET_LINE Parameter</i>
Revenue	P_REVENUE
Revenue Quantity	P_QUANTITY

Table 3 – 5 How budget amounts relate to API parameters (Page 2 of 2)

► **Select the budget entry method:**

Oracle Projects predefines the budget entry methods shown in the following table:

Budget Entry Method Code	Budget Entry Method
PA_LOWEST_TASK_BY_PA_PERIOD	By lowest tasks and PA period, categorized by resource
PA_LOWEST_TASK_BY_GL_PERIOD	By lowest tasks and GL period, categorized by resource
PA_LOWEST_TASK_BY_DATE_RANGE	By lowest tasks and date range, categorized by resource

Table 3 – 6 Budget entry methods predefined by Oracle Projects (Page 1 of 1)

The following PL/SQL statement retrieves the budget entry method:

```
SELECT code
, name
, categorization_code
, entry_level_code
, entry_level_name
, time_phased_type_code
, time_phased_type_name
FROM pa_budget_entry_methods_v
```

The selected value CODE is related to the budget parameter P_ENTRY_METHOD_CODE.

You can use the other selected values later to retrieve other budget-related data from Oracle Projects. Possible values for other budget-related fields include:

- For CATEGORIZATION_CODE
 - R Categorized by resource
 - N Not categorized
- For ENTRY_LEVEL_CODE
 - P Budgeting at the project level
 - T Budgeting at the top task level
 - L Budgeting at the lowest task level
 - M Budgeting at both top and lowest task (mixed) level
- For &TIME_PHASED_TYPE_CODE
 - P Budget lines by PA periods
 - G Budget lines by GL periods
 - R Budget lines by date ranges
 - N Budget lines not time-phased

► **Select a resource list:**

If you select a budget entry method that is categorized by resources, you must select a resource list for the budget. The following PL/SQL statement retrieves the resource list information:

```
SELECT resource_list_id
, resource_list_name
, description
FROM pa_qry_resource_lists_v
```

Table 3 – 7 illustrates the relationship between certain selected values and budget parameters. Pass only one of the two values:

<i>Selected Value</i>	<i>Budget Parameter</i>
&RESOURCE_LIST_ID	P_RESOURCE_LIST_ID
RESOURCE_LIST_NAME	P_RESOURCE_LIST_NAME

Table 3 – 7 How selected values relate to budget parameters (Page 1 of 1)

► **Select other budget-related parameters:**

The parameter P_DESCRIPTION holds the description for a budget. Use the view PA_BUDGET_CHANGE_REASON_V to pass an explanation for any changes made to the budget. The following PL/SQL statement retrieves the reason for the budget change:

```
SELECT code,
       name
FROM pa_budget_change_reason_v
```

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Table 3 – 8 illustrates the relationship between certain selected values and budget parameters:

<i>Selected Value</i>	<i>Budget Parameter</i>
CODE	P_CHANGE_REASON_CODE
DESCRIPTION	P_DESCRIPTION

Table 3 – 8 How selected values relate to budget parameters (Page 1 of 1)

Step 3 Get budget line data

The choices you made for your budget data strongly affect your budget line data. These effects are described on the following pages.

► **Select amount fields:**

As shown in Table 3 – 5, cost budgets can contain raw cost, burdened cost, and cost quantity amounts, while revenue budgets can contain only revenue and revenue quantity amounts.

► **Select tasks:**

Depending on the budget entry level, the budget line should include the appropriate TASK_ID or TASK_REFERENCE. With project-level budgeting, you do not pass task-related parameters. With task-level budgeting (top, lowest, or mixed), you can use the following PL/SQL statements to retrieve valid task values:

- Budget at the top task level

```
SELECT task_id
, pm_task_reference
, task_number
, task_name
FROM pa_tasks
WHERE project_id = &project_id
AND parent_task_id IS NULL
```

- Budget at the lowest task level

```
SELECT task_id
, pm_task_reference
, task_number
, task_name
FROM pa_tasks tasks1
WHERE tasks1.project_id = &project_id
```

Oracle Proprietary, Confidential Information--Use Restricted by Contract

```

and not EXISTS (select NULL
                 from pa_tasks tasks2
                 where tasks1.project_id = tasks2.project_id
                 and tasks1.task_id = tasks2.parent_task_id)

```

- **Budget at the top and lowest task levels**

```

SELECT task_id
, pm_task_reference
, task_number
, task_name
, decode(nvl(parent_task_id,'1'),1,'Y','N') TOP_TASK
FROM pa_tasks tasks1
WHERE tasks1.project_id = &project_id
and not EXISTS (select NULL
                 from pa_tasks tasks2
                 where tasks1.project_id = tasks2.project_id
                 and tasks1.task_id = tasks2.parent_task_id)
or
   (tasks1.parent_task_id IS NULL
   and tasks1.project_id = &project_id )

```

Table 3 – 9 illustrates the relationship between certain selected values and budget line parameters. Pass only one of the two values:

<i>Selected Value</i>	<i>Budget Line Parameter</i>
TASK_ID	P_PA_TASK_ID
PM_TASK_REFERENCE	P_PM_TASK_REFERENCE

Table 3 – 9 How selected values relate to budget line parameters (Page 1 of 1)

► **Select resource list members (resources):**

If your budget entry method is categorized by resources and you have selected a resource list, the budget line should include the individual resources associated with the resource list. You can use the following PL/SQL statement to retrieve the resource list member information:

```

SELECT resource_list_member_id
, alias
, employee_first_name
, employee_last_name
FROM pa_query_res_list_members_v
WHERE resource_list_id = &resource_list_id

```

Table 3 – 10 illustrates the relationship between certain of the selected values and budget line parameters. Pass only one of the two values:

<i>Selected Value</i>	<i>Budget Line Parameter</i>
RESOURCE_LIST_MEMBER_ID	P_RESOURCE_LIST_MEMBER_ID
ALIAS	P_RESOURCE_ALIAS

Table 3 – 10 How selected values relate to budget line parameters (Page 1 of 1)

► **Select periods:**

How the budget entry method is time-phased affects which budget line parameters accept passed values, as shown in the following table:

Time-Phased By	Parameters That Accept Values (START_DATE, END_DATE, and PERIOD_NAME)
No Time-Phasing	None
Date Ranges	START_DATE and END_DATE
PA or GL Period	START_DATE and END_DATE or PERIOD_NAME

Table 3 – 11 How budget entry methods affect budget line parameters (Page 1 of 1)

When using time-phased budgeting, you can use the following PL/SQL statements to retrieve the appropriate date information:

- **Period name**

```
SELECT period_name
FROM pa_budget_periods_v
WHERE period_type_code = &time_phased_type_code
```

- **Begin and end dates**

```
SELECT period_start_date
, period_end_date
FROM pa_budget_periods_v
WHERE period_type_code = &time_phased_type_code
```

Table 3 – 12 illustrates the relationship between certain selected values and budget line parameters. You can pass a value for either the PERIOD_NAME or both the PERIOD_START_DATE and PERIOD_END_DATE:

<i>Selected Value</i>	<i>Budget Line Parameter</i>
PERIOD_NAME	P_PERIOD_NAME
PERIOD_START_DATE	P_BUDGET_START_DATE
PERIOD_END_DATE	P_BUDGET_END_DATE

Table 3 – 12 How selected values relate to budget line parameters (Page 1 of 1)

► **Select descriptions:**

You do not need to pass a description for budget lines.

Step 4 **Interface budget information to the server**

If your external system supports composite datatype parameters, such as Oracle PL/SQL Version 2.3 or higher, you can call the CREATE_DRAFT_BUDGET and UPDATE_BUDGET APIs directly.

Not all external systems can call the AMG APIs that use composite datatypes. Systems that do not support composite datatypes must call the supplementary Load-Execute-Fetch APIs. The Load-Execute-Fetch procedures include Initialize, Load, Execute, Fetch, and Clear categories. For more information, see API Procedures: page 4 – 2.

The following figure illustrates the flow of the Load-Execute-Fetch procedures for budget APIs. The process first calls the API INIT_BUDGET, which resets the server-side global PL/SQL tables that temporarily store the budget and budget line data. Once you set up these tables, use LOAD_BUDGET_LINE to move the budget and budget line data to the Oracle Projects database.

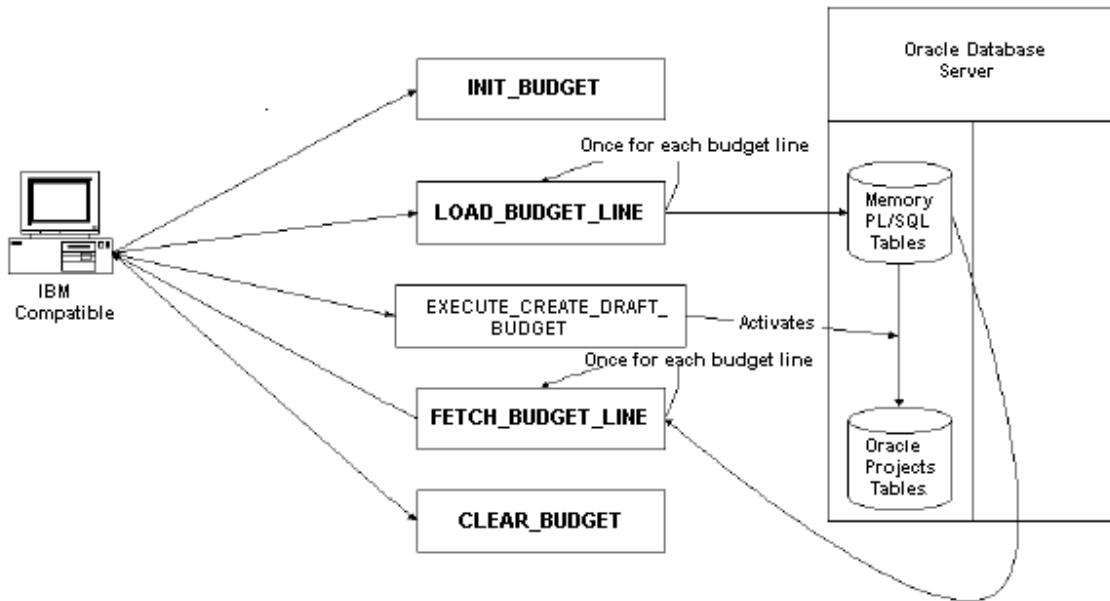


Figure 3-3: Load-Execute-Fetch Procedures for Budget APIs

Step 5 Start the server-side process

After the Load procedure successfully moves budget and budget line data to the Oracle Projects database, call the procedure API EXECUTE_CREATE_DRAFT_BUDGET to process the budget and budget line data in the global PL/SQL tables.

Step 6 Retrieve error messages

Each AMG API includes standard output parameters:

- P_RETURN_STATUS shows if the API was executed successfully.
- P_MSG_COUNT shows the number of errors detected during the execution of the API.

If the API detects one error, the API returns the error message text. If the API detects multiple errors, use GET_MESSAGES to retrieve the error messages. See GET_MESSAGES: page 2 - 18.

If the error relates to a budget line, use FETCH_BUDGET_LINE to identify the line causing the error. The API parameter P_LINE_RETURN_STATUS identifies the line by returning either E

Oracle Proprietary, Confidential Information--Use Restricted by Contract

(business rule violation) or U (unexpected error) for that line. (For more information about the return status, see Standard API Parameters: page 2 – 13. If you use `FETCH_BUDGET_LINE` for any other reason, it returns the error `NO_DATA_FOUND`.

Step 7 **Finish the Load–Execute–Fetch process**

After executing the Fetch procedures and retrieving any error messages, finish the Load–Execute–Fetch process by calling the API `CLEAR_BUDGET` and either saving or rolling back your changes to the database.

Using Agreement and Funding APIs

The following example describes how to create an interface between Oracle Projects and the agreement and funding information entered in your system. Depending on your company's business needs, your implementation of the project APIs may be more or less complex than the scenario shown here. As you work through the example, you may want to refer to information elsewhere in the manual.

- For a detailed description of agreement and funding APIs, see Agreement and Funding APIs: page 9 – 2.
- Most of the AMG APIs use a standard set of input and output parameters. For a description of these parameters, see Standard API Parameters: page 2 – 13.
- For an example of PL/SQL code for creating a project without using composite datatypes, see Creating a Project Using the Load-Execute-Fetch APIs: page A – 2.

Step 1 **Connect to an Oracle database**

To ensure that proper security is enforced while accessing Oracle Projects data, follow the steps in Security Requirements: page 2 – 8.

Step 2 **Collect agreement information**

Collect the following information to create an agreement in Oracle Projects:

- Agreement reference– Unique identifier of the Agreement.
- Customer– Valid customer in Oracle Projects.
- Agreement Type– Valid agreement type in Oracle Projects.
- Agreement Terms– Valid agreement terms in Oracle Projects.
- Owner of the agreement– Valid employee in Oracle Projects.

You can also use the following views to retrieve the list of values for collecting agreement information:

- PA_AGREEMENT_TYPE_LOV_V
- PA_TERMS_LOV_V
- PA_OWNED_BY_LOV_V
- PA_CUSTOMERS_LOV_V

Step 3 **Interface agreement information to the server**

Not all tools can call the AMG APIs that use composite datatypes. Tools that do not support composite datatypes must call the supplementary Load–Execute–Fetch APIs, also provided in the AMG. The Load–Execute–Fetch APIs include procedures to initialize, load, execute, fetch, and clear data. For more information, see View Definitions: page 4 – 2.

Use these APIs only if you use a tool that does not support composite data type parameters. If the tool (for example, Oracle PL/SQL Version 2.3 or higher) supports composite data type parameters, you can call the CREATE_AGREEMENT and ADD_FUNDING APIs directly. Following is the flow of the Load–Execute–Fetch Agreement and Funding procedures:

- Initialize Agreement (INIT_AGREEMENT)
- Load Agreement (LOAD_AGREEMENT)
- Load Funding (LOAD_FUNCING)
- Execute Create Agreement (EXECUTE_CREATE_AGREEMENT)
- Fetch Funding (Fetch Funding)
- Clear Agreement (CLEAR_AGREEMENT)

In the example above, INIT_AGREEMENT resets the server–side global PL/SQL tables that temporarily store the Agreement and Funding data. Once you set up these tables, you can use LOAD_AGREEMENT to move the Agreement data to the Oracle Projects database.

When you create a new agreement, this procedure must also pass parameters: P_PM_AGREEMENT_REFERENCE the unique reference code that identifies the agreement in the external system.

Step 4 **Interface funding information to the server**

After you interface the agreement–related data to the server, call LOAD_FUNDING to interface with the funding–related data to the server–side global PL/SQL tables. Call LOAD_FUNDING once for each funding in the agreement.



Attention: Each funding must specify at least the following information:

- Funding Reference (P_PM_FUNDING_REFERENCE): The unique reference code that identifies the funding in the external system.
- Agreement ID (P_AGREEMENT_ID): The identifier of the agreement for which the funding needs to be created.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- Project ID (P_PROJECT_ID): The identifier of the Project for which the funding needs to be created.
- Task ID (P_TASK_ID): For task-level funding, the identifier of the task for which the funding needs to be created.

Step 5 Start the server-side process

Once the Load procedures have successfully moved the agreement and funding data to the Oracle Projects global PL/SQL tables, call up the procedure EXECUTE_CREATE_AGREEMENT to process the agreement and funding data that you interfaced to the global PL/SQL tables. In addition to the standard input and output parameters, this Execute procedure requires the following parameters:

Input parameters

- P_PM_PRODUCT_CODE – The identification code of the product exporting the agreement . For information about setting up your product (external system) as a source, refer to Setting Up Your Product in Oracle Projects.

Output parameters

- P_AGREEMENT_ID – The unique Oracle Projects identification code for the new Agreement. .
- P_CUSTOMER_ID – The unique Oracle Projects customer id with which the agreement was created.

Step 6 Get return values for fundings

After the Load and Execute procedures create your agreement and funding in Oracle Projects, use FETCH_FUNDING to return each unique funding identification code from Oracle Projects.

The key input parameter for this procedure is P_FUNDING_INDEX, which points to a single funding, and the output parameters are P_FUNDING_ID and P_PM_FUNDING_REFERENCE.

To call the procedure for each funding, you can write a simple program to call FETCH_FUNDING in a loop with P_FUNDING_INDEX as the stepping variable (1 through the total number of funding). The output parameter P_RETURN_STATUS indicates whether the API handled the specific funding successfully (S). If the parameter returns E or U, the funding caused an error, and you must stop the Fetch procedure to retrieve the related error message. Fetch APIs do not return error message data. Instead, use GET_MESSAGES to retrieve the error text, as described in the next step.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Step 7 Retrieve error messages

Every AMG API includes two standard output parameters:

- P_RETURN_STATUS – indicates whether the API was executed successfully
- P_MSG_COUNT shows the number of errors detected during the execution of the API

If the API detects one error, the API returns the error message text. If the API detects multiple errors, use GET_MESSAGES to retrieve the error messages. See GET_MESSAGES: page 2 – 18.

Step 8 Finish the Load–Execute–Fetch process

After executing the Fetch procedures and retrieving any error messages, finish the Load–Execute–Fetch process by calling the API CLEAR_AGREEMENT and either save or roll back your changes to the database.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHAPTER

4

Detailed Design -- Overview

This chapter introduces the detailed view and procedure definitions.

Overview of Detailed Design

The Detailed Design sections following this overview describe how each API in the Activity Management Gateway is designed to work. Refer to these chapters to gain a thorough understanding of the underlying structure and processing of the AMG APIs.

For a discussion of standard AMG API parameters and shared APIs, see *Standard API Parameters: page 2 – 13* and *Common APIs: page 2 – 18*.

View Definitions

The Detailed Design sections define each API view in detail and describe each column and its characteristics. Characteristics include a description of datatypes and required values, as well as cross references showing you how each module accesses rows of a table. To view select statements, see *Oracle Projects Technical Reference Manual*.

API Procedures

The Detailed Design sections describe each PL/SQL procedure used to perform certain functions in Oracle Projects based on the information you maintain in your external system.

Some APIs use composite datatypes, such as records or tables of records, as input and output parameters. Composite datatypes are PL/SQL 2.3 features that are available with Oracle 7.3.2. For more information about composite datatypes, see *APIs That Use Composite Datatypes: page 2 – 13*.

Tools and products that cannot use composite datatypes must call supplementary Load-Execute-Fetch APIs instead. The Load-Execute-Fetch APIs were designed without composite datatype parameters for compatibility with any tool and perform the following functions:

- Accept parameters with standard datatypes (VARCHAR2, NUMBER, and DATE) as IN parameters
- Load global composite type structures (records and tables)
- Call the underlying business object APIs (passing the global structures as IN parameters)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- Read the results from a global message and results table
- Pass the message back to the calling programs upon demand (the calling program fetches each message separately)

Call the procedures in this order:

1. **Initialize.** This step initializes the global data structures.
2. **Load.** This function loads IN parameter PL/SQL tables and records. Repeat this step until all the input structures are populated.
3. **Execute.** This step calls a business object API cover that calls the business object API. The business object API uses the global structures that were populated during the Load procedure.
4. **Fetch.** This procedure fetches one output value at a time for a business object. It also fetches messages. The calling program may or may not call the Fetch procedure, depending on the function performed.
5. **Clear.** This step clears the global structures and resets any global counters used in the calling program.

Check Procedures

Chapter 5 also includes detailed descriptions of the PL/SQL procedures used to verify in real-time that:

- Project and task information you have entered into your external system is unique in Oracle Projects
- Certain functions, such as deleting a project or task, follow the business rules defined in Oracle Projects

CHAPTER

5

Detailed Design -- Project APIs

This chapter describes how to implement AMG APIs for project and task information, and provides detailed view and procedure definitions.

Project APIs

Develop a detailed project plan using the external system you prefer. Then you can use the project APIs to push your plan into Oracle Projects and create a project based on the information in your plan. As your project plan evolves, update project information in your external system and then periodically synchronize the two systems. The project APIs update the task information and work breakdown structures (WBS) in Oracle Projects to reflect your changes.

Note: When you call any project API that requires a project identifier, you must identify the project by passing either the P_PA_PROJECT_ID or the P_PM_PROJECT_REFERENCE parameter. When you call any project API that requires a task identifier, you must identify the task by passing either the P_PA_TASK_ID or the P_PM_TASK_REFERENCE parameter.

For an overview of the AMG views and APIs, see Overview of Detailed Design: page 4 – 2.

The views and procedures discussed in this section are listed below. The procedures are located in the public API package PA_PROJECT_PUB.

- Views
 - PA_CLASS_CATEGORIES_LOV_V: page 5 – 4
 - PA_CUSTOMERS_LOV_V: page 5 – 4
 - PA_DISTRIBUTION_RULES_LOV_V: page 5 – 4
 - PA_KEY_MEMBERS_LOV_V: page 5 – 5
 - PA_ORGANIZATIONS_LOV_V: page 5 – 5
 - PA_OVERRIDE_FIELDS_V: page 5 – 6
 - PA_OVERRIDE_FIELD_VALUES_V: page 5 – 8
 - PA_PROJECT_STATUS_LOV_V: page 5 – 8
 - PA_PROJECTS_AMG_V: page 5 – 9
 - PA_SELECT_TEMPLATE_V: page 5 – 14
 - PA_SERVICE_TYPE_LOV_V: page 5 – 15
 - PA_TASK MANAGERS_LOV_V: page 5 – 15
 - PA_TASKS_AMG_V: page 5 – 17
- Project and Task Procedures
 - ADD_TASK: page 5 – 22

- CREATE_PROJECT: page 5 – 26
- DELETE_PROJECT: page 5 – 37
- DELETE_TASK: page 5 – 38
- UPDATE_PROJECT: page 5 – 40
- UPDATE_TASK: page 5 – 45
- Load-Execute-Fetch Procedures
 - CLEAR_PROJECT: page 5 – 51
 - EXECUTE_CREATE_PROJECT: page 5 – 51
 - EXECUTE_UPDATE_PROJECT: page 5 – 52
 - FETCH_TASK: page 5 – 52
 - INIT_PROJECT: page 5 – 53
 - LOAD_CLASS_CATEGORY: page 5 – 53
 - LOAD_KEY_MEMBER: page 5 – 53
 - LOAD_PROJECT: page 5 – 54
 - LOAD_TASK: page 5 – 58
- Check Procedures
 - CHECK_ADD_SUBTASK_OK: page 5 – 61
 - CHECK_CHANGE_PARENT_OK: page 5 – 62
 - CHECK_CHANGE_PROJECT_ORG_OK: page 5 – 63
 - CHECK_DELETE_PROJECT_OK: page 5 – 63
 - CHECK_DELETE_TASK_OK: page 5 – 64
 - CHECK_TASK_NUMBER_CHANGE_OK: page 5 – 65
 - CHECK_UNIQUE_PROJECT_REFERENCE: page 5 – 65
 - CHECK_UNIQUE_TASK_NUMBER: page 5 – 66
 - CHECK_UNIQUE_TASK_REFERENCE: page 5 – 66

View Definitions

PA_CLASS_CATEGORIES_LOV_V

PA_CLASS_CATEGORIES_LOV_V is a view that retrieves valid class codes set up in Oracle Projects.

Note: Use the value in the DISPLAY_NAME field (retrieved by the PA_OVERRIDE_FIELDS_V view) to show only those class codes associated with a specific class category. For example,

```
SELECT code
       description
FROM   pa_class_categories_lov_v
WHERE  class_category = 'Funding Source';
```

Column descriptons for PA_CLASS_CATEGORIES_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CLASS_CATEGORY	NOT NULL	VARCHAR2(30)	
CODE	NOT NULL	VARCHAR2(30)	
DESCRIPTION		VARCHAR2(250)	

PA_CUSTOMERS_LOV_V

PA_CUSTOMERS_LOV_V is a view that retrieves all valid customers defined in or used by Oracle Projects.

Column descriptons for PA_CUSTOMERS_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	NUMBER(15)	
DESCRIPTION	NOT NULL	VARCHAR2(50)	
CUSTOMER_NUMBER	NOT NULL	VARCHAR2(30)	

PA_DISTRIBUTION_RULES_LOV_V

PA_DISTRIBUTION_RULES_LOV_V is a view that retrieves all valid revenue distribution rules defined within Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column descriptons for PA_DISTRIBUTION_RULES_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(20)	
DESCRIPTION	NOT NULL	VARCHAR2(80)	
PROJECT_TYPE	NOT NULL	VARCHAR2(20)	
SOURCE_TEMPLATE_ID	NOT NULL	NUMBER	

PA_KEY_MEMBERS_LOV_V

PA_KEY_MEMBERS_LOV_V is a view that retrieves the names and employee identification numbers of all valid key members from Oracle Projects.

Note: PA_EMPLOYEES returns all the valid employees defined in Oracle Projects.

Column descriptons for PA_KEY_MEMBERS_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	NUMBER(9)	
DESCRIPTION	NOT NULL	VARCHAR2(240)	

PA_ORGANIZATIONS_LOV_V

PA_ORGANIZATIONS_LOV_V is a view that retrieves the names of all valid organizations defined in Oracle Projects.

Column descriptons for PA_ORGANIZATIONS_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	NUMBER(15)	
BUSINESS_GROUP_ID	NOT NULL	NUMBER(15)	
DESCRIPTION	NOT NULL	VARCHAR2(60)	
TYPE		VARCHAR2(30)	

PA_OVERRIDE_FIELDS_V

PA_OVERRIDE_FIELDS_V is a view that retrieves the prompts for all Quick Entry fields associated with a particular template.

Column descriptons for PA_OVERRIDE_FIELDS_V are shown below.

Column Descriptions

Name	Null?	Type	Description
PA_SOURCE_TEMPLATE_ID	NOT NULL	NUMBER	The system number that uniquely identifies the template or project to be used as a basis for the new project
PA_FIELD_NAME	NOT NULL	VARCHAR2(30)	The unique name of the Quick Entry field
PA_DISPLAY_NAME	NOT NULL	VARCHAR2(30)	The name displayed in the Quick Entry window in Oracle Projects
TYPE		VARCHAR2(30)	A subtype of the value displayed in the field 'field name'
SORT_ORDER		NUMBER	Determines the sorting order in the Quick Entry window in Oracle Projects
MANDATORY_FLAG		VARCHAR2(1)	Indicates whether a value is required for the field
LOV_VIEW_NAME		VARCHAR2(50)	The name of the view that retrieves valid data for this particular field

The following table shows the contents of the view PA_OVERRIDE_FIELDS_V for a project with a project identification code of 1020 and all quick entry fields enabled. The column labeled "M" indicates whether a value is required in the quick entry field:

ID	Field Name	Display Name	Type	Order	M	View Name
1020	NAME	Project Name		20	Y	
1020	DESCRIPTION	Project Description		30	N	
1020	START_DATE	Project Start Date		40	N	
1020	COMPLETION_DATE	Project Completion Date		50	N	
1020	PROJECT_STATUS_CODE	Project Status		60	N	PA_PROJECT_STATUS_LOV_V
1020	PUBLIC_SECTOR_FLAG	Public Sector		70	N	
1020	DISTRIBUTION_RULE	Distribution Rule		80	N	PA_DISTRIBUTION_RULES_LOV_V
1020	CARRYING_OUT_ORGANIZATION_ID	Organization		90	N	PA_ORGANIZATIONS_LOV_V

Table 5 - 1 Example of PA_OVERRIDE_FIELDS_V for a project with all quick entry fields enabled (Page 1 of 2)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

ID	Field Name	Display Name	Type	Order	M	View Name
1020	KEY_MEMBER	Project Manager	PROJECT MANAGER	100	Y	PA_KEY_MEMBERS_LOV_V
1020	KEY_MEMBER	Project Coordinator	Project Coordinator	110	N	PA_KEY_MEMBERS_LOV_V
1020	CLASSIFICATION	Funding Source	Funding Source	120	Y	PA_CLASS_CATEGORIES_LOV_V
1020	CLASSIFICATION	Market Sector	Market Sector	130	N	PA_CLASS_CATEGORIES_LOV_V
1020	CUSTOMER_NAME	Customer Name	PRIMARY	140	N	PA_CUSTOMERS_LOV_V

Table 5 - 1 Example of PA_OVERRIDE_FIELDS_V for a project with all quick entry fields enabled (Page 2 of 2)

The views you use to select valid values all have CODE and DESCRIPTION columns. Use these two columns and the value of the field LOV_VIEW_NAME to retrieve the valid values for any Quick Entry field. Valid values are stored in the CODE field. The table below shows the valid values of the quick entry fields.

Quick Entry Fields	Valid Values
NAME	
CARRYING_OUT_ORGANIZATION_ID	PA_ORGANIZATIONS_LOV_V
PUBLIC_SECTOR_FLAG	'Y' or 'N'
PROJECT_STATUS_CODE	PA_PROJECT_STATUS_LOV_V
DESCRIPTION	
START_DATE	DD-MON-YY format (e.g., 10-SEP-68)
COMPLETION_DATE	DD-MON-YY format (e.g., 13-JUL-94)
DISTRIBUTION_RULE	PA_DISTRIBUTION_RULES_LOV_V
CUSTOMER_ID	PA_CUSTOMERS_LOV_V (currently, the default CUSTOMER_RELATIONSHIP_CODE is PRIMARY; no other value is accepted)

Table 5 - 2 Valid values for quick entry fields (Page 1 of 2)

Quick Entry Fields	Valid Values
KEY_MEMBERS (multiple)	PA_KEY_MEMBERS_LOV_V
CLASS_CATEGORIES (multiple)	PA_CLASS_CATEGORIES_LOV_V

Table 5 - 2 Valid values for quick entry fields (Page 2 of 2)

PA_OVERRIDE_FIELD_VALUES_V

PA_OVERRIDE_FIELD_VALUES_V is a view that retrieves the values passed to the Quick Entry fields while creating a project.

Column descriptors for PA_OVERRIDE_FIELD_VALUES_V are shown below.

Column Descriptions

Name	Null?	Type	Description
PA_PROJECT_ID	NOT NULL	NUMBER	The reference code that uniquely identifies a specific template or project
PA_FIELD_NAME	NOT NULL	VARCHAR2(30)	The unique name of the Quick Entry field
PA_DISPLAY_NAME	NOT NULL	VARCHAR2(30)	The name displayed in the Quick Entry window in Oracle Projects
SORT_ORDER		NUMBER	Determines the sorting order in the Quick Entry window in Oracle Projects
TYPE		VARCHAR2(30)	A subtype of the value displayed in the field 'field name'
MANDATORY_FLAG		VARCHAR2(1)	Indicates whether a value is required for the field
LOV_VIEW_NAME		VARCHAR2(50)	The name of the view that retrieves valid data for this field
CODE			Contains the codes, such as ORGANIZATION_ID, CUSTOMER_ID, AND PERSON_ID
FIELD_VALUE			Contains the field value, such as organization name, customer name, and employee name

PA_PROJECT_STATUS_LOV_V

PA_PROJECT_STATUS_LOV_V is a view that retrieves all valid project statuses set up in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column descriptons for PA_PROJECT_STATUS_LOV_V are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	
DESCRIPTION		VARCHAR2(80)	

PA_PROJECTS_AMG_V

This view retrieves information about all valid projects for the organization associated with the user's responsibility.

Column descriptons for PA_PROJECTS_AMG_V are shown below.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The system-generated number that uniquely identifies the project
NAME	NOT NULL	VARCHAR2(30)	User-defined name that uniquely identifies the project
SEGMENT1	NOT NULL	VARCHAR2(25)	The project number that uniquely identifies the project. The project number can be user-entered or system-generated as defined in the implementation options
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who Column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who Column
CREATION_DATE	NOT NULL	DATE	Standard Who Column
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who Column
LAST_UPDATE_LOGIN	NOT NULL	NUMBER(15)	Standard Who Column
PROJECT_TYPE	NOT NULL	VARCHAR2(20)	The project type that classifies the project and defaults project information upon project entry
CARRYING_OUT_ORGANIZATION_ID	NOT NULL	NUMBER(15)	The identifier of the organization that is responsible for the project work
PUBLIC_SECTOR_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates whether the project is in the public sector or the private sector
PROJECT_STATUS_CODE	NOT NULL	VARCHAR2(30)	The status of the project.
DESCRIPTION	NULL	VARCHAR2(250)	The description of the project
START_DATE	NULL	DATE	The date on which the project starts; expenditure items with item dates before the start date cannot be entered for the project
COMPLETION_DATE	NULL	DATE	The date on which the project is completed; expenditure items with item dates after the completion date cannot be entered for the project

CLOSED_DATE	NULL DATE	The date that the project was closed by changing the project status to Closed
DISTRIBUTION_RULE	NULL VARCHAR2(30)	The distribution rule that specifies the contract project's revenue accrual and billing method
LABOR_INVOICE_FORMAT_ID	NULL NUMBER(15)	The identifier of the invoice format used to group and format labor expenditure items into invoice lines
NON_LABOR_INVOICE_FORMAT_ID	NULL NUMBER(15)	The identifier of the invoice format used to group and format non-labor expenditure items into invoice lines
RETENTION_INVOICE_FORMAT_ID	NULL NUMBER(15)	The identifier of the invoice format used to format a retention line on the project's invoices
RETENTION_PERCENTAGE	NULL NUMBER(15)	The percentage to be retained on each invoice generated for the project
BILLING_OFFSET	NULL NUMBER(15)	The number of days after the project start date when the first invoice for the project is created
BILLING_CYCLE	NULL NUMBER(15)	Obsolete
BILLING_CYCLE_ID	NULL NUMBER(15)	The identifier of the billing cycle associated with the project
LABOR_STD_BILL_RATE_SCHDL	NULL VARCHAR2(20)	The labor standard bill rate schedule that defaults to new tasks created for the project
LABOR_BILL_RATE_ORG_ID	NULL NUMBER(15)	The identifier of the organization that owns the labor standard bill rate schedule
LABOR_SCHEDULE_FIXED_DATE	NULL DATE	The fixed date used to determine the effective bill rates of the standard labor bill rate schedule. The project fixed date is a default for the task fixed date
LABOR_SCHEDULE_DISCOUNT	NULL NUMBER(7,4)	The percentage to be discounted from the standard labor bill rate schedule. The project discount is a default for the task discount
TEMPLATE_FLAG	NULL VARCHAR2(1)	Indicates whether the project is a template or not
VERIFICATION_DATE	NULL DATE	Not currently used
CREATED_FROM_PROJECT_ID	NULL NUMBER(15)	Holds the Project_id of the source project. This value would be null for templates
TEMPLATE_START_DATE_ACTIVE	NULL DATE	The Effective start date of the Template
NON_LABOR_STD_BILL_RATE_SCHDL	NULL VARCHAR2(30)	The non-labor standard bill rate schedule that defaults to new tasks created for the project
TEMPLATE_END_DATE_ACTIVE	NULL DATE	Last effective date of the template
NON_LABOR_BILL_RATE_ORG_ID	NULL NUMBER(15)	The identifier of the organization that owns the non-labor standard bill rate schedule

Oracle Proprietary, Confidential Information--Use Restricted by Contract

NON_LABOR_SCHEDULE_FIXED_DATE	NULL	DATE	The fixed date used to determine the effective bill rates of the standard non-labor bill rate schedule. The project fixed date is a default for the task fixed date
NON_LABOR_SCHEDULE_DISCOUNT	NULL	NUMBER(7,4)	The percentage to be discounted from the standard non-labor bill rate schedule. The project discount is a default for the task discount
LIMIT_TO_TXN_CONTROLS_FLAG	NULL	VARCHAR2(1)	Flag that indicates if users can only charge expenditures to the project that are listed in the transaction controls. If flag =Y, then only items listed in the controls are allowed. If flag =N or null, then items not listed are allowed
PROJECT_LEVEL_FUNDING_FLAG	NULL	VARCHAR2(1)	Flag that indicates if the project is funded at the project level ('Y') or at the top task level ('N'). This flag is set when the project is baselined
INVOICE_COMMENT	NULL	VARCHAR2(240)	Free text comment that can be printed on an invoice. The invoice comment defined for the project is defaulted to the invoice comment for the draft invoices of the project
UNBILLED_RECEIVABLE_DR	NULL	NUMBER(22,5)	The unbilled receivables balance for the project that is posted to Oracle General Ledger or Oracle Receivables. This value is maintained by the Transfer Revenue and Transfer Invoices processes
UNEARNED_REVENUE_CR	NULL	NUMBER(22,5)	The unearned revenue balance for the project that is posted to Oracle General Ledger or Oracle Receivables. This value is maintained by the Transfer Revenue and Transfer Invoices processes
REQUEST_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who Column
SUMMARY_FLAG	NOT NULL	VARCHAR2(1)	Flag used with the project key flexfield. The project key flexfield is not implemented
ENABLED_FLAG	NOT NULL	VARCHAR2(1)	Flag used with the key flexfield. The project key flexfield is not implemented
SEGMENT2	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT3	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT4	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT5	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT6	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT7	NULL	VARCHAR2(25)	Key flexfield segment - not implemented

Oracle Proprietary, Confidential Information--Use Restricted by Contract

SEGMENT8	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT9	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
SEGMENT10	NULL	VARCHAR2(25)	Key flexfield segment - not implemented
ATTRIBUTE_CATEGORY	NULL	VARCHAR2(30)	Descriptive flexfield context field
ATTRIBUTE1	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE2	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE3	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE4	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE5	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE6	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE7	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE8	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE9	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE10	NULL	VARCHAR2(150)	Descriptive flexfield segment
COST_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default costing burden schedule
REV_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default revenue burden schedule
INV_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default invoice burden schedule
COST_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm costing burden schedule
REV_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm revenue burden schedule
INV_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm invoice burden schedule
LABOR_SCH_TYPE	NULL	VARCHAR2(1)	The schedule type of labor expenditure items
NON_LABOR_SCH_TYPE	NULL	VARCHAR2(1)	The schedule type of non-labor expenditure items
OVR_COST_IND_RATE_SCH_ID	NULL	NUMBER(15)	This column is not used
OVR_REV_IND_RATE_SCH_ID	NULL	NUMBER(15)	This column is not used
OVR_INV_IND_RATE_SCH_ID	NULL	NUMBER(15)	This column is not used
ORG_ID	NULL	NUMBER(15)	Operating unit identifier for multi-organization installations
PM_PRODUCT_CODE	NULL	VARCHAR2(30)	The identifier of the external project management system from which the project was imported.
PM_PROJECT_REFERENCE	NULL	VARCHAR2(25)	The identifier of the project in the external project management system from which the project was imported.
ACTUAL_START_DATE	NULL	DATE	The actual start date of the project. Applicable only for a project that has originated from an external system.
ACTUAL_FINISH_DATE	NULL	DATE	The actual end date of the project. Applicable only for a project that has originated from an external system.
EARLY_START_DATE	NULL	DATE	The early start date of the project. Applicable only for a project that has originated from an external system.
EARLY_FINISH_DATE	NULL	DATE	The early finish date of the project. Applicable only for a project that has originated from an external system.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

LATE_START_DATE	NULL	DATE	The late start date of the project. Applicable only for a project that has originated from an external system.
LATE_FINISH_DATE	NULL	DATE	The late finish date of the project. Applicable only for a project that has originated from an external system.
SCHEDULED_START_DATE	NULL	DATE	The scheduled start date of the project. Applicable only for a project that has originated from an external system.
SCHEDULED_FINISH_DATE	NULL	DATE	The scheduled finish date of the project. Applicable only for a project that has originated from an external system.
ADW_NOTIFY_FLAG	NULL	VARCHAR2(1)	The flag that indicates whether this row needs to be sent to the interface table or not. This column is required for Oracle Project Analysis Collection Pack.
WF_STATUS_CODE	NULL	VARCHAR2(30)	Indicates the status of the workflow associated with the project
PROJECT_CURRENCY_CODE	NOT NULL	VARCHAR2(15)	Currency code of the project
ALLOW_CROSS_CHARGE_FLAG	NOT NULL	VARCHAR2(1)	Flag to indicate whether cross charge is allowed
PROJECT_RATE_DATE	NULL	DATE	Default value for project rate date, which can be overridden at task level
PROJECT_RATE_TYPE	NULL	VARCHAR2(30)	Default value for project rate type, which can be overridden at task level
OUTPUT_TAX_CODE	NULL	VARCHAR2(30)	Indicates whether tax rate defined for the Project will be used for customer invoices.
RETENTION_TAX_CODE	NULL	VARCHAR2(30)	Indicates whether tax rate defined for retention will be used for customer invoices.
CC_PROCESS_LABOR_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates whether cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This defaults to a project from the project template.
LABOR_TP_SCHEDULE_ID	NOT NULL	NUMBER	Identifier for transfer price schedule to use for cross charged labor transactions. This defaults to a project from the project template.
LABOR_TP_FIXED_DATE	NULL	DATE	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CC_PROCESS_NL_FLAG	NOT NULL	VARCHAR2(1)	for labor transactions. This defaults to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor_flag is 'Y', this field is required.
NL_TP_SCHEDULE_ID	NOT NULL	NUMBER	Flag that indicates whether cross charge processing is to be performed for non-labor transactions charged to the project. Default value for the project template is 'N'. This defaults to a project from the project template.
NL_TP_FIXED_DATE	NULL	DATE	Identifier for transfer price schedule to use for cross charged non-transactions. This defaults to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required.
CC_TAX_TASK_ID	NULL	NUMBER	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This defaults to a project from the project template. This value for the project is a default for the task fixed date.
			Identifier of the task to which intercompany tax items on the intercompany AP invoice are charged.

PA_SELECT_TEMPLATE_V

PA_SELECT_TEMPLATE_V is a view that selects all valid project templates and projects created in Oracle Projects.

Column descriptons for PA_SELECT_TEMPLATE_V are shown below.

Column Descriptions

Name	Null?	Type	Description
PA_SOURCE_TEMPLATE_ID	NOT NULL	NUMBER(15)	The system number that uniquely identifies the template or project to be used as a basis for the new project

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	A project code that uniquely identifies the project
PA_PROJECT_NAME	NOT NULL	VARCHAR2(30)	User-defined name that uniquely identifies the project
TEMPLATE_FLAG		VARCHAR2(1)	Indicates whether or not the project is a template
CREATED_FROM_PROJECT_ID		NUMBER(15)	This number uniquely identifies the template from which this project originated, and is NULL when project the P_TEMPLATE_FLAG = TRUE
CARRYING_OUT_ORGANIZATION_ID		NUMBER(15)	The identification code of the project organization
DISTRIBUTION_RULE		VARCHAR2(30)	The distribution rule assigned to this template
TEMPLATE_START_DATE_ACTIVE		DATE	The date from which the template is active
TEMPLATE_END_DATE_ACTIVE		DATE	The date through which the template is active
DEFAULT_RESOURCE_LIST_ID		NUMBER(15)	The identification code of the default resource list assigned to this template

PA_SERVICE_TYPE_LOV_V

Because valid service type codes need to be selected for the parameter `SERVICE_TYPE_CODE`, you can use the `PA_SERVICE_TYPE_LOV_V` view to retrieve valid codes from Oracle Projects and display them in your external system.

Column descriptons for `PA_SERVICE_TYPE_LOV_V` are shown below.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	
DESCRIPTION	NOT NULL	VARCHAR2(80)	

PA_TASK MANAGERS_LOV_V

Because valid employees must be selected for the parameter `TASK_MANAGER_PERSON_ID`, you can use the `PA_TASK MANAGERS_LOV_V` view to retrieve valid employees from Oracle Projects and display them in your external system.

Column descriptons for `PA_TASK MANAGERS_LOV_V` are shown below.

Column Descriptions

Name	Null?	Type	Description
PERSON_ID	NOT NULL	NUMBER(9)	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

EMPLOYEE_NUMBER
FULL_NAME

VARCHAR2(30)
VARCHAR2(240)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_TASKS_AMG_V

This view retrieves information about all valid tasks for the organization associated with the user's responsibility.

Column descriptions for PA_TASKS_AMG_V are shown below.

Column Descriptions

Name	Null?	Type	Description
ROW_ID	NULL	ROWID	The row ID of the task in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The system-generated number that uniquely identifies the task
PROJECT_ID	NOT NULL	NUMBER(15)	The identifier of the project for which the task is a unit of work
TASK_NUMBER	NOT NULL	VARCHAR2(25)	User-defined number that uniquely identifies the task within a project. It is recommended to number tasks based on the wbs since that is how tasks are ordered in some reports. However note: task numbers do not record the wbs structure
CREATION_DATE	NOT NULL	DATE	Standard Who Column
CREATED_BY	NOT NULL	NUMBER(15)	Standard Who Column
LAST_UPDATE_DATE	NOT NULL	DATE	Standard Who Column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Standard Who Column
LAST_UPDATE_LOGIN	NOT NULL	NUMBER(15)	Standard Who Column
TASK_NAME	NOT NULL	VARCHAR2(20)	User-defined short name of the task
TOP_TASK_ID	NOT NULL	NUMBER(15)	The identifier of the top task to which this task rolls up. If the task is a top task, the top_task_id is set to its own task_id.
WBS_LEVEL	NOT NULL	NUMBER(3)	The level of the task in the work breakdown structure
READY_TO_BILL_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates whether the task is authorized to be invoiced. The flag is set only for top tasks, defaults to 'Y' upon creation of the task and is set to 'Y' or 'N' in the Control Revenue and Billing by Top Task form
READY_TO_DISTRIBUTE_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates whether the task is authorized for revenue accrual. The flag is set only for top tasks, defaults to 'Y' upon creation of the task, and is set to 'Y' or 'N' in the Control Revenue and Billing by Top Task form
PARENT_TASK_ID	NULL	NUMBER(15)	The identifier of the task that is the parent of the task in the project work breakdown structure
DESCRIPTION	NULL	VARCHAR2(250)	Description of the task

CARRYING_OUT_ORGANIZATION_ID	NOT NULL	NUMBER(15)	The identifier of the organization that is responsible for the task work. The project organization is defaulted to the task organization upon creation of the task
SERVICE_TYPE_CODE	NULL	VARCHAR2(30)	The type of work performed on the task
TASK_MANAGER_PERSON_ID	NULL	NUMBER(9)	The identifier of the employee that manages the task.
CHARGEABLE_FLAG	NULL	VARCHAR2(1)	Flag that indicates if expenditure items can be charged to the task. Only lowest level tasks can be chargeable
BILLABLE_FLAG	NULL	VARCHAR2(1)	Default flag for items charged to the task that indicates if the item can accrue revenue (Y or N). For capital projects this flag is used as capitalizable_flag. For indirect projects this flag is set to N and is not used.
LIMIT_TO_TXN_CONTROLS_FLAG	NULL	VARCHAR2(1)	Flag that indicates if users can only charge expenditures to the task that are listed in task's transaction controls. If flag =Y, then only items listed in controls are allowed. If flag = 'N' or null, then items not listed are allowed
START_DATE	NULL	DATE	The date on which the task starts; expenditure items with item dates before the start date cannot be entered for the task
COMPLETION_DATE	NULL	DATE	The date on which the task is completed; expenditure items with item dates after the task completion date cannot be entered for the task. You must enter a start date to enter a completion date.
ADDRESS_ID	NULL	NUMBER(15)	The identifier of the customer address that is the task work site. This value is defaulted from the project customer work site if only one project customer exists
LABOR_BILL_RATE_ORG_ID	NULL	NUMBER(15)	The identifier of the organization that owns the labor standard bill rate schedule
LABOR_STD_BILL_RATE_SCHDL	NULL	VARCHAR2(20)	The labor standard bill rate schedule that is used to calculate revenue for labor expenditure items charged to the task
LABOR_SCHEDULE_FIXED_DATE	NULL	DATE	The date used to determine the effective bill rates of the task standard labor bill rate schedule. If no fixed date is entered, the expenditure item date is used to determine the effective bill rate for the item

Oracle Proprietary, Confidential Information--Use Restricted by Contract

LABOR_SCHEDULE_DISCOUNT	NULL	NUMBER(7,4)	The percentage to be discounted from the task standard labor bill rate schedule
NON_LABOR_BILL_RATE_ORG_ID	NULL	NUMBER(15)	The identifier of the organization that owns the non-labor standard bill rate schedule
NON_LABOR_STD_BILL_RATE_SCHDL	NULL	VARCHAR2(30)	The non-labor standard bill rate schedule that is used to calculate revenue for non-labor expenditure items charged to the task
NON_LABOR_SCHEDULE_FIXED_DATE	NULL	DATE	The fixed date used to determine the effective bill rates of the standard non-labor bill rate schedule. If no fixed date is entered, the expenditure item date is used to determine the effective bill rate of the item
NON_LABOR_SCHEDULE_DISCOUNT	NULL	NUMBER(7,4)	The percentage to be discounted from the task standard non-labor bill rate schedule
LABOR_COST_MULTIPLIER_NAME	NULL	VARCHAR2(20)	The labor cost multiplier defined for the task of a premium project. The labor cost multiplier is populated for all overtime expenditure items charged to the task upon manual entry of the items or in the Overtime Calculation program
REQUEST_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_APPLICATION_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_ID	NULL	NUMBER(15)	Standard Who Column
PROGRAM_UPDATE_DATE	NULL	DATE	Standard Who Column
ATTRIBUTE_CATEGORY	NULL	VARCHAR2(30)	Descriptive flexfield context field
ATTRIBUTE1	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE2	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE3	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE4	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE5	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE6	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE7	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE8	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE9	NULL	VARCHAR2(150)	Descriptive flexfield segment
ATTRIBUTE10	NULL	VARCHAR2(150)	Descriptive flexfield segment
COST_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default costing burden schedule
REV_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default revenue burden schedule
INV_IND_RATE_SCH_ID	NULL	NUMBER(15)	The identifier of default invoice burden schedule
COST_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm costing burden schedule
REV_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm revenue burden schedule
INV_IND_SCH_FIXED_DATE	NULL	DATE	The schedule fixed date of firm invoice burden schedule
LABOR_SCH_TYPE	NULL	VARCHAR2(1)	The schedule type of labor expenditure items
NON_LABOR_SCH_TYPE	NULL	VARCHAR2(1)	The schedule type of non-labor expenditure items

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PM_TASK_REFERENCE	NULL	VARCHAR2(25)	The identifier of the task in the external project management system from which the budget was imported.
PM_PRODUCT_CODE	NULL	VARCHAR2(30)	The identifier of the external project management system from which the task was imported.
CHILD_INDICATOR	NULL	VARCHAR2(1)	Indicates if the task has a child task.
ACTUAL_START_DATE	NULL	DATE	The actual start date of the project. Applicable only for a project that has originated from an external system.
ACTUAL_FINISH_DATE	NULL	DATE	The actual end date of the project. Applicable only for a project that has originated from an external system.
EARLY_START_DATE	NULL	DATE	The early start date of the project. Applicable only for a project that has originated from an external system.
EARLY_FINISH_DATE	NULL	DATE	The early finish date of the project. Applicable only for a project that has originated from an external system.
LATE_START_DATE	NULL	DATE	The late start date of the project. Applicable only for a project that has originated from an external system.
LATE_FINISH_DATE	NULL	DATE	The late finish date of the project. Applicable only for a project that has originated from an external system.
SCHEDULED_START_DATE	NULL	DATE	The scheduled start date of the project. Applicable only for a project that has originated from an external system.
SCHEDULED_FINISH_DATE	NULL	DATE	The scheduled finish date of the project. Applicable only for a project that has originated from an external system.
ALLOW_CROSS_CHARGE_FLAG	NOT NULL	VARCHAR2(1)	Flag to indicate whether cross charge are allowed
PROJECT_RATE_DATE	NULL	DATE	Task level default value for project rate date
PROJECT_RATE_TYPE	NULL	VARCHAR2(30)	Task level default value for project rate type
CC_PROCESS_LABOR_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is N. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date
LABOR_TP_SCHEDULE_ID	NOT NULL	NUMBER	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date. If cc_process_labor_flag is set to Y, this field is required

Oracle Proprietary, Confidential Information--Use Restricted by Contract

LABOR_TP_FIXED_DATE	NULL	DATE	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date
CC_PROCESS_NL_FLAG	NOT NULL	VARCHAR2(1)	Flag that indicates cross charge processing is to be performed for non-labor transactions charged to the project. Default value for the project template is N. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date
NL_TP_SCHEDULE_ID	NOT NULL	NUMBER	Identifier for transfer price schedule to use for cross charged non-labor transactions. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date. If cc_process_nl_flag is set to Y, this field is required
NL_TP_FIXED_DATE	NULL	DATE	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. This value for the project is a default for the task fixed date
RECEIVE_PROJECT_INVOICE_FLAG	NULL	VARCHAR2(1)	Flag that indicates that the task may receive charges from internal supplies via inter-project billing

Procedure Definitions

ADD_TASK

ADD_TASK is a PL/SQL procedure used to add new subtasks to a task of a project in Oracle Projects. We replaced the task record type with a parameter with a standard datatype (NUMBER, VARCHAR2, or DATE) for every field in the record type definition so you can call this procedure directly.

Business Rules (task level)

Oracle Projects imposes the following task-level business rules:

- Each new task must have a unique number within a given project. You can use the Check procedure CHECK_UNIQUE_TASK_NUMBER to verify that the new task number does not already exist in your project.
 - You cannot create a subtask for any project if the parent task has any of the following attributes:
 - Transaction controls
 - Burden schedule overrides
 - A budget
 - A percentage complete value
 - An asset
 - An expenditure item
 - A purchase order distribution
 - A purchase order requisition
 - An Oracle Payables invoice
 - An Oracle Payables invoice distribution
- Note:** You can use the Check procedure CHECK_ADD_SUBTASK_OK to verify that you can add a subtask to a particular parent task.
- For contract projects, you cannot add a subtask to a parent task that has any of the following attributes:
 - Labor cost multiplier

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- Job bill rate override
- Employee bill rate override
- Labor multiplier
- Non-labor bill rate override
- Job bill title override
- Job assignment override

Note: You can use the Check procedure CHECK_ADD_SUBTASK_OK to verify that you can add a subtask to a particular parent task.

The following table shows the parameters for ADD_TASK.

Note: Some parameters in this table have "See: TASK_IN_TBL_TYPE" as their description. The descriptions for these parameters are shown in the parameter list for the TASK_IN_TBL_TYPE datatype on page 5 – 32. (The parameter names are identical in TASK_IN_TBL_TYPE, except that they do not begin with "P_".)

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the project in the external system
P_PA_PROJECT_ID	IN	NUMBER(15)	Yes	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	Yes	See the TASK_IN_TBL_TYPE Datatype table on page 5 – 32 for a description of this field. By default, you can pass the same value for both PM_TASK_REFERENCE and PA_TASK_NUMBER.
P_PA_TASK_NUMBER	IN	VARCHAR2(25)	Yes	By default, you can pass the same value for both PM_TASK_REFERENCE and PA_TASK_NUMBER.
P_TASK_NAME	IN	VARCHAR2(20)	Yes	See: TASK_IN_TBL_TYPE
P_TASK_DESCRIPTION	IN	VARCHAR2(250)	No	See: TASK_IN_TBL_TYPE

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_TASK_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_TASK_COMPLETION_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_PM_PARENT_TASK_REFERENCE	IN	VARCHAR2(25)	No	See: TASK_IN_TBL_TYPE
P_ADDRESS_ID	IN	NUMBER	No	See: TASK_IN_TBL_TYPE
P_CARRYING_OUT_ORGANIZATION_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_SERVICE_TYPE_CODE	IN	VARCHAR2(30)	No	See: TASK_IN_TBL_TYPE
P_TASK_MANAGER_PERSON_ID	IN	NUMBER(9)	No	See: TASK_IN_TBL_TYPE
P_BILLABLE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_CHARGEABLE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_READY_TO_BILL_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_READY_TO_DISTRIBUTE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_LIMIT_TO_TXN_CONTROLS_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_LABOR_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_LABOR_STD_BILL_RATE_SCHDL	IN	VARCHAR2(20)	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCHEDULE_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_STD_BILL_RATE_SCHDL	IN	VARCHAR2(30)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_SCHEDULE_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	See: TASK_IN_TBL_TYPE
P_LABOR_COST_MULTIPLIER_NAME	IN	VARCHAR2(20)	No	See: TASK_IN_TBL_TYPE
P_COST_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_REV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_INV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_COST_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_REV_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_INV_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCH_TYPE	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_SCH_TYPE	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_ACTUAL_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_ACTUAL_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_EARLY_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_EARLY_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LATE_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LATE_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_SCHEDULED_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_SCHEDULED_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	See: TASK_IN_TBL_TYPE
P_ALLOW_CROSS_CHARGE_FLAG	IN	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)
P_PROJECT_RATE_DATE	IN	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
P_PROJECT_RATE_TYPE	IN	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)
P_CC_PROCESS_LABOR_FLAG	IN	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
P_LABOR_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)

Name	Usage	Type	Req?	Description
P_LABOR_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor flag is set to 'Y', this field is required. (For future use)
P_CC_PROCESS_NL_FLAG	IN	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
P_NL_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required. (For future use)
P_NL_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
P_RECEIVE_PROJECT_INVOICE_FLAG	IN	VARCHAR2(1)	No	Flag that indicates that the task may receive charges from internal suppliers via inter-project billing. (For future use)
P_PA_PROJECT_ID_OUT	OUT	NUMBER(15)		API standard
P_PA_PROJECT_NUMBER_OUT	OUT	VARCHAR2(25)		API standard
P_TASK_ID	OUT	NUMBER(15)		API standard

CREATE_PROJECT

CREATE_PROJECT is a PL/SQL procedure that creates a project in Oracle Projects using a template or an existing project.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Note: CREATE_PROJECT will not copy the WBS structure to the newly created project when attempting to copy a project or template with tasks.

This API uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Note: When loading descriptive flexfields using AMG APIs, if the DFF is not context sensitive, then the parameter ATTRIBUTE_CATEGORY is required to have a value such as 'Global Data Elements'. Otherwise, AMG does not import rows.

The following table shows the parameters for CREATE_PROJECT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_WORKFLOW_STARTED	OUT	VARCHAR2		Shows if a workflow has been started (Y or N)
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PROJECT_IN	IN	PROJECT_IN_REC_TYPE	Yes	See the PROJECT_IN_REC_TYPE Datatype table (next)
P_PROJECT_OUT	OUT	PROJECT_OUT_REC_TYPE		See the PROJECT_OUT_REC_TYPE Datatype table on page 5 - 31
P_KEY_MEMBERS	IN	PROJECT_ROLE_TBL_TYPE	No	See the PROJECT_ROLE_TBL_TYPE Datatype table on page 5 - 31
P_CLASS_CATEGORIES	IN	CLASS_CATEGORY_TBL_TYPE	No	See the CLASS_CATEGORY_TBL_TYPE Datatype table on page 5 - 32
P_TASKS_IN	IN	TASK_IN_TBL_TYPE	No	See the TASK_IN_TBL_TYPE Datatype table on page 5 - 32
P_TASKS_OUT	OUT	TASK_OUT_TBL_TYPE		See the TASK_OUT_TBL_TYPE Datatype table on page 5 - 36

Record and Table Datatypes

The record and table datatypes in the preceding tables are defined on the following pages.

PROJECT_IN_REC_TYPE Datatype

The following table shows the PROJECT_IN_REC_TYPE datatype.

Name	Type	Req?	Description
PM_PROJECT_REFERENCE	VARCHAR2(25)	Yes	The reference code that uniquely identifies the project in the external system. See Examples and Remarks: page 5 - 36.
PA_PROJECT_ID	NUMBER(15)	For update	The reference code that uniquely identifies the project in Oracle Projects
PA_PROJECT_NUMBER	VARCHAR2(25)	No	The project number that
PROJECT_NAME	VARCHAR2(30)	Yes	Unique name of the project uniquely identifies the project in Oracle Projects
CREATED_FROM_PROJECT_ID	NUMBER(15)	Yes	Number that uniquely identifies the template from which this project originates
CARRYING_OUT_ORGANIZATION_ID	NUMBER(15)	Template*	The identification code of the organization responsible for the project work
PUBLIC_SECTOR_FLAG	VARCHAR2(1)	Template*	Flag that indicates whether this project is in the Public or the Private sector
PROJECT_STATUS_CODE	VARCHAR2(30)	Template*	The status of the project. Any status other than CLOSED is considered active.
DESCRIPTION	VARCHAR2(250)	Template*	The description of the project
START_DATE	DATE	Template*	The date on which the project starts
COMPLETION_DATE	DATE	Template*	The date on which the project is completed
DISTRIBUTION_RULE	VARCHAR2(30)	Template*	The distribution rule that specifies the contract project's revenue accrual and billing method
CUSTOMER_ID	NUMBER(15)	Template*	The identification code of the project's customer
PROJECT_RELATIONSHIP_CODE	VARCHAR2(30)	Yes	The type of customer relationship the customer has on the project
ACTUAL_START_DATE	DATE	No	The actual project start date in the external system

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
ACTUAL_FINISH_DATE	DATE	No	The actual project finish date in the external system
EARLY_START_DATE	DATE	No	The early project start date in the external system
EARLY_FINISH_DATE	DATE	No	The early project finish date in the external system
LATE_START_DATE	DATE	No	The late project start date in the external system
LATE_FINISH_DATE	DATE	No	The late project finish date in the external system
SCHEDULED_START_DATE	DATE	No	The scheduled project start date in the external system
SCHEDULED_FINISH_DATE	DATE	No	The scheduled project finish date in the external system
ATTRIBUTE_CATEGORY	VARCHAR2(30)	No	Used by descriptive flexfields
ATTRIBUTE1	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE2	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE3	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE4	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE5	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE6	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE7	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE8	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE9	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE10	VARCHAR2(150)	No	Descriptive flexfield
OUTPUT_TAX_CODE	VARCHAR2(30)	No	Indicates whether tax rate defined for the Project will be used for Customer Invoices. (For future use)
RETENTION_TAX_CODE	VARCHAR2(30)	No	Indicates whether tax rate defined for the Retention will be used for Customer Invoices. (For future use)
PROJECT_CURRENCY_CODE	VARCHAR2(15)	No	Project currency code. This value will not be displayed in the form for release 11.5. Currency code of the set of books will be defaulted. (For future use)
ALLOW_CROSS_CHARGE_FLAG	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
PROJECT_RATE_DATE	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
PROJECT_RATE_TYPE	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)
CC_PROCESS_LABOR_FLAG	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
LABOR_TP_SCHEDULE_ID	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
LABOR_TP_FIXED_DATE	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor flag is set to 'Y', this field is required. (For future use)
CC_PROCESS_NL_FLAG	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
NL_TP_SCHEDULE_ID	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required. (For future use)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
NL_TP_FIXED_DATE	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
CC_TAX_TASK_ID	NUMBER	No	Identifier of the task to which intercompany tax items on the intercompany AP invoice are charged. (For future use)

* Based on template setup

PROJECT_OUT_REC_TYPE Datatype

The following table shows the PROJECT_OUT_REC_TYPE datatype.

Name	Type	Req?	Description
PA_PROJECT_ID	NUMBER(15)		The reference code that uniquely identifies the project in Oracle Projects
PA_PROJECT_NUMBER	VARCHAR2(25)		The number that uniquely identifies the project in Oracle Projects
RETURN_STATUS	VARCHAR2(1)		API standard

PROJECT_ROLE_TBL_TYPE Datatype

The following table shows the PROJECT_ROLE_TBL_TYPE datatype.

Name	Type	Req?	Description
PERSON_ID	NUMBER(9)	Based on template setup	The identification code of the employee that manages or administers the project
PROJECT_ROLE_TYPE	VARCHAR2(20)	Yes, if PERSON_ID is not NULL	The type of role that the project player has on the project
START_DATE	DATE	No. Project start date is the default.	Indicates when this person starts playing this role
END_DATE	DATE	No	Indicates when this person stops playing this role

CLASS_CATEGORY_TBL_TYPE Datatype

The following table shows the CLASS_CATEGORY_TBL_TYPE datatype.

Name	Type	Req?	Description
CLASS_CATEGORY	VARCHAR2(30)	Template (based on template setup)	The class category by which the project is classified
CLASS_CODE	VARCHAR2(30)	Yes (only if CLASS_CATEGORY is not NULL)	The class code that classifies the project

TASK_IN_TBL_TYPE Datatype

The following table shows the TASK_IN_TBL_TYPE datatype.

Name	Type	Req?	Description
PM_TASK_REFERENCE	VARCHAR2(25)	Yes, or PA_TASK_ID is given	The reference code that identifies a project's task in the external system
PA_TASK_ID	NUMBER(15)	For update	The reference code that uniquely identifies a task within a project in Oracle Projects
TASK_NAME	VARCHAR2(20)	Yes	The name that uniquely identifies a task within a project
PA_TASK_NUMBER	VARCHAR2(25)	Yes	The number that identifies the task in Oracle Projects. Intended for systems that maintain a task number in addition to a unique TASK_REFERENCE.
TASK_DESCRIPTION	VARCHAR2(250)	No	Description of the task
TASK_START_DATE	DATE	No	The date on which the task starts
TASK_COMPLETION_DATE	DATE	No	The date on which the task is completed
PM_PARENT_TASK_REFERENCE	VARCHAR2(25)	No	The reference code that identifies the task's parent task in the external system
PA_PARENT_TASK_ID	NUMBER	For update	The identification code of the task's parent task in Oracle Projects
ADDRESS_ID	NUMBER	No	The address of one of the customers logically linked to this task
CARRYING_OUT_ORGANIZATION_ID	NUMBER(15)	No	The identification code of the organization responsible for the task work. The task organization defaults to the project organization upon creation of the task.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
SERVICE_TYPE_CODE	VARCHAR2(30)	No	The type of work performed on the task
TASK_MANAGER_PERSON_ID	NUMBER(9)	No	The identification code of the employee that manages the task. See Examples and Remarks: page 5 - 36.
BILLABLE_FLAG	VARCHAR2(1)	No	Default flag for items charged to the task that indicates if the item can accrue revenue (Y or N)
CHARGEABLE_FLAG	VARCHAR2(1)	No	Flag that indicates if expenditure items can be charged to the task. Only lowest tasks are chargeable.
READY_TO_BILL_FLAG	VARCHAR2(1)	No	Flag that indicates whether the task is authorized to be invoiced
READY_TO_DISTRIBUTE_FLAG	VARCHAR2(1)	No	Flag that indicates whether the task is authorized for revenue accrual
LIMIT_TO_TXN_CONTROLS_FLAG	VARCHAR2(1)	No	Flag that indicates that users can charge to the task only those expenditures listed in the task's transaction controls
LABOR_BILL_RATE_ORG_ID	NUMBER(15)	No	The identification code of the organization that owns the labor standard bill rate schedule
LABOR_STD_BILL_RATE_SCHDL	VARCHAR2(20)	No	The labor standard bill rate schedule used to calculate revenue for labor expenditure items charged to the task
LABOR_SCHEDULE_FIXED_DATE	DATE	No	The date used to determine the effective bill rates of the task standard labor bill rate schedule
LABOR_SCHEDULE_DISCOUNT	NUMBER(7,4)	No	The percentage to be discounted from the task standard labor bill rate schedule
NON_LABOR_BILL_RATE_ORG_ID	NUMBER(15)	No	The identification code of the organization that owns the non-labor standard bill rate schedule
NON_LABOR_STD_BILL_RATE_SCHDL	VARCHAR2(30)	No	The non-labor standard bill rate schedule used to calculate revenue for non-labor expenditure items charged to the task
NON_LABOR_SCHEDULE_FIXED_DATE	DATE	No	The fixed date used to determine the effective bill rates of the standard non-labor bill rate schedule
NON_LABOR_SCHEDULE_DISCOUNT	NUMBER(7,4)	No	The percentage to be discounted from the task standard non-labor bill rate schedule

Name	Type	Req?	Description
LABOR_COST_MULTIPLIER_NAME	VARCHAR2(20)	No	The labor cost multiplier defined for the task of a premium project. The labor cost multiplier is populated for all overtime expenditure items charged to the task.
COST_IND_RATE_SCH_ID	NUMBER(15)	No	The identification code of the default costing burden schedule
REV_IND_RATE_SCH_ID	NUMBER(15)	No	The identification code of the default revenue burden schedule
INV_IND_RATE_SCH_ID	NUMBER(15)	No	The identification code of the default invoice burden schedule
COST_IND_SCH_FIXED_DATE	DATE	No	The scheduled fixed date of the firm costing burden schedule
REV_IND_SCH_FIXED_DATE	DATE	No	The scheduled fixed date of the firm revenue burden schedule
INV_IND_SCH_FIXED_DATE	DATE	No	The scheduled fixed date of the firm invoice burden schedule
LABOR_SCH_TYPE	VARCHAR2(1)	No	The scheduled type of labor expenditure items
NON_LABOR_SCH_TYPE	VARCHAR2(1)	No	The scheduled type of non-labor expenditure items
ACTUAL_START_DATE	DATE	No	The actual start date of the project in the external system
ACTUAL_FINISH_DATE	DATE	No	The actual finish date of the project in the external system
EARLY_START_DATE	DATE	No	The early start date of the project in the external system
EARLY_FINISH_DATE	DATE	No	The early finish date of the project in the external system
LATE_START_DATE	DATE	No	The late start date of the project in the external system
LATE_FINISH_DATE	DATE	No	The late finish date of the project in the external system
SCHEDULED_START_DATE	DATE	No	The scheduled start date of the project in the external system
SCHEDULED_FINISH_DATE	DATE	No	The scheduled finish date of the project in the external system
ALLOW_CROSS_CHARGE_FLAG	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)
PROJECT_RATE_DATE	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
PROJECT_RATE_TYPE	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
CC_PROCESS_LABOR_FLAG	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
LABOR_TP_SCHEDULE_ID	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
LABOR_TP_FIXED_DATE	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
CC_PROCESS_NL_FLAG	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
NL_TP_SCHEDULE_ID	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor_flag is set to 'Y', this field is required. (For future use)
NL_TP_FIXED_DAT	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
RECEIVE_PROJECT_INVOICE_FLAG	VARCHAR2(1)	No	Flag that indicates that the task may receive charges from internal suppliers via inter-project billing. (For future use)
ATTRIBUTE_CATEGORY	VARCHAR2(30)	No	Used by descriptive flexfields

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Type	Req?	Description
ATTRIBUTE1	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE2	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE3	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE4	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE5	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE6	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE7	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE8	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE9	VARCHAR2(150)	No	Descriptive flexfield
ATTRIBUTE10	VARCHAR2(150)	No	Descriptive flexfield

TASK_OUT_TBL_TYPE Datatype

The following table shows the TASK_OUT_TBL_TYPE datatype.

Name	Type	Req?	Description
PA_TASK_ID	NUMBER(15)		The reference code that uniquely identifies a task within a project in Oracle Projects
PM_TASK_REFERENCE	VARCHAR2(25)		The reference code that identifies a project's task in the external system
RETURN_STATUS	VARCHAR2(1)		API standard

Examples and Remarks

PM_PROJECT_REFERENCE

Systems that you use to create projects in Oracle Projects assign a unique number to every project. You can set up Oracle Projects either to generate project numbers automatically or to support manual entry of numbers.

When Oracle Projects is set up for automatic numbering:

- The number generated automatically by Oracle Projects is stored in the column SEGMENT1.
- The number assigned by the external system is stored in the column PM_PROJECT_REFERENCE.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

When Oracle Projects is set up for manual numbering, the number assigned by the external system is stored in both SEGMENT1 and PM_PROJECT_REFERENCE.

Note: Oracle Projects windows display only SEGMENT1 as the project number, so you should set up Oracle Projects to support manual numbering if you plan to integrate Oracle Projects with an external system.

Project and Task Start and Finish Dates

Most external systems hold additional start and finish dates for projects and tasks. For information about using a client extension to pass these additional dates (instead of the default Oracle Projects project dates), see Project and Task Date Client Extension API: page G – 2.

TASK_MANAGER_PERSON_ID

To ensure that the task manager has been defined in Oracle Projects, use a list of values (PA_TASK MANAGERS_LOV_V) to select a task manager's person identification code.

DELETE_PROJECT

DELETE_PROJECT is a PL/SQL procedure used to delete a project and its tasks from Oracle Projects.

Business Rule (project level)

You cannot delete a project if any of these items exist:

- Event
- Expenditure item
- Purchase order distribution
- Purchase order requisition
- Supplier invoice
- Invoice distribution
- Funding
- Budget
- Commitment transaction

- Compensation rule set
- Reference from other project

Business Rule (task level)

You cannot delete a project if any of its tasks cannot be deleted. Use the Check procedure CHECK_DELETE_TASK_OK to see if you can delete a certain task. You cannot delete a task if any of the following exists:

- Event at top task
- Funding at top task
- Budget at top task
- Expenditure item at lowest task
- Purchase order line at lowest task
- Requisition line at lowest task
- Supplier invoice (Oracle Payables invoice) at lowest task
- Budget at lowest task

The following table shows the DELETE_PROJECT parameters.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PA_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects

DELETE_TASK

DELETE_TASK is a PL/SQL procedure used to delete tasks of a project in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Business Rules (task level)

Oracle Projects imposes the following business rules. Rules 1–8 apply to cascaded task deletion, which means that deleting a task also deletes all of its subtasks.

You can delete a top task only if the task satisfies Rules 1 through 8:

1. No top task event, such as revenue or billing, exists
2. No top task funding exists
3. No top task budget exists

You can delete a mid or lowest task if the task satisfies Rules 4 through 8 (for a mid task, the rules relate to the lowest tasks below that mid task):

4. No lowest task expenditure item exists
5. No lowest task purchase order line exists
6. No lowest task requisition line exists
7. No lowest task supplier invoice exists
8. No lowest task budget exists

Business rules 9 through 11 apply to non-cascaded task deletion, which means that deleting a task deletes only that task, and moves all subtasks below it up one level in the project's work breakdown structure.

9. You can delete a mid task at all times.
10. You can delete a top task if it satisfies Rules 1 through 3 above.
11. You can delete a lowest task if it satisfies Rules 4 through 8 above.

The following table shows the DELETE_TASK parameters.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system

Name	Usage	Type	Req?	Description
P_PA_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system
P_PA_TASK_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_CASCADE_DELETE_FLAG	IN	VARCHAR2(1)	No	When 'Y' is passed, this task and all its subtasks are deleted (default = 'N')
P_PROJECT_ID	OUT	NUMBER(15)		API standard
P_TASK_ID	OUT	NUMBER(15)		API standard

UPDATE_PROJECT

UPDATE_PROJECT is a PL/SQL procedure that pushes project and task information from your external system to Oracle Projects to reflect any changes you have made in the external system.

UPDATE_PROJECT uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Oracle Projects imposes project- and task-level business rules that restrict the changes you can make to project and task information. To ensure that Oracle Projects accepts all the project or task changes you make within your external system, read the following rules before making your changes. You can also use the Check procedures (starting on: page 5 – 62) to identify the types of changes that Oracle Projects supports.

Business Rules (project level)

Oracle Projects imposes the following business rules.

These rules apply to project numbers, names, types, and organizations:

- You cannot change a project number if expenditure items or invoices have been charged to the project.
 - New project numbers must be unique within Oracle Projects. Use CHECK_UNIQUE_PROJECT_REFERENCE (a Check procedure) to verify that the new project number is unique.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- If you use an external system to create original project plans, choose manual project numbering. Numbers generated automatically by external systems may not be unique in Oracle Projects and will be replaced by new project numbers generated by Oracle Projects.
- The new project name must be unique.
- You cannot change the project type (indirect, capital, or contract) of a project.
- You cannot change the project organization if cost distribution lines, draft revenue, or draft invoices have been charged against the project.

These rules apply to project key members and customers:

- A project can have only one active project manager.
- A project can have any number of key members (other than project managers).
- If you assign a new project manager to an existing project, the default start date for the new project manager is the system date. The default end date for the current project manager is the previous day.
- If the start date of a key member other than a project manager is not passed or passed as NULL, the start day is derived from the project start date. When PROJECT_START_DATE is NULL, the default start date for the key member is NULL.
- UPDATE_PROJECT does not support a person performing the same role (other than project manager) for a given project during different periods.
- A project can have only one primary customer.

These rules apply to project start and completion dates:

- Project start and completion dates must include the first task start date and the last task completion date for all tasks included in the project.
- You can leave both the start and completion dates or just the completion date blank; however, you must enter a start date if you want to enter a completion date.
- If you change the project status to Closed, then the default completion date is the system date. If you subsequently reopen the project, the default completion date is NULL.

- A NULL value for any of the project fields listed below results in an error message in Oracle Projects. Oracle Projects ignores incoming NULL values for these fields and retains their original values.
 - PROJECT_STATUS
 - PUBLIC_SECTOR_FLAG
 - PROJECT_NUMBER
 - PROJECT_NAME
 - CARRYING_OUT_ORGANIZATION_ID
 - DISTRIBUTION_RULE for a contract project. (A NULL value for this field raises an error.)

Business Rules (task level)

Oracle Projects imposes the following business rules.

This rule applies to the order in which task information is shared between your external system and Oracle Projects:

- You must interface parent tasks to Oracle Projects before you can interface the related child tasks.

The following rules apply to task numbers, identification codes, and organizations:

- New task numbers must be unique within a project. Use the Check procedure CHECK_UNIQUE_TASK_NUMBER to verify that a new task number is unique in Oracle Projects.
- If the external system pushes both the TASK_ID and the PM_TASK_REFERENCE to Oracle Projects, Oracle Projects uses the TASK_ID to identify the task and updates PM_TASK_REFERENCE with the incoming value (if different).
- You cannot change a task number if any of the following items have been charged against the task:
 - Expenditure items
 - Purchase order distributions
 - Purchase order requisition distributions
 - Supplier invoices
 - Supplier invoice distributions

Note: Use the Check procedure CHECK_TASK_NUMBER_CHANGE_OK to verify if Oracle Projects allows you to change the number of a certain task.

- You cannot change a task organization if any of the following items have been charged against the task:
 - Cost distribution lines
 - Revenue distribution lines
 - Draft invoices

These rules apply to task start and completion dates:

- A task start date must occur:
 - After the parent task start date
 - Before the start date of any subtasks
 - Between the project start and completion dates
- Each task with a completion date must also have a start date.
- A task completion date must occur before the project completion date.

These rules apply to moving a task within a project's work breakdown structure (WBS):

- Because billing, budgeting, and creating capital assets are driven from top tasks, you can move a subtask only if its new parent task belongs to the same top task.
- You cannot change a top task to a subtask.
- You cannot change a subtask to a top task.

These rules apply to task fields and attributes:

- You cannot change any of the following task fields to NULL:
 - TASK_NAME
 - PM_TASK_REFERENCE
 - TASK_NUMBER
 - READY_TO_BILL_FLAG
 - READY_TO_DISTRIBUTE_FLAG
 - CARRYING_OUT_ORGANIZATION_ID
 - SERVICE_TYPE_CODE
- You can change the following task attributes without restriction:
 - Task manager
 - Description
 - Other flags (not mentioned previously)
 - Labor and non-labor data
 - Schedules and rates

The following table shows the parameters for UPDATE_PROJECT.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_WORKFLOW_STARTED	OUT	VARCHAR2(1)		Shows if a workflow has been started (Y or N)
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PROJECT_IN	IN	PROJECT_IN_REC_TYPE	Yes	See the PROJECT_IN_REC_TYPE Datatype table on page 5 - 28
P_PROJECT_OUT	OUT	PROJECT_OUT_REC_TYPE		See the PROJECT_OUT_REC_TYPE Datatype table on page 5 - 31
P_KEY_MEMBERS_IN	IN	PROJECT_ROLE_TBL_TYPE	No	See the PROJECT_ROLE_TBL_TYPE Datatype table on page 5 - 31
P_CLASS_CATEGORIES_IN	IN	CLASS_CATEGORY_TBL_TYPE	No	See the CLASS_CATEGORY_TBL_TYPE Datatype table on page 5 - 32
P_TASKS_IN	IN	TASK_IN_TBL_TYPE	No	See the TASK_IN_TBL_TYPE Datatype table on page 5 - 32
P_TASKS_OUT	OUT	TASK_OUT_TBL_TYPE		See the TASK_OUT_TBL_TYPE Datatype table on page 5 - 36

UPDATE_TASK

UPDATE_TASK is a PL/SQL procedure used to update existing tasks of a project in Oracle Projects. We replaced the task record type with a parameter that uses a standard datatype (VARCHAR2, NUMBER, and DATE) for every field in the record type definition so you can call this procedure directly.

Business Rules (task level)

Oracle Projects imposes the following business rules.

This rule applies to the order in which task information is shared between your external system and Oracle Projects:

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- You must interface the definitions of parent tasks to Oracle Projects before you can interface the definitions of the related child tasks.

The following rules apply to task numbers, identification codes, and organizations:

- A new task number must be unique within a project. (You can use the Check procedure CHECK_UNIQUE_TASK_NUMBER to verify whether your new task number is unique in Oracle Projects.)
- If the external system pushes both the TASK_ID and the PM_TASK_REFERENCE to Oracle Projects, Oracle Projects uses the TASK_ID to identify the task and updates PM_TASK_REFERENCE with the incoming value (if different).
- You cannot change a task number if any of the following items have been charged against the task:
 - Expenditure items
 - Purchase order distributions
 - Purchase order requisition distributions
 - Supplier invoices
 - Supplier invoice distributions

Note: You can use the Check procedure CHECK_TASK_NUMBER_CHANGE_OK to verify whether Oracle Projects will allow you to change the number of a certain task.

- You cannot change a task organization if any of the following items have been charged against the task:
 - Cost distribution lines
 - Revenue distribution lines
 - Draft invoices

The following rules apply to task start and completion dates:

- A task start date must occur:
 - After the parent task start date
 - Before the start date of any subtasks
 - Between the project start and completion dates
- Each task with a completion date must also have a start date.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- A task completion date must occur before the project completion date.

The following rules apply to moving a task within a project's work breakdown structure (WBS).

- You can move a subtask as long as its new parent task belongs to the same top task, because billing, budgeting, and creating capital assets are driven from top tasks.
- You cannot change a top task to a subtask.
- You cannot change a subtask to a top task.

The following rules apply to changing task fields and attributes:

- You cannot update task fields with a NULL value. Only a field with a valid NOT NULL value will be updated.
- You cannot change any of the following task fields to NULL:
 - TASK_NAME
 - PM_TASK_REFERENCE
 - TASK_NUMBER
 - READY_TO_BILL_FLAG
 - READY_TO_DISTRIBUTE_FLAG
 - CARRYING_OUT_ORGANIZATION_ID
 - SERVICE_TYPE_CODE
- You can change the following task attributes without restriction:
 - Task manager
 - Description
 - Other flags not mentioned previously
 - Labor and non-labor data
 - Schedules and rates

The following table shows the parameters for UPDATE_TASK.

Note: Some parameters in this table have "See: TASK_IN_TBL_TYPE" as their description. The descriptions for these parameters are shown in the parameter list for the TASK_IN_TBL_TYPE datatype on page 5 – 32. (The parameter names are identical in TASK_IN_TBL_TYPE, except that they do not begin with "P_".)

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	Code identifying the external system
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the project in the external system
P_PA_PROJECT_ID	IN	NUMBER(15)	Yes	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	Yes	See: TASK_IN_TBL_TYPE
P_PA_TASK_ID	IN	NUMBER(15)	Yes	See: TASK_IN_TBL_TYPE
P_TASK_NAME	IN	VARCHAR2(20)	Yes	See: TASK_IN_TBL_TYPE
P_PA_TASK_NUMBER	IN	VARCHAR2(25)	Yes	See: TASK_IN_TBL_TYPE
P_TASK_DESCRIPTION	IN	VARCHAR2(250)	No	See: TASK_IN_TBL_TYPE
P_TASK_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_TASK_COMPLETION_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_PM_PARENT_TASK_REFERENCE	IN	VARCHAR2(25)	No	See: TASK_IN_TBL_TYPE
P_PA_PARENT_TASK_ID	IN	NUMBER	No	See: TASK_IN_TBL_TYPE
P_ADDRESS_ID	IN	NUMBER	No	See: TASK_IN_TBL_TYPE
P_CARRYING_OUT_ORGANIZATION_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_SERVICE_TYPE_CODE	IN	VARCHAR2(30)	No	See: TASK_IN_TBL_TYPE
P_TASK_MANAGER_PERSON_ID	IN	NUMBER(9)	No	See: TASK_IN_TBL_TYPE
P_BILLABLE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_CHARGEABLE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_READY_TO_BILL_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_READY_TO_DISTRIBUTE_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_LIMIT_TO_TXN_CONTROLS_FLAG	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_LABOR_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_LABOR_STD_BILL_RATE_SCHDL	IN	VARCHAR2(20)	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCHEDULE_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_STD_BILL_RATE_SCHDL	IN	VARCHAR2(30)	No	See: TASK_IN_TBL_TYPE

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_NON_LABOR_SCHEDULE_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	See: TASK_IN_TBL_TYPE
P_LABOR_COST_MULTIPLIER_NAME	IN	VARCHAR2(20)	No	See: TASK_IN_TBL_TYPE
P_COST_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_REV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_INV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	See: TASK_IN_TBL_TYPE
P_COST_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_REV_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_INV_IND_SCH_FIXED_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LABOR_SCH_TYPE	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_NON_LABOR_SCH_TYPE	IN	VARCHAR2(1)	No	See: TASK_IN_TBL_TYPE
P_ACTUAL_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_ACTUAL_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_EARLY_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_EARLY_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LATE_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_LATE_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_SCHEDULED_START_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_SCHEDULED_FINISH_DATE	IN	DATE	No	See: TASK_IN_TBL_TYPE
P_ALLOW_CROSS_CHARGE_FLAG	IN	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)
P_PROJECT_RATE_DATE	IN	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
P_PROJECT_RATE_TYPE	IN	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)
P_CC_PROCESS_LABOR_FLAG	IN	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)

Name	Usage	Type	Req?	Description
P_LABOR_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
P_LABOR_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor flag is set to 'Y', this field is required. (For future use)
P_CC_PROCESS_NL_FLAG	IN	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
P_NL_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required. (For future use)
P_NL_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
P_RECEIVE_PROJECT_INVOICE_FLAG	IN	VARCHAR2(1)	No	Flag that indicates that the task may receive charges from internal suppliers via inter-project billing. (For future use)
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	
P_OUT_PA_TASK_ID	OUT	NUMBER(15)		API standard
P_OUT_PM_TASK_REFERENCE	OUT	VARCHAR2(25)		API standard

CLEAR_PROJECT

CLEAR_PROJECT is a Load-Execute-Fetch procedure used to clear the global data structures set up during the Load process.

EXECUTE_CREATE_PROJECT

EXECUTE_CREATE_PROJECT is a Load-Execute-Fetch procedure used to create a project and its tasks using the data stored in the global tables during the Load process.

The following table shows the parameters for EXECUTE_CREATE_PROJECT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	
P_COMMIT	IN	VARCHAR2(1)	No	Default = 'F'
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Default = 'F'
P_MSG_COUNT	OUT	NUMBER		
P_MSG_DATA	OUT	VARCHAR2(2000)		
P_RETURN_STATUS	OUT	VARCHAR2(1)		
P_WORKFLOW_STARTED	OUT	VARCHAR2(1)		Shows if a workflow has been started (Y or N)
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	
P_PA_PROJECT_ID	OUT	NUMBER		
P_PA_PROJECT_NUMBER	OUT	VARCHAR2(25)		

EXECUTE_UPDATE_PROJECT

EXECUTE_UPDATE_PROJECT is a Load–Execute–Fetch procedure used to update an existing project, including changing or adding project data, adding new tasks, and updating existing tasks. This API does not delete tasks; rather, it uses the data stored in the global tables during the Load process.

The following table shows the parameters for EXECUTE_UPDATE_PROJECT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	
P_COMMIT	IN	VARCHAR2(1)	No	Default = 'F'
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Default = 'F'
P_MSG_COUNT	OUT	NUMBER		
P_MSG_DATA	OUT	VARCHAR2(2000)		
P_RETURN_STATUS	OUT	VARCHAR2(1)		
P_WORKFLOW_STARTED	OUT	VARCHAR2(1)		Shows if a workflow has been started (Y or N)
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	

FETCH_TASK

FETCH_TASK is a Load–Execute–Fetch procedure used to fetch output parameters related to tasks.

The following table shows the parameters for FETCH_TASK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Default = 'F'
P_RETURN_STATUS	OUT	VARCHAR2(1)		
P_TASK_INDEX	IN	NUMBER	Yes	
P_PA_TASK_ID	OUT	NUMBER(15)		
P_PM_TASK_REFERENCE	OUT	VARCHAR2(25)		
P_TASK_RETURN_STATUS	OUT	VARCHAR2(1)		

INIT_PROJECT

INIT_PROJECT is a Load-Execute-Fetch procedure used to set up the global data structures. Other Load-Execute-Fetch procedures use the structures to create a new project in Oracle Projects.

LOAD_CLASS_CATEGORY

LOAD_CLASS_CATEGORY is a Load-Execute-Fetch procedure used to load class categories to a global PL/SQL table.

The following table shows the parameters for LOAD_CLASS_CATEGORY.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_CLASS_CATEGORY	IN	VARCHAR2(30)	Depends on template setup	
P_CLASS_CODE	IN	VARCHAR2(30)	Yes, If P_CLASS_CATEGORY is not NULL	

LOAD_KEY_MEMBER

LOAD_KEY_MEMBER is a Load-Execute-Fetch procedure used to load key members to a global PL/SQL table.

The following table shows the parameters for LOAD_KEY_MEMBER.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PERSON_ID	IN	NUMBER(9)	Depends on template setup	
P_PROJECT_ROLE_TYPE	IN	VARCHAR2(20)	Yes, If P_PERSON_ID is not NULL	
P_START_DATE	IN	DATE	No	Default = sysdate
P_END_DATE	IN	DATE	No	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

LOAD_PROJECT

LOAD_PROJECT is a Load-Execute-Fetch procedure used to load a project to a global PL/SQL record.

The following table shows the parameters for LOAD_PROJECT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Default = 'F'
P_RETURN_STATUS	OUT	VARCHAR2(1)	No	
P_PA_PROJECT_ID	IN	NUMBER	No, used for update only	
P_PA_PROJECT_NUMBER	IN	VARCHAR2(25)	No, used for update only	
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	Yes	
P_PROJECT_NAME	IN	VARCHAR2(30)	Yes	
P_CREATED_FROM_PROJECT_ID	IN	NUMBER(15)	Yes	
P_CARRYING_OUT_ORGANIZATION_ID	IN	NUMBER(15)	Depends on template setup	
P_PUBLIC_SECTOR_FLAG	IN	VARCHAR2(1)	Depends on template setup	
P_PROJECT_STATUS_CODE	IN	VARCHAR2(30)	Depends on template setup	
P_DESCRIPTION	IN	VARCHAR2(250)	Depends on template setup	
P_START_DATE	IN	DATE	Depends on template setup	
P_COMPLETION_DATE	IN	DATE	Depends on template setup	
P_DISTRIBUTION_RULE	IN	VARCHAR2(30)	Depends on template setup	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_CUSTOMER_ID	IN	NUMBER(15)	Depends on template setup	
P_PROJECT_RELATIONSHIP_CODE	IN	VARCHAR2(30)	Depends on template setup	
P_ACTUAL_START_DATE	IN	DATE	No	
P_ACTUAL_FINISH_DATE	IN	DATE	No	
P_EARLY_START_DATE	IN	DATE	No	
P_EARLY_FINISH_DATE	IN	DATE	No	
P_LATE_START_DATE	IN	DATE	No	
P_LATE_FINISH_DATE	IN	DATE	No	
P_SCHEDULED_START_DATE	IN	DATE	No	
P_SCHEDULED_FINISH_DATE	IN	DATE	No	
P_OUTPUT_TAX_CODE	IN	VARCHAR2(30)	No	Indicates whether tax rate defined for the Project will be used for Customer Invoices. (For future use)
P_RETENTION_TAX_CODE	IN	VARCHAR2(30)	No	Indicates whether tax rate defined for the Retention will be used for Customer Invoices. (For future use)
P_PROJECT_CURRENCY_CODE	IN	VARCHAR2(15)	No	Project currency code. This value will not be displayed in the form for release 11.5. Currency code of the set of books will be defaulted. (For future use)
P_ALLOW_CROSS_CHARGE_FLAG	IN	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)
P_PROJECT_RATE_DATE	IN	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
P_PROJECT_RATE_TYPE	IN	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)
P_CC_PROCESS_LABOR_FLAG	IN	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_LABOR_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
P_LABOR_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor flag is set to 'Y', this field is required. (For future use)
P_CC_PROCESS_NL_FLAG	IN	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
P_NL_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required. (For future use)
P_NL_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
P_CC_TAX_TASK_ID	IN	NUMBER	No	Identifier of the task to which intercompany tax items on the intercompany AP invoice are charged. (For future use)
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	

LOAD_TASK

LOAD_TASK is a Load-Execute-Fetch procedure used to load a task to a global PL/SQL table.

Business Rule (task level)

Oracle Projects imposes the following business rule:

- Parent tasks must be loaded before their subtasks.

The following table shows the parameters for LOAD_TASK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No (default = 'F')	
P_RETURN_STATUS	OUT	VARCHAR2(1)		
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	
P_PA_TASK_ID	IN	NUMBER	No	
P_TASK_NAME	IN	VARCHAR2(20)	Yes	
P_PA_TASK_NUMBER	IN	VARCHAR2(25)	Yes	
P_TASK_DESCRIPTION	IN	VARCHAR2(250)	No	
P_TASK_START_DATE	IN	DATE	No	
P_TASK_COMPLETION_DATE	IN	DATE	No	
P_PM_PARENT_TASK_REFERENCE	IN	VARCHAR2(25)	No	
P_PA_PARENT_TASK_ID	IN	NUMBER	No	
P_ADDRESS_ID	IN	NUMBER	No	
P_CARRYING_OUT_ORGANIZATION_ID	IN	NUMBER(15)	No	
P_SERVICE_TYPE_CODE	IN	VARCHAR2(30)	No	
P_TASK_MANAGER_PERSON_ID	IN	NUMBER(9)	No	
P_BILLABLE_FLAG	IN	VARCHAR2(1)	No	
P_CHARGEABLE_FLAG	IN	VARCHAR2(1)	No	
P_READY_TO_BILL_FLAG	IN	VARCHAR2(1)	No	
P_READY_TO_DISTRIBUTE_FLAG	IN	VARCHAR2(1)	No	
P_LIMIT_TO_TXN_CONTROLS_FLAG	IN	VARCHAR2(1)	No	
P_LABOR_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	
P_LABOR_STD_BILL_RATE_SCHDL	IN	VARCHAR2(20)	No	
P_LABOR_SCHEDULE_FIXED_DATE	IN	DATE	No	
P_LABOR_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_NL_BILL_RATE_ORG_ID	IN	NUMBER(15)	No	
P_NL_STD_BILL_RATE_SCHDL	IN	VARCHAR2(30)	No	
P_NL_SCHEDULE_FIXED_DATE	IN	DATE	No	
P_NL_SCHEDULE_DISCOUNT	IN	NUMBER(7,4)	No	
P_LABOR_COST_MULTIPLIER_NAME	IN	VARCHAR2(20)	No	
P_COST_IND_RATE_SCH_ID	IN	NUMBER(15)	No	
P_REV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	
P_INV_IND_RATE_SCH_ID	IN	NUMBER(15)	No	
P_COST_IND_SCH_FIXED_DATE	IN	DATE	No	
P_REV_IND_SCH_FIXED_DATE	IN	DATE	No	
P_INV_IND_SCH_FIXED_DATE	IN	DATE	No	
P_LABOR_SCH_TYPE	IN	VARCHAR2(1)	No	
P_NL_SCH_TYPE	IN	VARCHAR2(1)	No	
P_ACTUAL_START_DATE	IN	DATE	No	
P_ACTUAL_FINISH_DATE	IN	DATE	No	
P_EARLY_START_DATE	IN	DATE	No	
P_EARLY_FINISH_DATE	IN	DATE	No	
P_LATE_START_DATE	IN	DATE	No	
P_LATE_FINISH_DATE	IN	DATE	No	
P_SCHEDULED_START_DATE	IN	DATE	No	
P_SCHEDULED_FINISH_DATE	IN	DATE	No	
P_ALLOW_CROSS_CHARGE_FLAG	IN	VARCHAR2(1)	No	Cross charge allowed? Value is required. Default Value is 'N'. This value can be overridden at any task level. (For future use)
P_PROJECT_RATE_DATE	IN	DATE	No	Default project currency rate date (date for accounting currency rate for a given rate type). (For future use)
P_PROJECT_RATE_TYPE	IN	VARCHAR2(30)	No	Default project currency rate type (e.g., Spot, Corporate). (For future use)
P_CC_PROCESS_LABOR_FLAG	IN	VARCHAR2(1)	No	Flag that indicates cross charge processing is to be performed for labor transactions charged to the project. Default value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)

Name	Usage	Type	Req?	Description
P_LABOR_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged labor transactions. This is defaulted to a project from the project template. If cc_process_labor_flag is set to 'Y', this field is required. (For future use)
P_LABOR_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for labor transactions. This is defaulted to a project from the project template. This value for the project is default for the task fixed date. If cc_process_labor flag is set to 'Y', this field is required. (For future use)
P_CC_PROCESS_NL_FLAG	IN	VARCHAR2(1)	No	Flag that indicated cross charge processing is to be performed for non-labor transactions charged to the project. Defaulted value for the project template is 'N'. This is defaulted to a project from the project template. (For future use)
P_NL_TP_SCHEDULE_ID	IN	NUMBER	No	Identifier for transfer price schedule to use for cross charged non-transactions. This is defaulted to a project from the project template. If cc_process_nl_labor flag is set to 'Y', this field is required. (For future use)
P_NL_TP_FIXED_DATE	IN	DATE	No	Fixed date to find the effective rate of the bill rate or burden schedule when determining the transfer price for non-labor transactions. This is defaulted to a project from the project template. If cc_process_nl_flag is set to 'Y', this field is required. (For future use)
P_RECEIVE_PROJECT_INVOICE_FLAG	IN	VARCHAR2(1)	No	Flag that indicates that the task may receive charges from internal suppliers via inter-project billing. (For future use)
P_NL_SCH_TYPE	IN	VARCHAR2(1)	No	

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	

CHECK_ADD_SUBTASK_OK

Use the Check procedure CHECK_ADD_SUBTASK_OK to determine if a subtask can be added to a parent task.

The following table shows the parameters for CHECK_ADD_SUBTASK_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that identifies the task in the external system

Name	Usage	Type	Req?	Description
P_TASK_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_ADD_SUBTASK_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not a subtask can be added to this task (Y or N)

CHECK_CHANGE_PARENT_OK

Use the Check procedure CHECK_CHANGE_PARENT_OK to determine if you can move a task from one parent task to another. You can move a task as long as it retains the same top task.

The following table shows the parameters for CHECK_CHANGE_PARENT_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system
P_TASK_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_NEW_PARENT_TASK_ID	IN	NUMBER(15)		The Oracle Projects identification code of the new parent task
P_PM_NEW_PARENT_TASK_REFERENCE	IN	VARCHAR2(25)		The external system reference code of the new parent task
P_CHANGE_PARENT_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not this task can be assigned to a new parent task (Y or N)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHECK_CHANGE_PROJECT_ORG_OK

Use the Check procedure CHECK_CHANGE_PROJECT_ORG_OK to determine if you can change the CARRYING_OUT_ORGANIZATION_ID field for a particular project or task.

The following table shows the parameters for CHECK_CHANGE_PROJECT_ORG_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_CHANGE_PROJECT_ORG_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not the carrying out organization of this project can be changed (Y or N)

CHECK_DELETE_PROJECT_OK

Use the Check procedure CHECK_DELETE_PROJECT_OK to determine if you can delete a project.

The following table shows the parameters for CHECK_DELETE_PROJECT_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

Name	Usage	Type	Req?	Description
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_DELETE_PROJECT_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not this project may be deleted (Y or N)

CHECK_DELETE_TASK_OK

Use the Check procedure CHECK_DELETE_TASK_OK to determine if you can delete a task.

The following table shows the parameters for CHECK_DELETE_TASK_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system
P_TASK_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_DELETE_TASK_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not this task can be deleted (Y or N)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHECK_TASK_NUMBER_CHANGE_OK

Use the Check procedure CHECK_TASK_NUMBER_CHANGE_OK to determine if you can change a task's number.

The following table shows the parameters for CHECK_TASK_NUMBER_CHANGE_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	BOOLEAN		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system
P_TASK_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies a task within a project in Oracle Projects
P_TASK_NUMBER_CHANGE_OK_FLAG	OUT	VARCHAR2(1)		Indicates whether or not the task number can be changed (Y or N)

CHECK_UNIQUE_PROJECT_REFERENCE

Use the Check procedure CHECK_UNIQUE_PROJECT_REFERENCE to determine if a new or changed project reference (PM_PROJECT_REFERENCE) is unique.

The following table shows the parameters for CHECK_UNIQUE_PROJECT_REFERENCE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	BOOLEAN		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_UNIQUE_PROJECT_REF_FLAG	OUT	VARCHAR2(1)		Indicates whether or not this project reference is unique in Oracle Projects (Y or N)

CHECK_UNIQUE_TASK_NUMBER

Use the Check procedure CHECK_UNIQUE_TASK_NUMBER to determine if a new or changed task number is unique within a project.

The following table shows the parameters for CHECK_UNIQUE_TASK_NUMBER.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_TASK_NUMBER	IN	VARCHAR2(25)	No	The number that identifies the task in Oracle Projects
P_UNIQUE_TASK_NUMBER_FLAG	OUT	VARCHAR2(1)		Indicates whether or not this task number is unique in the project within Oracle Projects (Y or N)

CHECK_UNIQUE_TASK_REFERENCE

Use the Check procedure CHECK_UNIQUE_TASK_REFERENCE to determine if a new or changed task reference (PM_TASK_REFERENCE) is unique.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

The following table shows the parameters for
CHECK_UNIQUE_TASK_REFERENCE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system
P_PROJECT_ID	IN	NUMBER(15)	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system
P_UNIQUE_TASK_REF_FLAG	OUT	VARCHAR2(1)		Shows if this task reference is unique in this project within Oracle Projects (Y or N)

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHAPTER

6

Detailed Design -- Budget APIs

This chapter describes how to implement Oracle Activity Management Gateway APIs for budget and budget line information, and provides detailed view and procedure definitions.

Budget APIs

Budgets track the time and resources that you expect to use to complete a project or task. Use your external system to prepare your budget, and then use Budget APIs to interface the budget and budget line into Oracle Projects. Oracle Projects then generates a budget based on the resource budgets and rates stored in the external system. You can interface multiple budget versions to Oracle Projects and baseline them as needed.

Note: When you call a budget API that requires a project identifier, you must pass either the `P_PA_PROJECT_ID` or the `P_PM_PROJECT_REFERENCE` parameter to identify the project. When you call a budget API that requires a resource list identifier, you must pass either the `P_RESOURCE_LIST_NAME` or the `P_RESOURCE_LIST_ID` parameter to identify the resource list.

For introductory remarks about this chapter, see *Overview of Detailed Design*: page 4 – 2.

The views and procedures discussed in this section are listed below. The procedures are located in the public API package `PA_BUDGET_PUB`.

- Views
 - `PA_BASE_BUDGET_BY_GL_PERIOD_V`: page 6 – 4
 - `PA_BASE_BUDGET_BY_PA_PERIOD_V`: page 6 – 4
 - `PA_BUDGET_CHANGE_REASON_V`: page 6 – 5
 - `PA_BUDGET_ENTRY_METHODS_V`: page 6 – 6
 - `PA_BUDGET_STATUS_CODES_V`: page 6 – 6
 - `PA_BUDGET_TYPES_V`: page 6 – 7
 - `PA_ORIG_BUDGET_BY_GL_PERIOD_V`: page 6 – 7
 - `PA_ORIG_BUDGET_BY_PA_PERIOD_V`: page 6 – 8
- Budget and Budget Line Procedures
 - `ADD_BUDGET_LINE`: page 6 – 9
 - `BASELINE_BUDGET`: page 6 – 12
 - `CALCULATE_AMOUNTS`: page 6 – 13
 - `CREATE_DRAFT_BUDGET`: page 6 – 14
 - `DELETE_BUDGET_LINE`: page 6 – 19
 - `DELETE_DRAFT_BUDGET`: page 6 – 20

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- UPDATE_BUDGET: page 6 – 21
- UPDATE_BUDGET_LINE: page 6 – 25
- Load-Execute-Fetch Procedures
 - CLEAR_BUDGET: page 6 – 28
 - EXECUTE_CALCULATE_AMOUNTS: page 6 – 28
 - EXECUTE_CREATE_DRAFT_BUDGET: page 6 – 30
 - EXECUTE_UPDATE_BUDGET: page 6 – 31
 - FETCH_BUDGET_LINE: page 6 – 32
 - FETCH_CALCULATE_AMOUNTS: page 6 – 32
 - INIT_BUDGET: page 6 – 33
 - INIT_CALCULATE_AMOUNTS: page 6 – 34
 - LOAD_BUDGET_LINE: page 6 – 34

View Definitions

PA_BASE_BUDGET_BY_GL_PERIOD_V

PA_BASE_BUDGET_BY_GL_PERIOD_V is a view that holds the most recent baselined budget amounts by GL period.

Column descriptions for PA_BASE_BUDGET_BY_GL_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
BUDGET_VERSION_ID	NOT NULL	NUMBER(15)	The identification code of the budget version
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	The relevant GL period
PERIOD_YEAR	NOT NULL	NUMBER(15)	The corresponding GL year
PERIOD_START_DATE	NOT NULL	DATE	The start date of the GL period
PERIOD_END_DATE	NOT NULL	DATE	The end date of the GL period
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identifier of the resource list member
RESOURCE_ALIAS	NOT NULL	VARCHAR2(30)	The alias used for the resource
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list used in the budget
RESOURCE_ID	NOT NULL	NUMBER(15)	The identification code of the resource
RESOURCE_ACCUMULATED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether or not this budget has been summarized
RAW_COST	NULL	NUMBER	The budgeted raw cost
BURDENED_COST	NULL	NUMBER	The budgeted burdened cost
REVENUE	NULL	NUMBER	The budgeted revenue
QUANTITY	NULL	NUMBER	The budgeted quantity
LABOR_QUANTITY	NULL	NUMBER	The labor hours, if the resource is being tracked as labor
UNIT_OF_MEASURE	NULL	VARCHAR2(30)	The unit of measure for budget amounts for the resource

PA_BASE_BUDGET_BY_PA_PERIOD_V

PA_BASE_BUDGET_BY_PA_PERIOD_V is a view that holds the most recent baselined budget amounts by PA period.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column descriptions for PA_BASE_BUDGET_BY_PA_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
BUDGET_VERSION_ID	NOT NULL	NUMBER(15)	The reference code that identifies the budget version
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NULL	NUMBER(15)	The unique reference code that identifies the task within a project in Oracle Projects
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
PA_PERIOD	NOT NULL	VARCHAR2(20)	The relevant PA period
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	The corresponding GL period
PERIOD_YEAR	NOT NULL	NUMBER(15)	The corresponding GL year
PERIOD_START_DATE	NOT NULL	DATE	The start date of the PA period
PERIOD_END_DATE	NOT NULL	DATE	The end date of the PA period
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_ALIAS	NOT NULL	VARCHAR2(30)	The alias used for the resource
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list used in the budget
RESOURCE_ID	NOT NULL	NUMBER(15)	The identification code of the resource
RESOURCE_ACCUMULATED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether or not this budget has been summarized
RAW_COST	NULL	NUMBER	The budgeted raw cost
BURDENED_COST	NULL	NUMBER	The budgeted burdened cost
REVENUE	NULL	NUMBER	The budgeted revenue
QUANTITY	NULL	NUMBER	The budgeted quantity
LABOR_QUANTITY	NULL	NUMBER	The labor hours, if the resource is being tracked as labor
UNIT_OF_MEASURE	NULL	VARCHAR2(30)	The unit of measure for budget amounts for the resource

PA_BUDGET_CHANGE_REASON_V

PA_BUDGET_CHANGE_REASON_V is a view used to display valid budget change reason codes.

Column descriptions for PA_BUDGET_CHANGE_REASON_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies the reason to change a budget
NAME	NOT NULL	VARCHAR2(80)	The descriptive name of this change reason

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_BUDGET_ENTRY_METHODS_V

PA_BUDGET_ENTRY_METHODS_V is a view used to display valid budget entry methods.

Column descriptions for PA_BUDGET_ENTRY_METHODS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies this budget entry method
NAME	NOT NULL	VARCHAR2(30)	The name of this budget entry method
ENTRY_LEVEL_CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies the level of budget line entry
ENTRY_LEVEL_NAME	NOT NULL	VARCHAR2(80)	The descriptive name of the budget level entry
CATEGORIZATION_CODE	NOT NULL	VARCHAR2(30)	The indicator that specifies whether or not this method is categorized by resource
COST_QUANTITY_FLAG	NOT NULL	VARCHAR2(1)	The indicator that specifies that cost quantity is mandatory
RAW_COST_FLAG	NOT NULL	VARCHAR2(1)	The indicator that specifies that raw cost is mandatory
BURDENED_COST_FLAG	NOT NULL	VARCHAR2(1)	The indicator that specifies that burdened cost is mandatory
REV_QUANTITY_FLAG	NOT NULL	VARCHAR2(1)	The indicator that specifies that revenue quantity is mandatory
REVENUE_FLAG	NOT NULL	VARCHAR2(1)	The indicator that specifies that revenue is mandatory
TIME_PHASED_TYPE_CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies the time phasing
TIME_PHASED_TYPE_NAME	NOT NULL	VARCHAR2(80)	The name of this time phase code

PA_BUDGET_STATUS_CODES_V

PA_BUDGET_STATUS_CODES_V is a view used to display valid budget status codes.

Column descriptions for PA_BUDGET_STATUS_CODES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies this budget status
NAME	NOT NULL	VARCHAR2(80)	The name of this budget status

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_BUDGET_TYPES_V

PA_BUDGET_TYPES_V is a view used to display valid budget types.

Column descriptions for PA_BUDGET_TYPES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
CODE	NOT NULL	VARCHAR2(30)	The code that uniquely identifies this budget type
NAME	NOT NULL	VARCHAR2(30)	The name of this budget type

PA_ORIG_BUDGET_BY_GL_PERIOD_V

PA_ORIG_BUDGET_BY_GL_PERIOD_V is a view that holds the original budget amounts by GL period.

Column descriptions for PA_ORIG_BUDGET_BY_GL_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
BUDGET_VERSION_ID	NOT NULL	NUMBER(15)	The identification code of the budget version
PROJECT_ID	NOT NULL	NUMBER(15)	The unique reference code that identifies the project in Oracle Projects
TASK_ID	NULL	NUMBER(15)	The unique reference code that identifies the task within a project in Oracle Projects
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	The relevant GL period
PERIOD_YEAR	NOT NULL	NUMBER(15)	The corresponding GL year
PERIOD_START_DATE	NOT NULL	DATE	The start date of the GL period
PERIOD_END_DATE	NOT NULL	DATE	The end date of the GL period
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_ALIAS	NOT NULL	VARCHAR2(30)	The alias used for the resource
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identifier of the resource list used in the budget
RESOURCE_ID	NOT NULL	NUMBER(15)	The resource identifier
RESOURCE_ACCUMULATED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether or not this budget has been summarized
RAW_COST	NULL	NUMBER	The budgeted raw cost
BURDENED_COST	NULL	NUMBER	The budgeted burdened cost
REVENUE	NULL	NUMBER	The budgeted revenue
QUANTITY	NULL	NUMBER	The budgeted quantity
LABOR_QUANTITY	NULL	NUMBER	The labor hours, if the resource is being tracked as labor
UNIT_OF_MEASURE	NULL	VARCHAR2(30)	The unit of measure for budget amounts for the resource

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_ORIG_BUDGET_BY_PA_PERIOD_V

PA_ORIG_BUDGET_BY_PA_PERIOD_V is a view that holds the original budget amounts by PA period.

Column descriptions for PA_ORIG_BUDGET_BY_PA_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
BUDGET_VERSION_ID	NOT NULL	NUMBER(15)	The budget version identifier
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
PA_PERIOD	NOT NULL	VARCHAR2(20)	The relevant PA period
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	The corresponding GL period
PERIOD_YEAR	NOT NULL	NUMBER(15)	The corresponding GL year
PERIOD_START_DATE	NOT NULL	DATE	The start date of the PA period
PERIOD_END_DATE	NOT NULL	DATE	The end date of the PA period
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identifier of the resource list member
RESOURCE_ALIAS	NOT NULL	VARCHAR2(30)	The alias used for the resource
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list used in the budget
RESOURCE_ID	NOT NULL	NUMBER(15)	The identification code of the resource
RESOURCE_ACCUMULATED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether this budget has been summarized
RAW_COST	NULL	NUMBER	The budgeted raw cost
BURDENED_COST	NULL	NUMBER	The budgeted burdened cost
REVENUE	NULL	NUMBER	The budgeted revenue
QUANTITY	NULL	NUMBER	The budgeted quantity
LABOR_QUANTITY	NULL	NUMBER	The labor hours, if the resource is being tracked as labor
UNIT_OF_MEASURE	NULL	VARCHAR2(30)	The unit of measure for budget amounts for the resource

Procedure Definitions

ADD_BUDGET_LINE

ADD_BUDGET_LINE is a PL/SQL procedure used to add a budget line to a working budget in Oracle Projects for a given project and budget type.

Business Rules

- After you use ADD_BUDGET_LINE to create a draft budget and budget lines, save the data to the database before calling the API BASELINE_BUDGET. (A draft budget requires approval before you can baseline it.) For a revenue budget, enter the funding in Oracle Projects before you baseline the budget.
- We establish the following links between information stored in your external system and certain information in Oracle Projects, so you can pass the following parameters instead of their corresponding Oracle Projects identification codes.
 - For budgets:
P_PM_PROJECT_REFERENCE links to P_PA_PROJECT_ID.
P_RESOURCE_LIST_NAME links to
P_RESOURCE_LIST_ID.
 - For budget lines:
P_PM_TASK_REFERENCE links to P_PA_TASK_ID.
P_RESOURCE_ALIAS links to
P_RESOURCE_LIST_MEMBER_ID.
- The following pairs of parameters may both have NULL values if the budget is not categorized by resources, as defined by the budget entry method:
 - P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID
 - RESOURCE_ALIAS and RESOURCE_LIST_MEMBER_ID
- Specify values for the parameters PA_TASK_ID or PM_TASK_REFERENCE only when budgeting by tasks, as defined by the budget entry method.
- Specify values for the parameter PERIOD_NAME only when budgeting by PA or GL period, as defined by the budget entry method.

- If you budget by PA or GL period and do not provide a period name, Oracle Projects uses the budget start and end dates to select a valid period name from the database. If Oracle Projects fails to retrieve a valid period name, the API will abort.
- The task level at which you pass budget information should correspond to the level specified in the budget entry method. For example, if the budget entry method specifies that you can enter a budget only at the lowest task level, then ADD_BUDGET_LINE passes only lowest tasks.
- When the budget entry method (BEM) flags shown in the following table are set to N, do not pass the related parameters.

<i>BEM Flag</i>	<i>Related Parameter</i>
COST_QUANTITY_FLAG	QUANTITY
RAW_COST_FLAG	RAW_COST
BURDENED_COST_FLAG	BURDENED_COST
REV_QUANTITY_FLAG	QUANTITY
REVENUE_FLAG	REVENUE

Table 6 - 1 Parameters not to Pass if BEM Flags Are Set to N (Page 1 of 1)

- You can add a budget line only to a budget with a status of Working.

The following table shows the parameters for ADD_BUDGET_LINE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_PA_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of a resource
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	The identification code of the resource
P_BUDGET_START_DATE	IN	DATE	No	Start date of a budget line
P_BUDGET_END_DATE	IN	DATE	No	End date of a budget line
P_PERIOD_NAME	IN	VARCHAR2(30)	No	GL or PA period name
P_DESCRIPTION	IN	VARCHAR2(255)		(currently unavailable)
P_RAW_COST	IN	NUMBER	No	Budgeted raw cost amount
P_BURDENED_COST	IN	NUMBER	No	Budgeted burdened cost amount
P_REVENUE	IN	NUMBER	No	Budgeted revenue amount
P_QUANTITY	IN	NUMBER	No	Budgeted quantity
P_PM_BUDGET_LINE_REFERENCE	IN	VARCHAR2(30)	No	The reference code that identifies the budget line on the client side
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget line descriptive flexfield

Oracle Proprietary, Confidential Information--Use Restricted by Contract

BASELINE_BUDGET

BASELINE_BUDGET is a PL/SQL procedure used to baseline an existing budget in Oracle Projects for a given project and budget type.

Business Rules

- You must set up funding in Oracle Projects before you can baseline a revenue budget.
- If you have not yet submitted a budget, Oracle Projects submits it automatically before baselining it.
- You can submit a budget only if it contains budget lines.
- If no value (or an invalid value) is passed for the parameter P_MARK_AS_ORIGINAL, the default is N. When you baseline a budget for the first time, the P_MARK_AS_ORIGINAL is set to Y.

The following table shows the parameters for BASELINE_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_WORKFLOW_STARTED	OUT	VARCHAR2(1)		Shows if a workflow has been started (Y or N)
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_MARK_AS_ORIGINAL	IN	VARCHAR2(1)	No	Mark as original

CALCULATE_AMOUNTS

Using the PA_CLIENT_EXTN_BUDGET extension, you can use the public API CALCULATE_AMOUNTS to recalculate raw cost, burdened cost, and revenue amounts for existing budget lines. If P_UPDATE_DB_FLAG is set to Y, then the budget lines for the specified project will be updated upon the successful execution of this API.

Business Rules

- Since this API calls the PA_CLIENT_EXTN_BUDGET extension, you must modify the extension to calculate the amounts you want.
- To recalculate the corresponding amount, pass an uppercase Y for each calculation flag.
- Regardless of its update status, CALCULATE_AMOUNTS returns one row of amounts for each budget line it reads.
- To update the budget lines for a project with the amounts generated from CALCULATE_AMOUNTS, set P_UPDATE_DB_FLAG to an uppercase Y.

The following table shows the parameters for CALCULATE_AMOUNTS.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(25)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_CALC_RAW_COST_YN	IN	VARCHAR2(1)	No	Calculate raw cost (Y = Yes, N = No)

Name	Usage	Type	Req?	Description
P_CALC_BURDENED_COST_YN	IN	VARCHAR2(1)	No	Calculate burdened cost (Y = Yes, N = No)
P_CALC_REVENUE_YN	IN	VARCHAR2(1)	No	Calculate revenue (Y = Yes, N = No)
P_UPDATE_DB_FLAG	IN	VARCHAR2(1)	No	Update budget lines (Y = Yes, N = No)
P_CALC_BUDGET_LINES_OUT	IN	TABLE OF RECORD		
PA_TASK_ID	OUT	NUMBER		The reference code that uniquely identifies the task within a project in Oracle Projects
PM_TASK_REFERENCE	OUT	VARCHAR2(30)		The reference code that uniquely identifies the task in the external system
RESOURCE_ALIAS	OUT	VARCHAR2(30)		Alias of a resource
RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		The identification code of the resource
BUDGET_START_DATE	OUT	DATE		Start date of a budget
BUDGET_END_DATE	OUT	DATE		End date of a budget
PERIOD_NAME	OUT	VARCHAR2(30)		PA or GL period name
CALCULATED_RAW_COST	OUT	NUMBER		Calculated raw cost
CALCULATED_BURDENED_COST	OUT	NUMBER		Calculated burdened cost
CALCULATED_REVENUE	OUT	NUMBER		Calculated revenue
QUANTITY	OUT	NUMBER		Quantity
RETURN_STATUS	OUT	VARCHAR2(1)		API standard

CREATE_DRAFT_BUDGET

CREATE_DRAFT_BUDGET is a PL/SQL procedure used to create a draft budget and its budget lines in Oracle Projects for a given project, using a selected budget type and budget entry method.

This API uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Business Rules

- A draft budget requires approval before you can baseline it. After you use this API to create a draft budget and budget lines, save the data to the database before calling the API BASELINE_BUDGET. For a revenue budget, enter the funding in Oracle Projects before you can baseline the budget.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- We establish the following links between information stored in your system and certain information in Oracle Projects, so you can pass the following parameters instead of their corresponding Oracle Projects identification codes.
 - For budgets:
 - P_PM_PROJECT_REFERENCE links to P_PA_PROJECT_ID.
 - P_RESOURCE_LIST_NAME links to P_RESOURCE_LIST_ID.
 - For budget lines
 - P_PM_TASK_REFERENCE links to P_PA_TASK_ID.
 - P_RESOURCE_ALIAS links to P_RESOURCE_LIST_MEMBER_ID.
- Products that call budget APIs must specify their respective product codes. Oracle Projects predefines product codes and provides these codes to the appropriate vendors.
- The following pairs of parameters can both have NULL values if the budget is not categorized by resources, as defined by the budget entry method:
 - P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID
 - RESOURCE_ALIAS and RESOURCE_LIST_MEMBER_ID
- You can specify a value for the PA_TASK_ID or PM_TASK_REFERENCE parameter only when budgeting by tasks, as defined by the budget entry method.
- You can specify a value for the PERIOD_NAME parameter only when budgeting by PA or GL period, as defined by the budget entry method.
- If you budget by PA or GL period and do not provide a period name, Oracle Projects uses the budget start and end dates to select a valid period name from the database. If Oracle Projects fails to retrieve a valid period name, the API will abort.
- When budgeting by date range, you must provide the budget start and end dates. These dates may not overlap for a certain resource assignment.
- The task level at which you pass budget information should correspond to the level specified in the budget entry method. For example, if the budget entry method specifies that you can enter a budget only at the lowest task level, then this API passes only lowest tasks.

- When the budget entry method (BEM) flags shown in the following table are set to N, do not pass the related parameters.

<i>BEM Flag</i>	<i>Related Parameter</i>
COST_QUANTITY_FLAG	QUANTITY
RAW_COST_FLAG	RAW_COST
BURDENED_COST_FLAG	BURDENED_COST
REV_QUANTITY_FLAG	QUANTITY
REVENUE_FLAG	REVENUE

Table 6 – 2 Parameters not to Pass if BEM Flags Are Set to N (Page 1 of 1)

- Your budget entry method must reflect the needs of your external system.
- You can specify values for the parameters P_RAW_COST and P_BURDENED_COST amounts only for a cost budget, as defined by the budget type.
- You can specify a value for the parameter P_REVENUE_AMOUNT only for a revenue budget, as defined by the budget type.
- Passing the PL/SQL table P_BUDGET_LINES_TBL is optional. A draft budget does not require you to create budget lines simultaneously.
- If a draft budget already exists for a project and budget type, creating a new draft budget deletes the existing budget and budget lines.

The following table shows the parameters for CREATE_DRAFT_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_PM_BUDGET_REFERENCE	IN	VARCHAR2(30)	No	The reference code of the budget on the client side
P_BUDGET_VERSION_NAME	IN	VARCHAR2	No	The user-defined name for the budget version
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_CHANGE_REASON_CODE	IN	VARCHAR2(30)	No	The reference code that identifies the change reason
P_DESCRIPTION	IN	VARCHAR2(255)	No	Description of the budget
P_ENTRY_METHOD_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget entry method
P_RESOURCE_LIST_NAME	IN	VARCHAR2(60)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	The identification code of the resource list
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_BUDGET_LINES_IN	IN	TABLE OF RECORD		
PA_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects

Name	Usage	Type	Req?	Description
PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
RESOURCE_ALIAS	IN	VARCHAR2(30)	No	The alias of a resource
RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	The identification code of the resource
BUDGET_START_DATE	IN	DATE	No	Start date of budget line
BUDGET_END_DATE	IN	DATE	No	End date of budget line
PERIOD_NAME	IN	VARCHAR2(30)	No	GL or PA period name
DESCRIPTION	IN	VARCHAR2(255)	No	(currently unavailable)
RAW_COST	IN	NUMBER	No	Budgeted raw cost amount
BURDENED_COST	IN	NUMBER	No	Budgeted burdened cost amount
REVENUE	IN	NUMBER	No	Budgeted revenue amount
QUANTITY	IN	NUMBER	No	Budgeted quantity
PM_PRODUCT_CODE	IN	VARCHAR2(30)	No	The product code of the vendor of the external system
PM_BUDGET_LINE_REFERENCE	IN	VARCHAR2(30)	No	The reference code that identifies the budget line on client side
ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_BUDGET_LINES_OUT	OUT	TABLE OF RECORD		
RETURN_STATUS	OUT	VARCHAR2(1)		Return status

Oracle Proprietary, Confidential Information--Use Restricted by Contract

DELETE_BUDGET_LINE

DELETE_BUDGET_LINE is a PL/SQL procedure used to delete a budget line from a working budget in Oracle Projects for a given project and budget type.

Business Rules

This rule applies to the budget status that supports budget line deletion:

- You can delete only budget lines from working budgets. You cannot delete budget lines from baselined budgets.

This rule applies to the budget start date and period name:

- If values for P_START_DATE and P_PERIOD_NAME are not passed or are both passed as NULL, deleting a budget line deletes **all** the budget lines for the task/resource combination.

These rules apply to the budget entry method:

- Depending on the budget entry method, this API may require that you pass task and/or resource data.
- If budget APIs have passed no task data, Oracle Projects assumes that the budget entry method has specified uncategorized budgeting (budgets not tracked by resource) and project-level budgeting.
- If APIs pass both the P_START_DATE and the P_PERIOD_NAME to Oracle Projects, Oracle Projects uses the latter.

The following table shows the parameters for DELETE_BUDGET_LINE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects

Name	Usage	Type	Req?	Description
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_PA_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of a resource
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	The identification code of the resource
P_START_DATE	IN	DATE	No	The identification code of the budget line
P_PERIOD_NAME	IN	VARCHAR2(30)	No	The identification code of the budget line; overrules P_START_DATE

DELETE_DRAFT_BUDGET

DELETE_DRAFT_BUDGET is a PL/SQL procedure used to delete a working budget in Oracle Projects for a given project and budget type.

Business Rules

- You can delete working budgets only. You cannot delete baselined or submitted budgets.
- When you delete a budget, you also delete its budget lines and resource assignments.

The following table shows the parameters for DELETE_DRAFT_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type

UPDATE_BUDGET

UPDATE_BUDGET is a PL/SQL procedure used to update the working budget with its budget lines in Oracle Projects for a given project. This API updates existing budget lines or inserts new budget lines, depending on whether the budget lines already exist.

This API uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Business Rules

- A draft budget requires approval before you can baseline it. After you use this API to create a draft budget and budget lines, save the data to the database before calling the API BASELINE_BUDGET. For a revenue budget, you must enter the funding in Oracle Projects before you can baseline the budget.
- We establish links between information stored in your external system and certain information in Oracle Projects, so you can pass the following parameters instead of their corresponding Oracle Projects identifiers.
 - For budgets:
 - P_PM_PROJECT_REFERENCE links to P_PA_PROJECT_ID.
 - P_RESOURCE_LIST_NAME links to P_RESOURCE_LIST_ID.
 - For budget lines:
 - P_PM_TASK_REFERENCE links to P_PA_TASK_ID.
 - P_RESOURCE_ALIAS links to P_RESOURCE_LIST_MEMBER_ID.

- The following pairs of parameters can both have NULL values if the budget is not categorized by resources, as defined by the budget entry method:
 - P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID
 - RESOURCE_ALIAS and RESOURCE_LIST_MEMBER_ID
- You can specify values for the parameters PA_TASK_ID or PM_TASK_REFERENCE only when budgeting by tasks, as defined by the budget entry method.
- You can specify values for the parameter PERIOD_NAME only when budgeting by PA or GL period, as defined by the budget entry method.
- If you budget by PA or GL period and do not provide a period name, Oracle Projects uses the budget start date and budget end date to select a valid period name from the database. If Oracle Projects fails to retrieve a valid period name, the API will abort.
- The task level at which you pass budget information should correspond to the level specified in the budget entry method. For example, if the budget entry method specifies that you can enter a budget only at the lowest task level, then this API passes only lowest tasks.
- When the budget entry method flags shown in the following table are set to N, do not pass the related parameters.

<i>BEM Flag</i>	<i>Related Parameter</i>
COST_QUANTITY_FLAG	QUANTITY
RAW_COST_FLAG	RAW_COST
BURDENED_COST_FLAG	BURDENED_COST
REV_QUANTITY_FLAG	QUANTITY
REVENUE_FLAG	REVENUE

Table 6 – 3 Parameters not to Pass if BEM Flags Are Set to N (Page 1 of 1)

- You can add a budget line only to a budget with a status of Working.
- You cannot update a submitted budget.
- You can use this API only to update or add budget lines. To delete existing budget lines, use DELETE_BUDGET_LINE.

- Oracle Projects identifies a budget line by its budget start date, so you cannot update the budget start date.
- You can update only the budget header P_CHANGE_REASON_CODE and P_DESCRIPTION parameters.
- You can update only the following budget line parameters:
 - DESCRIPTION
 - RAW_COST
 - BURDENED_COST
 - REVENUE
 - QUANTITY
- You can pass flexfield parameters for both budget headers and budget lines. However, you can currently use flexfield parameters only to create new budget line rows.
- When passed, the parameters marked with an asterisk (*) in the following table identify the budget and budget line.

The following table shows the parameters for UPDATE_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID*	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE*	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE*	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_CHANGE_REASON_CODE	IN	VARCHAR2(30)	No	The reference code that identifies the change reason
P_DESCRIPTION	IN	VARCHAR2(255)	No	Description of the budget
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget descriptive flexfield

Oracle Proprietary, Confidential Information—Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_BUDGET_LINES_IN	IN	TABLE OF RECORD		
PA_TASK_ID*	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects
PM_TASK_REFERENCE*	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
RESOURCE_ALIAS*	IN	VARCHAR2(30)	No	Alias of a resource uniquely identifies the task in the external system
RESOURCE_LIST_MEMBER_ID*	IN	NUMBER	No	The identification code of the resource
BUDGET_START_DATE*	IN	DATE	No	Start date of budget line
BUDGET_END_DATE*	IN	DATE	No	End date of budget line
PERIOD_NAME*	IN	VARCHAR2(30)	No	GL or PA period name
DESCRIPTION	IN	VARCHAR2(255)	No	(currently unavailable)
RAW_COST	IN	NUMBER	No	Budgeted raw cost amount
BURDENED_COST	IN	NUMBER	No	Budgeted burdened cost amount
REVENUE	IN	NUMBER	No	Budgeted revenue amount
QUANTITY	IN	NUMBER	No	Budgeted quantity
PA_PRODUCT_CODE	IN	VARCHAR2(30)	No	The product code of the vendor of the external system
PM_BUDGET_LINE_REFERENCE	IN	VARCHAR2(30)	No	Reference code that identifies the budget line on the client side
ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget line descriptive flexfield

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_BUDGET_LINES_OUT	OUT	TABLE OF RECORD		Return status
RETURN_STATUS	OUT	VARCHAR2(1)	No	

UPDATE_BUDGET_LINE

UPDATE_BUDGET_LINE is a PL/SQL procedure used to update an existing budget line of a working budget in Oracle Projects for a given project and budget type.

Business Rules

- A draft budget requires approval before you can baseline it. After you use this API to create a draft budget and budget lines, you must save the data to the database before calling the API BASELINE_BUDGET. For a revenue budget, enter the funding in Oracle Projects before you baseline the budget.
- We establish links between information stored in your external system and certain information in Oracle Projects, so you can pass the following parameters instead of their corresponding Oracle Projects identification codes.
 - For budgets:
 - P_PM_PROJECT_REFERENCE links to P_PA_PROJECT_ID.
 - P_RESOURCE_LIST_NAME links to P_RESOURCE_LIST_ID.

Oracle Proprietary, Confidential Information—Use Restricted by Contract

- For budget lines:
P_PM_TASK_REFERENCE links to P_PA_TASK_ID.
P_RESOURCE_ALIAS links to
P_RESOURCE_LIST_MEMBER_ID.
- The following pairs of parameters can both have NULL values if the budget is not categorized by resources, as defined by the budget entry method:
 - P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID
 - RESOURCE_ALIAS and RESOURCE_LIST_MEMBER_ID
- You can specify values for the parameters PA_TASK_ID or PM_TASK_REFERENCE only when budgeting by tasks, as defined by the budget entry method.
- You can specify values for the parameter PERIOD_NAME only when budgeting by PA or GL period, as defined by the budget entry method.
- If you budget by PA or GL period and do not provide a period name, Oracle Projects uses the budget start date and budget end date to select a valid period name from the database. If Oracle Projects fails to retrieve a valid period name, the API will abort.
- The task level at which you pass budget information should correspond to the level specified in the budget entry method. For example, if the budget entry method specifies that you can enter a budget only at the lowest task level, then this API should pass only lowest tasks.
- When the budget entry method flags shown in the following table are set to N, do not pass the related parameters.

<i>BEM Flag</i>	<i>Related Parameter</i>
COST_QUANTITY_FLAG	QUANTITY
RAW_COST_FLAG	RAW_COST
BURDENED_COST_FLAG	BURDENED_COST
REV_QUANTITY_FLAG	QUANTITY
REVENUE_FLAG	REVENUE

Table 6 - 4 Parameters not to Pass if BEM Flags Are Set to N (Page 1 of 1)

- You can add a budget line only to a budget with a status of Working.

- Since Oracle Projects identifies a budget line by its budget start date, you cannot update the P_BUDGET_START_DATE. You can update only the budget line parameters below:
 - P_DESCRIPTION
 - P_RAW_COST
 - P_BURDENED_COST
 - P_REVENUE
 - P_QUANTITY
- When passed, the parameters marked with an asterisk (*) in the table below are used to identify the budget line.
- Although flexfield parameters appear in the parameter list below, this API does not currently use them to update budget line flexfields.

The following table shows the parameters for UPDATE_BUDGET_LINE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PA_PROJECT_ID*	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE*	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE*	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_PA_TASK_ID*	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE*	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_RESOURCE_ALIAS*	IN	VARCHAR2(30)	No	Alias of a resource
P_RESOURCE_LIST_MEMBER_ID*	IN	NUMBER	No	The identification code of the resource

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_BUDGET_START_DATE*	IN	DATE	No	Start date of budget line
P_BUDGET_END_DATE*	IN	DATE	No	End date of budget line
P_PERIOD_NAME*	IN	VARCHAR2(30)	No	GL or PA period name
P_DESCRIPTION	IN	VARCHAR2(255)		(currently unavailable)
P_RAW_COST	IN	NUMBER	No	Budgeted raw cost amount
P_BURDENED_COST	IN	NUMBER	No	Budgeted burdened cost amount
P_REVENUE	IN	NUMBER	No	Budgeted revenue amount
P_QUANTITY	IN	NUMBER	No	Budgeted quantity
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
P_ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget line descriptive flexfield

CLEAR_BUDGET

CLEAR_BUDGET is a Load-Execute-Fetch procedure used to clear the global data structures set up during the Initialize step.

EXECUTE_CALCULATE_AMOUNTS

EXECUTE_CALCULATE_AMOUNTS is a Load-Execute-Fetch procedure used to calculate the raw cost, burdened cost, and revenue amounts using existing budget lines for a given project and budget type.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

For each budget line, this API writes to globals that can be read by the API `FETCH_CALCULATE_AMOUNTS`.

Business Rules

- Since this API calls the `PA_CLIENT_EXTN_BUDGET` extension, you must modify the extension to calculate the amounts you want.
- You must pass an uppercase 'Y' for each calculation flag to recalculate the corresponding amount.
- Regardless of its update status, this API returns one row of amounts for each budget line it reads.
- To update the budget lines for a project with the calculated amounts generated from this API, you must set the `P_UPDATE_DB_FLAG` to an uppercase 'Y'.
- This API returns the total number of budget lines processed in the OUT parameter `P_TOT_BUDGET_LINES_CALCULATED`. This total determines how many times to call `FETCH_CALCULATE_AMOUNTS` in a loop.

The following table shows the parameters for `EXECUTE_CALCULATE_AMOUNTS`.

Name	Usage	Type	Req?	Description
<code>P_API_VERSION_NUMBER</code>	IN	NUMBER	Yes	API standard
<code>P_COMMIT</code>	IN	VARCHAR2(1)	No	API standard (default = 'F')
<code>P_INIT_MSG_LIST</code>	IN	VARCHAR2(1)	No	API standard (default = 'F')
<code>P_MSG_COUNT</code>	OUT	NUMBER		API standard
<code>P_MSG_DATA</code>	OUT	VARCHAR2(2000)		API standard
<code>P_RETURN_STATUS</code>	OUT	VARCHAR2(1)		API standard
<code>P_TOT_BUDGET_LINES_CALCULATED</code>	OUT	NUMBER		Indicates the total number of budget lines calculated and determines how many times to call the API <code>FETCH_CALCULATE_AMOUNTS</code>
<code>P_PM_PRODUCT_CODE</code>	IN	VARCHAR2(25)	Yes	The product code of the external system
<code>P_PA_PROJECT_ID</code>	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
<code>P_PM_PROJECT_REFERENCE</code>	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
<code>P_BUDGET_TYPE_CODE</code>	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_CALC_RAW_COST_YN	IN	VARCHAR2(1)	No	Calculate raw cost (Y or N)
P_CALC_BURDENED_COST_YN	IN	VARCHAR2(1)	No	Calculate burdened cost (Y or N)
P_CALC_REVENUE_YN	IN	VARCHAR2(1)	No	Calculate revenue (Y or N)
P_UPDATE_DB_FLAG	IN	VARCHAR2(1)	No	Update budget line (Y or N)

EXECUTE_CREATE_DRAFT_BUDGET

EXECUTE_CREATE_DRAFT_BUDGET is used to create a budget and its budget lines using the data stored in the global tables during the Load process.

The following table shows the parameters for EXECUTE_CREATE_DRAFT_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system
P_PM_BUDGET_REFERENCE	IN	VARCHAR2(30)	No	The reference code that identifies the budget on the client side
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_VERSION_NAME	IN	VARCHAR2	No	The user-defined name for the budget version
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_CHANGE_REASON_CODE	IN	VARCHAR2(30)	No	The reference code that identifies the change reason
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget descriptive flexfield

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget descriptive flexfield
P_DESCRIPTION	IN	VARCHAR2(255)	No	Description of the budget
P_ENTRY_METHOD_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget entry method
P_RESOURCE_LIST_NAME	IN	VARCHAR2(60)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	The identification code of the resource list

EXECUTE_UPDATE_BUDGET

EXECUTE_UPDATE_BUDGET is a Load–Execute–Fetch procedure used to update a budget and its budget lines using the data stored in the global tables during the Load process.

The following table shows the parameters for EXECUTE_UPDATE_BUDGET.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(10)	Yes	The product code of the vendor of the external system

Name	Usage	Type	Req?	Description
P_PA_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_BUDGET_TYPE_CODE	IN	VARCHAR2(30)	Yes	The reference code that identifies the budget type
P_CHANGE_REASON_CODE	IN	VARCHAR2(30)	No	The reference code that identifies the change reason
P_DESCRIPTION	IN	VARCHAR2(255)	No	Description of the budget

FETCH_BUDGET_LINE

FETCH_BUDGET_LINE is a Load-Execute-Fetch procedure used to retrieve the return status returned during the creation of a budget line from a global PL/SQL table.

The following table shows the parameters for FETCH_BUDGET_LINE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)		API standard (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_LINE_INDEX	IN	NUMBER	Yes	Pointer to specific budget line
P_LINE_RETURN_STATUS	OUT	VARCHAR2(1)		Return status for specific line

FETCH_CALCULATE_AMOUNTS

FETCH_CALCULATE_AMOUNTS is a Load-Execute-Fetch procedure used to get the raw cost, burdened cost, and revenue amounts by budget line from global records updated by the API EXECUTE_CALCULATE_AMOUNTS.

Business Rule

- Call this API in a loop for each calculated budget line using the API EXECUTE_CALCULATE_AMOUNTS. The value the API EXECUTE_CALCULATE_AMOUNTS returns for

Oracle Proprietary, Confidential Information--Use Restricted by Contract

P_TOT_BUDGET_LINES_CALCULATED determines how many times to call this API.

The following table shows the parameters for FETCH_CALCULATE_AMOUNTS.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_LINE_INDEX	OUT	NUMBER		Pointer to specific budget line
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PA_TASK_ID	OUT	NUMBER		The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE	OUT	VARCHAR2(30)		The reference code that uniquely identifies the task in the external system
P_BUDGET_START_DATE	OUT	DATE		Start date of budget line
P_BUDGET_END_DATE	OUT	DATE		End date of budget line
P_PERIOD_NAME	OUT	VARCHAR2(20)		PA or GL period name
P_RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		The identification code of the resource
P_QUANTITY	OUT	NUMBER		The quantity entered into the budget line
P_RESOURCE_ALIAS	OUT	VARCHAR2(30)		Alias of resource
P_CALCULATED_RAW_COST	OUT	NUMBER		Calculated raw cost
P_CALCULATED_BURDENED_COST	OUT	NUMBER		Calculated burdened cost
P_CALCULATED_REVENUE	OUT	NUMBER		Calculated revenue
P_LINE_RETURN_STATUS	OUT	VARCHAR2(1)		Return status for a specific line

INIT_BUDGET

INIT_BUDGET is a Load-Execute-Fetch procedure used to set up the global data structures that other Load-Execute-Fetch procedures use to create a new or update an existing draft budget in Oracle Projects.

INIT_CALCULATE_AMOUNTS

INIT_CALCULATE_AMOUNTS is a Load-Execute-Fetch procedure used to set up the global data structures used by the Load-Execute-Fetch API CALCULATE_AMOUNTS.

LOAD_BUDGET_LINE

LOAD_BUDGET_LINE is a Load-Execute-Fetch procedure used to load a budget line to a global PL/SQL table.

The following table shows the parameters for LOAD_BUDGET_LINE.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)	No	API standard
P_PA_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of a resource
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	The identification code of the resource
P_BUDGET_START_DATE	IN	DATE	No	Start date of budget line
P_BUDGET_END_DATE	IN	DATE	No	End date of budget line
P_PERIOD_NAME	IN	VARCHAR2(30)	No	PA or GL period name
P_DESCRIPTION	IN	VARCHAR2(255)	No	Description of the budget
P_RAW_COST	IN	NUMBER	No	Budgeted raw cost amount
P_BURDENED_COST	IN	NUMBER	No	Budgeted burdened cost amount
P_REVENUE	IN	NUMBER	No	Budgeted revenue amount
P_QUANTITY	IN	NUMBER	No	Budgeted quantity
PM_PRODUCT_CODE	IN	VARCHAR2(30)	No	The product code of the vendor of the external system
PM_BUDGET_LINE_REFERENCE	IN	VARCHAR2(30)	No	The reference code that identifies the budget line on the client side
ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Used by descriptive flexfields
ATTRIBUTE1	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE2	IN	VARCHAR2(150)	No	Budget line descriptive flexfield

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
ATTRIBUTE3	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE4	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE5	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE6	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE7	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE8	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE9	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE10	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE11	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE12	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE13	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE14	IN	VARCHAR2(150)	No	Budget line descriptive flexfield
ATTRIBUTE15	IN	VARCHAR2(150)	No	Budget line descriptive flexfield

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHAPTER

7

Detailed Design -- Resource APIs

This chapter describes how to implement AMG APIs for resource list and resource list member information, and provides detailed view and procedure definitions.

Resource APIs

You can keep track of and organize both labor and non-labor resources using the system that you prefer. Then, use the resource APIs to export your resource lists and the resources they include to Oracle Projects. Oracle Projects updates its resource information accordingly. As your resources and resource lists change, update the information in your system and periodically synchronize the two systems.

Note: When you call any resource API that requires a resource list identifier, pass either the `P_RESOURCE_LIST_NAME` or the `P_RESOURCE_LIST_ID` parameter to identify the resource list. When you call any resource API that requires a resource identifier, pass either the `P_RESOURCE_ALIAS` or the `P_RESOURCE_LIST_MEMBER_ID` parameter to identify the resource.

For introductory remarks about this chapter, see Overview of Detailed Design: page 4 – 2. The views and procedures discussed in this section are listed below. The procedures are located in the public API package `PA_RESOURCE_PUB`.

- Views
 - `PA_AMG_RESOURCE_INFO_V`: page 7 – 4
 - `PA_EMPLOYEES_RES_V`: page 7 – 5
 - `PA_EVENT_TYPES_RES_V`: page 7 – 5
 - `PA_EXPEND_CATEGORIES_RES_V`: page 7 – 6
 - `PA_EXPENDITURE_TYPES_RES_V`: page 7 – 6
 - `PA_JOBS_RES_V`: page 7 – 7
 - `PA_LOWEST_LEVEL_RESOURCES`: page 7 – 8
 - `PA_ORGANIZATIONS_RES_V`: page 7 – 9
 - `PA_PROJ_ORG_STRUCTURES_V`: page 7 – 9
 - `PA_QRY_RESOURCE_LISTS_V`: page 7 – 10
 - `PA_QUERY_RES_LIST_MEMBERS_V`: page 7 – 10
 - `PA_RESOURCE_LIST_GROUPS_V`: page 7 – 11
 - `PA_RESOURCE_LIST_V`: page 7 – 12
 - `PA_RESOURCE_TYPES_ACTIVE_V`: page 7 – 12

- PA_REVENUE_CATEGORIES_RES_V: page 7 – 13
- PA_VENDORS_RES_V: page 7 – 13
- Resource List and Resource List Member Procedures
 - ADD_RESOURCE_LIST_MEMBER: page 7 – 15
 - CREATE_RESOURCE_LIST: page 7 – 16
 - DELETE_RESOURCE_LIST: page 7 – 18
 - DELETE_RESOURCE_LIST_MEMBER: page 7 – 19
 - SORT_RESOURCE_LIST_MEMBERS: page 7 – 19
 - UPDATE_RESOURCE_LIST: page 7 – 20
 - UPDATE_RESOURCE_LIST_MEMBER: page 7 – 22
- Load-Execute-Fetch Procedures
 - CLEAR_CREATE_RESOURCE_LIST: page 7 – 23
 - CLEAR_UPDATE_MEMBERS: page 7 – 23
 - EXEC_CREATE_RESOURCE_LIST: page 7 – 23
 - EXEC_UPDATE_RESOURCE_LIST: page 7 – 24
 - FETCH_MEMBERS: page 7 – 24
 - FETCH_RESOURCE_LIST: page 7 – 25
 - INIT_CREATE_RESOURCE_LIST: page 7 – 25
 - INIT_UPDATE_MEMBERS: page 7 – 25
 - LOAD_MEMBERS: page 7 – 25
 - LOAD_RESOURCE_LIST: page 7 – 27

View Definitions

PA_AMG_RESOURCE_INFO_V

You can customize this view to retrieve additional information about valid resource list members, such as cost rates or overtime rates.

Column descriptions for PA_AMG_RESOURCE_INFO_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PRODUCT_CODE	NULL	VARCHAR2(9)	The product used to access AMG
RESOURCE_LIST_MEMBER_ID	NULL	NUMBER	Identification of the member in the resource list
POPULATE_TEXT3_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column text3 is populated or not
TEXT3	NULL		Source to be decided by implementation team
POPULATE_TEXT4_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column text4 is populated or not
TEXT4	NULL		Source to be decided by implementation team
POPULATE_TEXT5_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column text5 is populated or not
TEXT5	NULL		Source to be decided by implementation team
POPULATE_TEXT6_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column text6 is populated or not
TEXT6	NULL		Source to be decided by implementation team
POPULATE_COST1_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column cost1 is populated or not
COST1	NULL		Source to be decided by implementation team
POPULATE_COST2_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column cost2 is populated or not
COST2	NULL		Source to be decided by implementation team
POPULATE_COST3_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column cost3 is populated or not
COST3	NULL		Source to be decided by implementation team
POPULATE_COST4_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column cost4 is populated or not
COST4	NULL		Source to be decided by implementation team
POPULATE_OVERTIME_RT_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column populate_overtime_rt_col is populated or not
OVERTIME_RATE	NULL		Source to be decided by implementation team
POPULATE_COST_PER_USE_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column cost_per_use is populated or not

Oracle Proprietary, Confidential Information--Use Restricted by Contract

COST_PER_USE	NULL		Source to be decided by implementation team
POPULATE_STANDARD_RT_COL_FLAG	NULL	VARCHAR2(1)	Flag indicating whether column standard_rate is populated or not
STANDARD_RATE	NULL		Source to be decided by implementation team

PA_EMPLOYEES_RES_V

PA_EMPLOYEES_RES_V is a view that displays information about all employees defined in your human resources application. You can define any employee returned by this view as a resource in Oracle Projects.

Column descriptions for PA_EMPLOYEES_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PERSON_ID		NUMBER(9)	Holds the employee or person identification code of the employee
EMPLOYEE_NAME		VARCHAR2(60)	Holds the name of the employee
EMPLOYEE_NUMBER		VARCHAR2(30)	Holds the number of the employee
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether this is to be tracked as a labor resource
UNIT_OF_MEASURE		VARCHAR2(5)	Indicates the unit of measure for this resource
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource

PA_EVENT_TYPES_RES_V

PA_EVENT_TYPES_RES_V is a view that displays all event types defined in Oracle Projects. You can define any event type returned by this view as a resource in Oracle Projects.

Column descriptions for PA_EVENT_TYPES_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
EVENT_TYPE		VARCHAR2(30)	Holds the implementation-defined name that uniquely identifies the event type

Oracle Proprietary, Confidential Information--Use Restricted by Contract

DESCRIPTION	VARCHAR2(250)	Holds the description of the event type
EVENT_TYPE_CLASSIFICATION	VARCHAR2(30)	The classification of the event type to drive system processing of events of the event type
REVENUE_CATEGORY_CODE	VARCHAR2(30)	The revenue category to which the event belongs
TRACK_AS_LABOR_FLAG	VARCHAR2(1)	Indicates whether the resource is to be tracked as labor (default = 'N')
UNIT_OF_MEASURE	VARCHAR2(0)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG	VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource (default = 'N')

PA_EXPEND_CATEGORIES_RES_V

PA_EXPEND_CATEGORIES_RES_V is a view that displays all expenditure categories defined in Oracle Projects. You can define any expenditure category returned by this view as a resource in Oracle Projects.

Column descriptions for PA_EXPEND_CATEGORIES_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
EXPENDITURE_CATEGORY		VARCHAR2(30)	Holds the implementation-defined name that uniquely identifies the expenditure category
DESCRIPTION		VARCHAR2(250)	Holds the description of the expenditure category
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor (no default value)
UNIT_OF_MEASURE		VARCHAR2(5)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource (default = 'N')

PA_EXPENDITURE_TYPES_RES_V

PA_EXPENDITURE_TYPES_RES_V is a view that displays all the expenditure types defined in Oracle Projects. You can define any expenditure type returned by this view as a resource in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column descriptions for PA_EXPENDITURE_TYPES_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
EXPENDITURE_TYPE		VARCHAR2(30)	Holds the name of the expenditure type
DESCRIPTION		VARCHAR2(250)	Holds the description of the expenditure type
EXPENDITURE_CATEGORY		VARCHAR2(30)	The expenditure category that classifies the expenditure type into a cost group
REVENUE_CATEGORY_CODE		VARCHAR2(30)	The revenue category that classifies the expenditure type into a revenue group
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether this resource is to be tracked as labor
UNIT_OF_MEASURE		VARCHAR2(30)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource

PA_JOBS_RES_V

PA_JOBS_RES_V is a view that displays information about all the jobs defined in your human resources application. You can define any job returned by this view as a resource in Oracle Projects.

Column descriptions for PA_JOBS_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
JOB_ID		NUMBER(9)	Holds the identification code of the job
JOB_NAME		VARCHAR2(60)	Holds the name of the job
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor. Default value for a job resource is 'Y'.
UNIT_OF_MEASURE		VARCHAR2(5)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Rollup quantity flag. Default value for a job resource is 'N'.

PA_LOWEST_LEVEL_RESOURCES_V

PA_LOWEST_LEVEL_RESOURCES_V is a view that retrieves Oracle Projects identification codes and names for resource lists and lowest-level resource list members.

Column descriptions for PA_LOWEST_LEVEL_RESOURCES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_LIST_ID		NUMBER	The reference code that uniquely identifies the resource list
RESOURCE_LIST_NAME		VARCHAR2(60)	The name of the resource list
RESOURCE_LIST_MEMBER_ID		NUMBER	The reference code that uniquely identifies the resource list member
RESOURCE_ID		NUMBER	The identification code of the resource that is being used as a member of the resource list
RESOURCE_ALIAS		VARCHAR2(30)	The alias of the resource
RESOURCE_NAME		VARCHAR2(60)	The name of the resource
RESOURCE_TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor

PA_ORGANIZATIONS_RES_V

PA_ORGANIZATIONS_RES_V is a view that displays information about the organizations defined in your human resources application. You can define any organization returned by this view as a resource in Oracle Projects.

Column descriptions for PA_ORGANIZATIONS_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
ORGANIZATION_ID		NUMBER(15)	Holds the identification code of the Organization
ORGANIZATION_NAME		VARCHAR2(60)	Holds the name of the organization
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor. Default value for an organization resource is 'Y'.
UNIT_OF_MEASURE		VARCHAR2(5)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource. Default value for an organization resource is 'N'.

PA_PROJ_ORG_STRUCTURES_V

PA_PROJ_ORG_STRUCTURES_V is a view that retrieves the organization hierarchy.

Column descriptions for PA_PROJ_ORG_STRUCTURES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
BUSINESS_GROUP_ID		NUMBER	The identification code of the business group
ORGANIZATION_ID		NUMBER	The identification code of the organization
PARENT_ORG_ID		NUMBER	The identification code of the parent organization
ORGANIZATION_NAME		VARCHAR2(60)	The name of the organization
PARENT_ORG_NAME		VARCHAR2(60)	The name of the parent organization
ORG_STRUCTURE_VERSION_ID		NUMBER	The reference code that identifies the organization structure version
PROJECT_ORG_FLAG		VARCHAR2(2000)	Project organization flag

PA_QRY_RESOURCE_LISTS_V

PA_QRY_RESOURCE_LISTS_V is a view that retrieves all the resource lists defined in Oracle Projects.

Column descriptions for PA_QRY_RESOURCE_LISTS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	Holds the reference code that uniquely identifies the resource list
RESOURCE_LIST_NAME	NOT NULL	VARCHAR2(60)	Holds the name for the resource list
GROUPED_BY_RESOURCE_TYPE	NOT NULL	VARCHAR2(30)	Holds the resource type code that indicates how the resource list is grouped. Valid values are EXPENDITURE_CATEGORY, REVENUE_CATEGORY, and ORGANIZATION. If the resource list is not grouped, the value is NONE.
GROUPED_BY_RESOURCE_TYPE_ID	NOT NULL	NUMBER	Holds the identification code of the resource type by which the resource list is grouped
DESCRIPTION	NOT NULL	VARCHAR2(255)	Holds the description of the resource list
START_DATE	NOT NULL	DATE	Holds the effective start date of the resource list
END_DATE	NOT NULL	DATE	Holds the end date through which the resource list is effective

PA_QUERY_RES_LIST_MEMBERS_V

PA_QUERY_RES_LIST_MEMBERS_V is a view that retrieves all the members of a resource list defined in Oracle Projects.

Column descriptions for PA_QUERY_RES_LIST_MEMBERS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	Holds the reference code that uniquely identifies the resource list
RESOURCE_LIST_NAME	NOT NULL	VARCHAR2(60)	Holds the name of the resource list
RESOURCE_NAME	NOT NULL	VARCHAR2(60)	Holds the name of the resource
RESOURCE_GROUP_ALIAS	NOT NULL	VARCHAR2(30)	Holds the alias of the resource group

Oracle Proprietary, Confidential Information--Use Restricted by Contract

RESOURCE_GROUP_MEMBER_ID	NULL	NUMBER	Holds the unique identification code of the resource group member
ALIAS	NOT NULL	VARCHAR2(30)	Holds the alias of the resource list member
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER	Holds the reference code that uniquely identifies the resource list member
SORT_ORDER	NOT NULL	NUMBER	Holds the sorting order of the resource list member
ENABLED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether the resource list member has been enabled
RESOURCE_TYPE_CODE	NOT NULL	VARCHAR2(30)	Holds the resource type of the resource
EMPLOYEE_LAST_NAME	NULL	VARCHAR2(60)	Holds the last name of the employee if the resource is of the type Employee
EMPLOYEE_FIRST_NAME	NULL	VARCHAR2(60)	Holds the first name of the employee if the resource is of the type Employee
EMPLOYEE_ORG_NAME	NULL	VARCHAR2(60)	Holds the organization name of the employee if the resource is of the type Employee
EXPENDITURE_ORG_NAME	NULL	VARCHAR2(60)	Holds the name of the organization if the resource is of the type Organization
SUPPLIER_NAME	NULL	VARCHAR2(60)	Holds the name of the vendor if the resource is of the type Vendor
JOB_NAME	NULL	VARCHAR2(60)	Holds the job name if the resource is of the type Job
EVENT_TYPE	NULL	VARCHAR2(60)	Holds the event type if the resource is of the type Event Type
EXPENDITURE_TYPE	NULL	VARCHAR2(60)	Holds the expenditure type if the resource is of the type Expenditure Type
EXPENDITURE_CATEGORY	NULL	VARCHAR2(60)	Holds the expenditure category if the resource is of the type Expenditure Category
REVENUE_CATEGORY	NULL	VARCHAR2(60)	Holds the revenue category if the resource is of the type Revenue Category

PA_RESOURCE_LIST_GROUPS_V

PA_RESOURCE_LIST_GROUPS_V is a view that retrieves all the resource groups within a resource list defined in Oracle Projects.

Column descriptions for PA_RESOURCE_LIST_GROUPS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	Holds the reference code that uniquely identifies the resource list
RESOURCE_LIST_NAME	NOT NULL	VARCHAR2(60)	Holds the name for the resource list

Oracle Proprietary, Confidential Information--Use Restricted by Contract

RESOURCE_GROUP_ALIAS	NOT NULL	VARCHAR2(30)	Holds the alias of the resource group
RESOURCE_GROUP_MEMBER_ID	NOT NULL	NUMBER(15)	Holds the RESOURCE_LIST_MEMBER_ID of the resource group
SORT_ORDER	NOT NULL	NUMBER(15)	Holds the sorting order of the resource group within the resource list
ENABLED_FLAG	NOT NULL	VARCHAR2(1)	Indicates whether the resource group has been enabled

PA_RESOURCE_LIST_V

PA_RESOURCE_LIST_V is a view that retrieves all the resource lists defined in Oracle Projects.

Column descriptions for PA_RESOURCE_LIST_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_LIST_ID		NUMBER	The reference code that uniquely identifies the resource list
RESOURCE_LIST_NAME		VARCHAR2(60)	The name for the resource list
RESOURCE_LIST_MEMBER_ID		NUMBER	The identification code for the resource list member
RESOURCE_ID		NUMBER	The identification code of the resource that is being used as a member of the resource list
RESOURCE_ALIAS		VARCHAR2(30)	Resource alias
RESOURCE_NAME		VARCHAR2(60)	The name of the resource
RESOURCE_TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor

PA_RESOURCE_TYPES_ACTIVE_V

PA_RESOURCE_TYPES_ACTIVE_V is a view that retrieves all the active resource types defined in Oracle Projects.

Column descriptions for PA_RESOURCE_TYPES_ACTIVE_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
RESOURCE_TYPE_ID		NUMBER(15)	Holds the reference code that identifies the resource type
RESOURCE_TYPE_NAME		VARCHAR2(60)	Holds the name of the resource type
RESOURCE_CLASS_CODE		VARCHAR2(30)	Foreign key from PA_LOOKUPS

Oracle Proprietary, Confidential Information--Use Restricted by Contract

RESOURCE_TYPE_CODE	VARCHAR2(30)	Holds the code used to identify the resource type. The supported resource type codes are EMPLOYEE, JOB, EXPENDITURE_TYPE, EVENT_TYPE, EXPENDITURE_CATEGORY, REVENUE_CATEGORY, and VENDOR.
GROUP_FLAG	VARCHAR2(1)	Indicates that this resource type can be used as a group
DESCRIPTION	VARCHAR2(255)	Description of the resource type
TABLE_NAME	VARCHAR2(30)	Source table/view that holds the resource table information
ACCESS_KEY	VARCHAR2(30)	Key column through which the table is to be accessed
SQL_TEXT	VARCHAR2(2000)	Text containing all of the required column names

PA_REVENUE_CATEGORIES_RES_V

PA_REVENUE_CATEGORIES_RES_V is a view that displays all revenue categories defined in Oracle Projects. You can define any revenue category returned by this view as a resource in Oracle Projects.

Column descriptions for PA_REVENUE_CATEGORIES_RES_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
REVENUE_CATEGORY_CODE		VARCHAR2(30)	Holds the reference code that identifies the revenue category
DESCRIPTION		VARCHAR2(250)	Holds the description of the revenue category
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor (no default value)
UNIT_OF_MEASURE		VARCHAR2(5)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource (default = 'N')

PA_VENDORS_RES_V

PA_VENDORS_RES_V is a view that displays information pertaining to all vendors defined in Oracle Purchasing. You can define any vendor returned by this view as a resource in Oracle Projects.

Column descriptions for PA_VENDORS_RES_V are shown in the following table.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column Descriptions

<u>Name</u>	<u>Null?</u>	<u>Type</u>	<u>Description</u>
VENDOR_ID		NUMBER(15)	Holds the reference code that identifies the vendor
VENDOR_NAME		VARCHAR2(60)	Holds the name of the vendor
TRACK_AS_LABOR_FLAG		VARCHAR2(1)	Indicates whether the resource is to be tracked as labor
UNIT_OF_MEASURE		VARCHAR2(0)	Holds the unit of measure
ROLLUP_QUANTITY_FLAG		VARCHAR2(1)	Indicates whether the quantity is to be rolled up to higher level tasks while summarizing budgets and actuals against this resource

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Procedures

ADD_RESOURCE_LIST_MEMBER

ADD_RESOURCE_LIST_MEMBER is a PL/SQL procedure that adds a resource member to an existing resource list.

Business Rules

1. Calling modules can pass either the RESOURCE_LIST_NAME or the RESOURCE_LIST_ID.
2. If the calling modules pass both RESOURCE_LIST_NAME and RESOURCE_LIST_ID, the API uses the latter.
3. If the resource list is grouped, you must pass a valid resource group alias.
4. The value for P_RESOURCE_ATTR_VALUE must correspond to the value for P_RESOURCE_TYPE. For example, the person identification code for P_RESOURCE_ATTR_VALUE must be valid if P_RESOURCE_TYPE equals EMPLOYEE.
5. If the calling module passes information for both RESOURCE_GROUP and RESOURCE_MEMBER parameters to this API, the API first verifies that the resource group exists. If the resource group does not exist, the API creates the resource group and then creates the resource.
6. If a given resource member already exists, this API does not return an error. Instead, it returns a successful return status and the RESOURCE_LIST_MEMBER_ID of the existing resource member.

Note: Because you can store only one transaction attribute for a given resource, this API accepts only a single RESOURCE_ATTR_VALUE, which may hold PERSON_ID, JOB_ID, and so on.

The following table shows the parameters for ADD_RESOURCE_LIST_MEMBER.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of resource list

Oracle Proprietary, Confidential Information—Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_RESOURCE_LIST_ID	IN	NUMBER	No	The identification code of the resource list
P_RESOURCE_GROUP_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource group
P_RESOURCE_GROUP_NAME	IN	VARCHAR2(80)	No	Name of the resource group
P_RESOURCE_TYPE_CODE	IN	VARCHAR2(30)	Yes	Type code of the resource
P_RESOURCE_ATTR_VALUE	IN	VARCHAR2(80)	Yes	Attribute value of the resource
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	Yes	Alias of the resource member
P_SORT_ORDER	IN	NUMBER	No	Sort order of the resource member
P_ENABLED_FLAG	IN	VARCHAR2(1)	No	Enabled flag of the resource member (default = 'Y')
P_RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		The identification code of the resource member on a specific resource list
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

CREATE_RESOURCE_LIST

CREATE_RESOURCE_LIST is a PL/SQL procedure that creates a resource list and optionally creates the resource list members.

This API uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Business Rules

- Valid values for P_GROUP_RESOURCE_TYPE are EXPENDITURE_CATEGORY, REVENUE_CATEGORY, ORGANIZATION, and NONE.
- The resource list name must be unique.
- If calling programs pass the P_MEMBER_TBL (optional), this API creates the relevant resource list member records.
- If your resource list is grouped, you must pass a valid resource group alias.
- The value for P_RESOURCE_ATTR_VALUE must correspond with the value for P_RESOURCE_TYPE. For example, the

Oracle Proprietary, Confidential Information--Use Restricted by Contract

person identification code for P_RESOURCE_ATTR_VALUE must be valid if P_RESOURCE_TYPE equals EMPLOYEE.

- If the value for GROUP_RESOURCE_TYPE is NONE, this API will ignore resource group IN parameters.
- If you do not specify the resource group alias, the group resource type must be NONE.

The following table shows the parameters for CREATE_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RESOURCE_LIST_REC	IN	RECORD		
RESOURCE_LIST_NAME	IN	VARCHAR2(80)	Yes	Name of resource list
DESCRIPTION	IN	VARCHAR2(255)	No	Description of resource list
GROUP_RESOURCE_TYPE	IN	VARCHAR2(30)	Yes	Type of resource group
START_DATE	IN	DATE	No	Start date of resource list
END_DATE	IN	DATE	No	End date of resource list
P_RESOURCE_LIST_OUT_REC	OUT	RECORD		API standard
RESOURCE_LIST_ID	OUT	NUMBER		The reference code that uniquely identifies the resource list
RETURN_STATUS	OUT	VARCHAR2(1)		Return status of specific resource list
P_MEMBER_TBL	IN	TABLE OF RECORD		
RESOURCE_GROUP_ALIAS	IN	VARCHAR2(30)	No	Alias of resource group
RESOURCE_GROUP_NAME	IN	VARCHAR2(80)	No	Name of resource group
RESOURCE_TYPE_CODE	IN	VARCHAR2(30)	Yes	Type code of resource member
RESOURCE_ATTR_VALUE	IN	VARCHAR2(80)	Yes	Attribute value of resource member
RESOURCE_ALIAS	IN	VARCHAR2(30)	Yes	Alias of resource member
SORT_ORDER	IN	NUMBER	No	Sort order of resource member
ENABLED_FLAG	IN	VARCHAR2(1)	No	Enabled flag for resource member (default = 'Y')
P_MEMBER_OUT_TBL	OUT	TABLE OF RECORD		

Name	Usage	Type	Req?	Description
RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		The identification code of resource member on a specific resource list
RETURN_STATUS	OUT	VARCHAR2(1)		Return status of specific resource member

DELETE_RESOURCE_LIST

DELETE_RESOURCE_LIST is a PL/SQL procedure that deletes a given resource list.

Business Rules

- Calling modules can pass either the P_RESOURCE_LIST_NAME or the P_RESOURCE_LIST_ID.
- If calling modules pass both P_RESOURCE_LIST_NAME and the P_RESOURCE_LIST_ID, this API uses the latter.
- You cannot delete a resource list if:
 - You summarize project actuals by that resource list.
 - A budget uses that resource list.
 - The list contains resource list members.

The following table shows the parameters for DELETE_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of the resource list
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

DELETE_RESOURCE_LIST_MEMBER

DELETE_RESOURCE_LIST_MEMBER is a PL/SQL procedure that deletes a given resource list member.

Business Rules

- Calling modules can pass either the P_RESOURCE_LIST_NAME or the P_RESOURCE_LIST_ID. Calling modules can also pass the P_ALIAS or the P_ALIAS_MEMBER_ID.
- If the calling modules pass both P_RESOURCE_LIST_NAME and the P_RESOURCE_LIST_ID, this API uses the latter.
- You cannot delete a resource list member if:
 - You summarize project actuals by that resource list member.
 - A budget uses that resource list member.

The following table shows the parameters for DELETE_RESOURCE_LIST_MEMBER.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of the resource list
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource list
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	Identification code of the resource list member
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

SORT_RESOURCE_LIST_MEMBERS

SORT_RESOURCE_LIST_MEMBERS is a PL/SQL procedure that updates the sort order for resource members in a given resource list.

Business Rules

- Calling modules can pass either the P_RESOURCE_LIST_NAME or the P_RESOURCE_LIST_ID.
- If the calling modules pass both the P_RESOURCE_LIST_NAME and the P_RESOURCE_LIST_ID, this API uses the latter.
- If you specify a resource group alias, this API sorts only resources below that resource group. Otherwise, this API sorts all resources in the resource list.
- You can sort resources by alias or resource name. Valid values for P_SORT_BY PARAMETER are ALIAS and RESOURCE_NAME.

The following table shows the parameters for SORT_RESOURCE_LIST_MEMBERS.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of the resource list
P_RESOURCE_GROUP_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource group
P_SORT_BY	IN	VARCHAR2(30)	Yes	Sort-by code
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

UPDATE_RESOURCE_LIST

UPDATE_RESOURCE_LIST is a PL/SQL procedure that updates an existing resource list, including updating existing or adding new resource list members.

This API uses composite datatypes. For more information, see APIs That Use Composite Datatypes: page 2 – 13.

Business Rules

- Calling modules can pass either the RESOURCE_LIST_NAME or the RESOURCE_LIST_ID.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- If the calling modules pass both the RESOURCE_LIST_NAME and the RESOURCE_LIST_ID, this API uses the latter.
- You cannot change GROUPED_BY_TYPE if the resource list already contains active members.
- You can change the following fields at any time:
 - RESOURCE LIST NAME
 - DESCRIPTION
 - START DATE
 - END DATE
- You must enter a unique new resource list name.
- You can update existing or add new resource list members by including the member records in the MEMBER_TBL. If a resource list member already exists, you can update the following fields:
 - ALIAS. Specify the P_NEW_ALIAS.
 - SORT_ORDER. Specify the P_SORT_ORDER.

Note: The alias must be unique within a resource group.

The following table shows the parameters for UPDATE_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of the resource list
P_NEW_LIST_NAME	IN	VARCHAR2(30)	No	New name of the existing resource list
P_GROUPED_BY_TYPE	IN	VARCHAR2(30)	No	GROUP_BY_TYPE of the resource list
P_DESCRIPTION	IN	VARCHAR2(80)	No	Description
P_START_DATE	IN	DATE	No	Start date of the resource list
P_END_DATE	IN	DATE	No	End date of the resource list

Name	Usage	Type	Req?	Description
P_MEMBER_TBL	IN	TABLE OF RECORD		
RESOURCE_GROUP_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource group
RESOURCE_GROUP_NAME	IN	VARCHAR2(80)	No	Name of the resource group
RESOURCE_TYPE_CODE	IN	VARCHAR2(30)	Yes	Type code of the resource member
RESOURCE_ATTR_VALUE	IN	VARCHAR2(80)	Yes	Attribute value of the resource member
RESOURCE_ALIAS	IN	VARCHAR2(30)	Yes	Alias of the resource member
SORT_ORDER	IN	NUMBER	No	Sort order of the resource member
RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	Identification code of the resource list member
NEW_ALIAS_MEMBER	IN	VARCHAR2(30)	No	New alias of the resource
ENABLED_FLAG	IN	VARCHAR2(1)	No	Enabled flag of the resource member (default = 'Y')
P_MEMBER_OUT_TBL	OUT	TABLE OF RECORD		
RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		Identification code of the resource list member
RETURN_STATUS	OUT	VARCHAR2(1)		Return status of a specific resource list member

UPDATE_RESOURCE_LIST_MEMBER

UPDATE_RESOURCE_LIST_MEMBER is a PL/SQL procedure that updates the alias and enables or disables the resource list members.

Business Rules

- Calling modules can pass either the P_RESOURCE_LIST_NAME or P_RESOURCE_LIST_ID.
- If the calling modules pass both the P_RESOURCE_LIST_NAME and the P_RESOURCE_LIST_ID, this API uses the latter.
- You can use the P_ENABLED_FLAG to enable or disable a resource member. If the parameter value is passed as NULL or something other than Y, the column value remains the same.

Note: The alias must be unique within a resource group.

The following table shows the parameters for UPDATE_RESOURCE_LIST_MEMBER.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of the resource list
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource member
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	Identification code of the resource list member
P_NEW_ALIAS	IN	VARCHAR2(30)	No	New alias of the resource member
P_SORT_ORDER	IN	NUMBER	No	Sort order of the resource member
P_ENABLED_FLAG	IN	VARCHAR2(1)	No	Enabled flag of the resource member (default = 'Y')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

CLEAR_CREATE_RESOURCE_LIST

CLEAR_CREATE_RESOURCE_LIST is a Load-Execute-Fetch procedure used to clear the global data structures set up during the Initialize step.

CLEAR_UPDATE_MEMBERS

CLEAR_UPDATE_MEMBERS is a Load-Execute-Fetch procedure used to clear the global data structures that were set up during the Initialize step for the Load-Execute-Fetch update APIs.

EXEC_CREATE_RESOURCE_LIST

EXEC_CREATE_RESOURCE_LIST is a Load-Execute-Fetch procedure used to execute the composite API CREATE_RESOURCE_LIST.

The following table shows the parameters for EXEC_CREATE_RESOURCE_LIST.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)	Yes	API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

EXEC_UPDATE_RESOURCE_LIST

EXEC_UPDATE_RESOURCE_LIST is a Load-Execute-Fetch procedure used to execute the composite API UPDATE_RESOURCE_LIST.

The following table shows the parameters for EXEC_UPDATE_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

FETCH_MEMBERS

FETCH_MEMBERS is a Load-Execute-Fetch procedure used to fetch resource members from the global output structure for resource list members.

The following table shows the parameters for FETCH_MEMBERS.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_MEMBER_INDEX	IN	NUMBER		Member Index (default = 1)
P_RESOURCE_LIST_MEMBER_ID	OUT	NUMBER		Identification code of the resource list member
P_MEMBER_RETURN_STATUS	OUT	VARCHAR2(1)		Return status of the specific resource list member

Oracle Proprietary, Confidential Information--Use Restricted by Contract

FETCH_RESOURCE_LIST

FETCH_RESOURCE_LIST is a Load–Execute–Fetch procedure used to fetch one resource list identifier at a time from the global output structure for resource lists.

The following table shows the parameters for FETCH_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_RESOURCE_LIST_ID	OUT	NUMBER		Identification code of the resource list
P_LIST_RETURN_STATUS	OUT	VARCHAR2(1)		Return status of the specific resource list

INIT_CREATE_RESOURCE_LIST

INIT_CREATE_RESOURCE_LIST is a Load–Execute–Fetch procedure used to set up the global data structures used by other Load–Execute–Fetch procedures.

INIT_UPDATE_MEMBERS

INIT_UPDATE_MEMBERS is a Load–Execute–Fetch procedure used to set up the global data structures used by other Load–Execute–Fetch procedures.

LOAD_MEMBERS

LOAD_MEMBERS is a Load–Execute–Fetch procedure used to load the resource list member global input structure.

Business Rules

- Calling modules can pass either P_RESOURCE_LIST_NAME or P_RESOURCE_LIST_ID.
- If the calling modules pass both P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID, the API uses the latter.

Oracle Proprietary, Confidential Information—Use Restricted by Contract

- If the resource list is grouped, you must pass a valid resource group alias.
- The value for P_RESOURCE_ATTR_VALUE must correspond to the value for P_RESOURCE_TYPE. For example, person identification code for P_RESOURCE_ATTR_VALUE must be valid if P_RESOURCE_TYPE equals EMPLOYEE.
- If the calling module passes information to this API for both resource group and resource member parameters, the API first verifies that the resource group exists. If the resource group does not exist, the API creates the resource group and then creates the resource.
- If a given resource member already exists, this API does not return an error. Instead, it returns a successful return status and the resource list member identification code of the existing resource member.

Note: Because you can store only one transaction attribute for a given resource, this API accepts only a single RESOURCE_ATTR_VALUE, which may hold PERSON_ID, JOB_ID, and so on.

- You can use the P_ENABLED_FLAG to enable or disable a resource member. If the parameter value is passed as NULL or something other than Y, the column value remains the same.

Note: The alias must be unique within a resource group.

The following table shows the parameters for LOAD_MEMBERS.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RESOURCE_GROUP_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource group
P_RESOURCE_GROUP_NAME	IN	VARCHAR2(30)	No	Name of the resource group
P_RESOURCE_TYPE_CODE	IN	VARCHAR2(30)	No	Type of the resource
P_RESOURCE_ATTR_VALUE	IN	VARCHAR2(80)	No	Attribute value of the resource
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	Alias of the resource member
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	No	Identification code of the resource list member
P_NEW_ALIAS	IN	VARCHAR2(30)	No	New alias of the resource member
P_SORT_ORDER	IN	NUMBER	No	Sort order of the resource member

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ENABLED_FLAG	OUT	VARCHAR2(1)	No	Enabled flag of the resource member (default = 'Y')
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

LOAD_RESOURCE_LIST

LOAD_RESOURCE_LIST is a Load-Execute-Fetch procedure used to load the resource list global input structure.

Business Rules

- Valid values for P_GROUP_RESOURCE_TYPE are EXPENDITURE_CATEGORY, REVENUE_CATEGORY, ORGANIZATION, and NONE.
- The resource list name must be unique.
- If calling programs pass the P_MEMBER_TBL (optional), this API creates the relevant resource list member records.
- If your resource list is grouped, you must pass a valid resource group alias.
- The value for P_RESOURCE_ATTR_VALUE must correspond with the value for P_RESOURCE_TYPE. For example, P_RESOURCE_ATTR_VALUE must have a valid person identification code if P_RESOURCE_TYPE equals EMPLOYEE.
- If the value for GROUP_RESOURCE_TYPE is NONE, this API ignores resource group IN parameters.
- If you do not specify the resource group alias, the group resource type must be NONE.
- Calling modules can pass either P_RESOURCE_LIST_NAME or P_RESOURCE_LIST_ID.
- If the calling modules pass both P_RESOURCE_LIST_NAME and P_RESOURCE_LIST_ID, this API uses only the latter.
- If the resource list already contains active members, you cannot change GROUPED_BY_TYPE.
- You can change the following fields at any time:
 - RESOURCE_LIST_NAME
 - DESCRIPTION

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- START DATE
- END DATE
- To update existing or add new resource list members, include the member records in MEMBER_TBL. If a resource list member already exists, you can update the following fields:
 - ALIAS. Specify P_NEW_ALIAS.
 - SORT_ORDER. Specify P_SORT_ORDER.
- You can use the value for P_ALIAS as the key to fetch the member record.

Note: The alias must be unique within a resource group.

The following table shows the parameters for LOAD_RESOURCE_LIST.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_RESOURCE_LIST_NAME	IN	VARCHAR2(30)	No	Name of the resource list
P_RESOURCE_LIST_ID	IN	NUMBER	No	Identification code of resource list
P_GROUP_RESOURCE_TYPE	IN	VARCHAR2(30)	No	Type of the resource group
P_DESCRIPTION	IN	VARCHAR2(80)	No	Description
P_START_DATE	IN	DATE	No	Start date of the resource list
P_END_DATE	IN	DATE	No	End date of the resource list
P_NEW_LIST_NAME	IN	VARCHAR2(30)	No	New name of the resource list
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

CHAPTER

8

Detailed Design -- Status APIs

This chapter describes how to implement AMG APIs for progress and earned value information, and provides detailed view and procedure definitions.

Status APIs

Use your external system to calculate and monitor the progress of your project in terms of earned value and percentage complete. Then use the status APIs to report project status inquiry (and billing, if required) to Oracle Projects.

Using the status views described in this section, you can display actual and budgeted amounts in various formats:

- GL period
- PA period
- Work breakdown structure
- Resource
- Burden components

For introductory remarks about this chapter, see Overview of Detailed Design: page 4 – 2.

All the views and procedures discussed in this section are listed below. The procedures are located in the public API package PA_STATUS_PUB.

- Views
 - PA_ACCUM_CMT_TXNS_V: page 8 – 8
 - PA_ACCUM_RSRC_ACT_V: page 8 – 9
 - PA_ACCUM_RSRC_CMT_V: page 8 – 10
 - PA_ACCUM_RSRC_COST_BGT_V: page 8 – 12
 - PA_ACCUM_RSRC_REV_BGT_V: page 8 – 14
 - PA_ACCUM_WBS_ACT_V: page 8 – 15
 - PA_ACCUM_WBS_CMT_V: page 8 – 17
 - PA_ACCUM_WBS_COST_BGT_V: page 8 – 18
 - PA_ACCUM_WBS_REV_BGT_V: page 8 – 20
 - PA_ACT_BY_GL_PERIOD_V: page 8 – 21
 - PA_ACT_BY_PA_PERIOD_V: page 8 – 22
 - PA_BURDEN_COMPONENT_CMT_V: page 8 – 22
 - PA_BURDEN_COMPONENT_COST_V: page 8 – 23
 - PA_CMT_BY_GL_PERIOD_V: page 8 – 24
 - PA_CMT_BY_PA_PERIOD_V: page 8 – 25

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- PA_GL_PERIODS_V: page 8 – 25
- PA_PA_PERIODS_V: page 8 – 25
- PA_PM_REFERENCE_V: page 8 – 26
- PA_TXN_ACCUM_V: page 8 – 27
- Update Progress and Earned Value Procedures
 - UPDATE_EARNED_VALUE: page 8 – 29
 - UPDATE_PROGRESS: page 8 – 30

View Definitions

At the resource level, labor hours (not quantities) are summarized for resources that are tracked as labor. To determine if a resource tracks labor hours but not quantities, join the RESOURCE_LIST_MEMBER_ID to the PA_RESOURCE_LIST_V for the TRACK_AS_LABOR_FLAG column. If TRACK_AS_LABOR_FLAG is Y, the column tracks only labor hours for the resource. Otherwise, quantities are summarized.

For higher-level project and task-level views, the labor hour and quantity summarization rules mentioned above also apply. For example, a project-level labor resource with a TRACK_AS_LABOR_FLAG of Y may show summarized inception-to-date costs, revenues, budgets, and labor hours, but not quantities.

This method of tracking labor hours and quantities has its roots in the way *predefined resources* are summarized in Oracle Projects. All resources in Oracle Projects can be defined as a combination of one or more predefined resources. Predefined resources have three summarization attributes:

- Unit of measure
- Track as labor
- Roll-up actual quantity

The values for the summarization attributes are hard-coded in a view that is used for mapping actuals to resources. The client may change the values by changing the view. The logic of the view is outlined in the two following tables. Table 8 – 1 shows the logic of the view as it relates to predefined resource types.

<i>Predefined Resource Types</i>	<i>Track as Labor</i>	<i>Unit of Measure</i>	<i>Roll-up Actual Quantity</i>
Employee	Yes	Hours	No
Job	Yes	Hours	No
Organization	Yes	Hours	No

Table 8 – 1 Example values returned by the select statement for predefined resource types (Page 1 of 2)

<i>Predefined Resource Types</i>	<i>Track as Labor</i>	<i>Unit of Measure</i>	<i>Roll-up Actual Quantity</i>
Expenditure Type	Depends on the expenditure type attribute with track as labor	Unit of measure specified for the expenditure type	Yes
Event Type	No	blank	No
Supplier	No	blank	No
Expenditure Category	Depends whether the expenditure category includes a labor expenditure type	Depends whether the resource type is tracked as labor	No
Revenue Category	Depends whether the revenue category includes a labor expenditure type	Depends whether the resource type is tracked as labor	No

Table 8 – 1 Example values returned by the select statement for predefined resource types (Page 2 of 2)

Table 8 – 2 shows the logic of the view as it relates to predefined resources.

<i>Predefined Resources</i>	<i>Track as Labor</i>	<i>Unit of Measure</i>	<i>Roll-up Actual Quantity</i>
Uncategorized	Yes	Hours	No
Unclassified	No	blank	No

Table 8 – 2 Example values returned by the select statement for predefined resources (Page 1 of 1)

By defining the uncategorized resource as tracking labor, the client can budget labor hours when entering the uncategorized budget.

To facilitate conditional labor hour and quantity queries on resources, TRACK_AS_LABOR_FLAG is also maintained in the resource member list table. You can query TRACK_AS_LABOR_FLAG column via the PA_RESOURCE_LIST_V view.

To write a select statement on a project-level resource view, you can write a SQL statement similar to the following to conditionally return either labor hours or quantities by resource:

```
SELECT rl.resource_alias List
      , decode(rl.resource_track_as_labor_flag,'Y', ara.actuals_labor_hours_itd,
              'N', ara.actuals_quantity_itd, 0) Units
      , ara.actuals_labor_hours_itd Hours
      , ara.actuals_quantity_itd Qty
FROM   pa_resource_list_v rl
      , pa_accum_rsrc_act_v ara
WHERE  ara.resource_list_member_id = rl.resource_list_member_id
AND    rl.resource_list_id = 1000
AND    ara.project_id = 1043
AND    ara.task_id = 0
```

Although the resource list identification code, project identification code, and retrieved data vary by database, the select statement above should return values similar to those shown in the following table:

<i>List</i>	<i>Units</i>	<i>Hours</i>	<i>Quantity</i>
Labor	406	406	0
Senior.Consultant	40	40	0
Principal.Consultant	122	122	0
Senior.Engineer	164	164	0
Principal.Engineer	80	80	0
Travel	4372	0	4372
Air Travel	3762	0	3762
Personal Auto Use	105	0	105
In-House Recoverables	222	0	222
Computer Services	62	0	62
Automobile Rental	50	0	50
Meals	125	0	125
Other Asset	160	0	160

Table 8 - 3 Example of Values Returned by the Select Statement (Page 1 of 2)

<i>List</i>	<i>Units</i>	<i>Hours</i>	<i>Quantity</i>
Other Expenses	225	0	225
Lodging	330	0	330
Other Expenses	225	0	225

Table 8 – 3 Example of Values Returned by the Select Statement (Page 2 of 2)

PA_ACCUM_CMT_TXNS_V

PA_ACCUM_CMT_TXNS_V is a view that retrieves project-, task-, and resource-related commitments. These commitments include line attributes such as commitment number, dates, expenditure type, and expenditure organization.

This view retrieves three major sets of project-related commitments:

- Project level commitments (TASK_ID and RESOURCE_LIST_MEMBER_ID are zero)
- Project-task level commitments (RESOURCE_LIST_MEMBER_ID is zero)
- Project-task-resource level commitments

Column descriptions for PA_ACCUM_CMT_TXNS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID		NUMBER	The reference code that uniquely identifies the project
TASK_ID		NUMBER	The reference code that uniquely identifies the task within a project
RESOURCE_LIST_MEMBER_ID		NUMBER	The reference code that identifies the resource list member
TXN_ACCUM_ID		NUMBER	The reference code that uniquely identifies the rows in the PA_TXN_ACCUM table
VENDOR_ID		NUMBER	The reference code that identifies the supplier
VENDOR_NAME		VARCHAR2(80)	The name of the supplier
CMT_CREATION_DATE		DATE	Commitment creation date
CMT_TYPE_CODE		VARCHAR2(1)	The reference code that identifies the commitment type
CMT_TYPE_CODE_M		VARCHAR2(80)	Commitment type meaning
CMT_NUMBER		VARCHAR2(50)	The number of the commitment
CMT_LINE_NUMBER		NUMBER	The number of the commitment line
TOT_CMT_RAW_COST		NUMBER	Total commitment raw cost
TOT_CMT_BURDENED_COST		NUMBER	Total commitment burdened cost
CMT_APPROVED_DATE		DATE	The date on which the commitment was approved
CMT_PROMISED_DATE		DATE	The date on which delivery of the commitment is promised
CMT_NEED_BY_DATE		DATE	The date on which the commitment is needed
DESCRIPTION		VARCHAR2(255)	The description of the commitment
TASK_NUMBER		VARCHAR2(25)	The number of the task
TASK_NAME		VARCHAR2(20)	The name of the task
EXPENDITURE_TYPE		VARCHAR2(30)	Expenditure type
ORGANIZATION_ID		NUMBER	The reference code that identifies the organization

Oracle Proprietary, Confidential Information--Use Restricted by Contract

EXPENDITURE_ORG_NAME	VARCHAR2(60)	The name of the expenditure organization
CMT_APPROVED_FLAG	VARCHAR2(1)	Commitment approved flag
QUANTITY_OUTSTANDING	NUMBER	Quantity outstanding
UNIT_OF_MEASURE	VARCHAR2(30)	Unit of measure
CMT_REQUESTOR_NAME	VARCHAR2(240)	The name of the person requesting the commitment
CMT_BUYER_NAME	VARCHAR2(240)	The name of the person buying the commitment
AMOUNT_DELIVERED	NUMBER	Amount delivered
TRANSACTION_SOURCE	VARCHAR2(30)	Transaction source
TRANSACTION_SOURCE_M	VARCHAR2(80)	Transaction source meaning

PA_ACCUM_RSRC_ACT_V

PA_ACCUM_RSRC_ACT_V is a view that returns current project- and task-level resource actual cost and revenue summary amounts by the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date

For more information, see: PA_ACCUM_RSRC_ACT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_RSRC_ACT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID		NUMBER	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID		NUMBER	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID		NUMBER	The identification code of the resource list member
PARENT_MEMBER_ID		NUMBER	The identification code of the parent member
RESOURCE_LIST_ASSIGNMENT_ID		NUMBER	The identification code of the resource list assignment
RAW_COST_ITD		NUMBER	Inception-to-date raw cost
RAW_COST_YTD		NUMBER	Year-to-date raw cost
RAW_COST_PP		NUMBER	Prior period raw cost
RAW_COST_PTD		NUMBER	Period-to-date raw cost
BILLABLE_RAW_COST_ITD		NUMBER	Inception-to-date billable raw cost
BILLABLE_RAW_COST_YTD		NUMBER	Year-to-date billable raw cost

Oracle Proprietary, Confidential Information--Use Restricted by Contract

BILLABLE_RAW_COST_PP	NUMBER	Prior period billable raw cost
BILLABLE_RAW_COST_PTD	NUMBER	Period-to-date billable raw cost
BURDENED_COST_ITD	NUMBER	Inception-to-date burdened cost
BURDENED_COST_YTD	NUMBER	Year-to-date burdened cost
BURDENED_COST_PP	NUMBER	Prior period burdened cost
BURDENED_COST_PTD	NUMBER	Period-to-date burdened cost
BILLABLE_BURDENED_COST_ITD	NUMBER	Inception-to-date billable burdened cost
BILLABLE_BURDENED_COST_YTD	NUMBER	Year-to-date billable burdened cost
BILLABLE_BURDENED_COST_PP	NUMBER	Prior period billable burdened cost
BILLABLE_BURDENED_COST_PTD	NUMBER	Period-to-date billable burdened cost
ACTUALS_QUANTITY_ITD	NUMBER	Inception-to-date actuals quantity
ACTUALS_QUANTITY_YTD	NUMBER	Year-to-date actuals quantity
ACTUALS_QUANTITY_PP	NUMBER	Prior period actuals quantity
ACTUALS_QUANTITY_PTD	NUMBER	Period-to-date actuals quantity
BILLABLE_QUANTITY_ITD	NUMBER	Inception-to-date billable quantity
BILLABLE_QUANTITY_YTD	NUMBER	Year-to-date billable quantity
BILLABLE_QUANTITY_PP	NUMBER	Prior period billable quantity
BILLABLE_QUANTITY_PTD	NUMBER	Period-to-date billable quantity
ACTUALS_LABOR_HOURS_ITD	NUMBER	Inception-to-date actuals quantity
ACTUALS_LABOR_HOURS_YTD	NUMBER	Year-to-date actuals quantity
ACTUALS_LABOR_HOURS_PP	NUMBER	Prior period actuals quantity
ACTUALS_LABOR_HOURS_PTD	NUMBER	Period-to-date actuals quantity
BILLABLE_LABOR_HOURS_ITD	NUMBER	Inception-to-date billable labor hours
BILLABLE_LABOR_HOURS_YTD	NUMBER	Year-to-date billable labor hours
BILLABLE_LABOR_HOURS_PP	NUMBER	Prior period billable labor hours
BILLABLE_LABOR_HOURS_PTD	NUMBER	Period-to-date billable labor hours
REVENUE_ITD	NUMBER	Inception-to-date revenue
REVENUE_YTD	NUMBER	Year-to-date revenue
REVENUE_PP	NUMBER	Prior period revenue
REVENUE_PTD	NUMBER	Period-to-date revenue

PA_ACCUM_RSRC_CMT_V

PA_ACCUM_RSRC_CMT_V is a view that returns current project- and task-level resource commitment summary amounts by the following periods:

- Inception-to-date
- Year-to-date
- Prior period

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- Period-to-date

For more information, see PA_ACCUM_RSRC_CMT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_RSRC_CMT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID		NUMBER	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID		NUMBER	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID		NUMBER	The identification code of the resource list member
PARENT_MEMBER_ID		NUMBER	The identification code of the parent member
RESOURCE_LIST_ASSIGNMENT_ID		NUMBER	The identification code of the resource list assignment
CMT_RAW_COST_ITD		NUMBER	Inception-to-date commitment raw cost
CMT_RAW_COST_YTD		NUMBER	Year-to-date commitment raw cost
CMT_RAW_COST_PP		NUMBER	Prior period commitment raw cost
CMT_RAW_COST_PTD		NUMBER	Period-to-date commitment raw cost
CMT_BURDENED_COST_ITD		NUMBER	Inception-to-date commitment burdened cost
CMT_BURDENED_COST_YTD		NUMBER	Year-to-date commitment burdened cost
CMT_BURDENED_COST_PP		NUMBER	Prior period commitment burdened cost
CMT_BURDENED_COST_PTD		NUMBER	Period-to-date commitment burdened cost
CMT_QUANTITY_ITD		NUMBER	Inception-to-date commitment quantity
CMT_QUANTITY_YTD		NUMBER	Year-to-date commitment quantity
CMT_QUANTITY_PP		NUMBER	Prior period commitment quantity
CMT_QUANTITY_PTD		NUMBER	Period-to-date commitment quantity

PA_ACCUM_RSRC_COST_BGT_V

PA_ACCUM_RSRC_COST_BGT_V is a view that returns project- and task-level resource cost budget summary amounts for the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date
- Total

For more information, see PA_ACCUM_RSRC_COST_BGT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_RSRC_COST_BGT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID		NUMBER	The identification code of the resource list member
PARENT_MEMBER_ID		NUMBER	The identification code of the parent member
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
RESOURCE_LIST_ASSIGNMENT_ID		NUMBER	The identification code of the resource list assignment
BASELINE_RAW_COST_ITD		NUMBER	Inception-to-date baseline raw cost
BASELINE_RAW_COST_YTD		NUMBER	Year-to-date baseline raw cost
BASELINE_RAW_COST_PP		NUMBER	Prior period baseline raw cost
BASELINE_RAW_COST_PTD		NUMBER	Period-to-date baseline raw cost
BASELINE_BURDENED_COST_ITD		NUMBER	Inception-to-date baseline burdened cost
BASELINE_BURDENED_COST_YTD		NUMBER	Year-to-date baseline burdened cost
BASELINE_BURDENED_COST_PP		NUMBER	Prior period baseline burdened cost
BASELINE_BURDENED_COST_PTD		NUMBER	Period-to-date baseline burdened cost
ORIGINAL_RAW_COST_ITD		NUMBER	Inception-to-date original raw cost
ORIGINAL_RAW_COST_YTD		NUMBER	Year-to-date original raw cost
ORIGINAL_RAW_COST_PP		NUMBER	Prior period original raw cost

Oracle Proprietary, Confidential Information--Use Restricted by Contract

ORIGINAL_RAW_COST_PTD	NUMBER	Period-to-date original raw cost
ORIGINAL_BURDENED_COST_ITD	NUMBER	Inception-to-date original burdened cost
ORIGINAL_BURDENED_COST_YTD	NUMBER	Year-to-date original burdened cost
ORIGINAL_BURDENED_COST_PP	NUMBER	Prior period original burdened cost
ORIGINAL_BURDENED_COST_PTD	NUMBER	Period-to-date original burdened cost
BASELINE_QUANTITY_ITD	NUMBER	Inception-to-date baseline quantity
BASELINE_QUANTITY_YTD	NUMBER	Year-to-date baseline quantity
BASELINE_QUANTITY_PP	NUMBER	Prior period baseline quantity
BASELINE_QUANTITY_PTD	NUMBER	Period-to-date baseline quantity
ORIGINAL_QUANTITY_ITD	NUMBER	Inception-to-date original quantity
ORIGINAL_QUANTITY_YTD	NUMBER	Year-to-date original quantity
ORIGINAL_QUANTITY_PP	NUMBER	Prior period original quantity
ORIGINAL_QUANTITY_PTD	NUMBER	Period-to-date original quantity
ORIGINAL_LABOR_HOURS_ITD	NUMBER	Inception-to-date original labor hours
ORIGINAL_LABOR_HOURS_YTD	NUMBER	Year-to-date original labor hours
ORIGINAL_LABOR_HOURS_PP	NUMBER	Prior period original labor hours
ORIGINAL_LABOR_HOURS_PTD	NUMBER	Period-to-date original labor hours
BASELINE_LABOR_HOURS_ITD	NUMBER	Inception-to-date baseline labor hours
BASELINE_LABOR_HOURS_YTD	NUMBER	Year-to-date baseline labor hours
BASELINE_LABOR_HOURS_PP	NUMBER	Prior period baseline labor hours
BASELINE_LABOR_HOURS_PTD	NUMBER	Period-to-date baseline labor hours
BASELINE_RAW_COST_TOT	NUMBER	Total baseline raw cost
BASELINE_BURDENED_COST_TOT	NUMBER	Total baseline burdened cost
ORIGINAL_RAW_COST_TOT	NUMBER	Total original raw cost
ORIGINAL_BURDENED_COST_TOT	NUMBER	Total original burdened cost
ORIGINAL_LABOR_HOURS_TOT	NUMBER	Total original labor hours
BASELINE_LABOR_HOURS_TOT	NUMBER	Total baseline labor hours
BASELINE_QUANTITY_TOT	NUMBER	Total baseline quantity
ORIGINAL_QUANTITY_TOT	NUMBER	Total original quantity

PA_ACCUM_RSRC_REV_BGT_V

PA_ACCUM_RSRC_REV_BGT_V is a view that returns project- and task-level resource revenue budget summary amounts for the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date
- Total

For more information, see PA_ACCUM_RSRC_REV_BGT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_RSRC_REV_BGT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID		NUMBER	The identification code of the resource list member
PARENT_MEMBER_ID		NUMBER	The identification code of the parent member
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The reference code that identifies the budget type
RESOURCE_LIST_ASSIGNMENT_ID		NUMBER	The identification code of the resource list assignment
BASELINE_REVENUE_ITD		NUMBER	Inception-to-date baseline revenue
BASELINE_REVENUE_YTD		NUMBER	Year-to-date baseline revenue
BASELINE_REVENUE_PP		NUMBER	Prior period baseline revenue
BASELINE_REVENUE_PTD		NUMBER	Period-to-date baseline revenue
ORIGINAL_REVENUE_ITD		NUMBER	Inception-to-date original revenue
ORIGINAL_REVENUE_YTD		NUMBER	Year-to-date original revenue
ORIGINAL_REVENUE_PP		NUMBER	Prior period original revenue
ORIGINAL_REVENUE_PTD		NUMBER	Period-to-date original revenue
BASELINE_QUANTITY_ITD		NUMBER	Inception-to-date baseline quantity
BASELINE_QUANTITY_YTD		NUMBER	Year-to-date baseline quantity
BASELINE_QUANTITY_PP		NUMBER	Prior period baseline quantity
BASELINE_QUANTITY_PTD		NUMBER	Period-to-date baseline quantity
ORIGINAL_QUANTITY_ITD		NUMBER	Inception-to-date original quantity

Oracle Proprietary, Confidential Information--Use Restricted by Contract

ORIGINAL_QUANTITY_YTD	NUMBER	Year-to-date original quantity
ORIGINAL_QUANTITY_PP	NUMBER	Prior period original quantity
ORIGINAL_QUANTITY_PTD	NUMBER	Period-to-date original quantity
ORIGINAL_LABOR_HOURS_ITD	NUMBER	Inception-to-date original labor hours
ORIGINAL_LABOR_HOURS_YTD	NUMBER	Year-to-date original labor hours
ORIGINAL_LABOR_HOURS_PP	NUMBER	Prior period original labor hours
ORIGINAL_LABOR_HOURS_PTD	NUMBER	Period-to-date original labor hours
BASELINE_LABOR_HOURS_ITD	NUMBER	Inception-to-date baseline labor hours
BASELINE_LABOR_HOURS_YTD	NUMBER	Year-to-date baseline labor hours
BASELINE_LABOR_HOURS_PP	NUMBER	Prior period baseline labor hours
BASELINE_LABOR_HOURS_PTD	NUMBER	Period-to-date baseline labor hours
BASELINE_REVENUE_TOT	NUMBER	Total baseline revenue
ORIGINAL_REVENUE_TOT	NUMBER	Total original revenue
ORIGINAL_LABOR_HOURS_TOT	NUMBER	Total original labor hours
BASELINE_LABOR_HOURS_TOT	NUMBER	Total baseline labor hours
BASELINE_QUANTITY_TOT	NUMBER	Total baseline quantity
ORIGINAL_QUANTITY_TOT	NUMBER	Total original quantity

PA_ACCUM_WBS_ACT_V

PA_ACCUM_WBS_ACT_V is a view that returns current project- and task-level actual cost and revenue summary amounts by the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date

For more information, see PA_ACCUM_WBS_ACT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_WBS_ACT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	The number of the project in Oracle Projects

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PROJECT_NAME	NOT NULL	VARCHAR2(30)	The name of the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
TASK_NUMBER		VARCHAR2(25)	The number of the task in Oracle Projects
TASK_NAME		VARCHAR2(20)	The name of the task in Oracle Projects
INDENTED_TASK_NUMBER		VARCHAR2(2000)	Indented task number
INDENTED_TASK_NAME		VARCHAR2(2000)	Indented task name
WBS_LEVEL		NUMBER(3)	The level of the task in the project's work breakdown structure
TASK_START_DATE		DATE	Task start date
TASK_COMPLETION_DATE		DATE	Task completion date
TASK_MANAGER_PERSON_ID		NUMBER(9)	The person identification code of the task manager
TASK_MANAGER_NAME		VARCHAR2(240)	The name of the task manager
RAW_COST_ITD		NUMBER	Inception-to-date raw cost
RAW_COST_YTD		NUMBER	Year-to-date raw cost
RAW_COST_PP		NUMBER	Prior period raw cost
RAW_COST_PTD		NUMBER	Period-to-date raw cost
BILLABLE_RAW_COST_ITD		NUMBER	Inception-to-date billable raw cost
BILLABLE_RAW_COST_YTD		NUMBER	Year-to-date billable raw cost
BILLABLE_RAW_COST_PP		NUMBER	Prior period billable raw cost
BILLABLE_RAW_COST_PTD		NUMBER	Period-to-date billable raw cost
BURDENED_COST_ITD		NUMBER	Inception-to-date burdened cost
BURDENED_COST_YTD		NUMBER	Year-to-date burdened cost
BURDENED_COST_PP		NUMBER	Prior period burdened cost
BURDENED_COST_PTD		NUMBER	Period-to-date burdened cost
BILLABLE_BURDENED_COST_ITD		NUMBER	Inception-to-date billable burdened cost
BILLABLE_BURDENED_COST_YTD		NUMBER	Year-to-date billable burdened cost
BILLABLE_BURDENED_COST_PP		NUMBER	Prior period billable burdened cost
BILLABLE_BURDENED_COST_PTD		NUMBER	Period-to-date billable burdened cost
ACTUALS_LABOR_HOURS_ITD		NUMBER	Inception-to-date actuals labor hours
ACTUALS_LABOR_HOURS_YTD		NUMBER	Year-to-date actuals labor hours
ACTUALS_LABOR_HOURS_PP		NUMBER	Prior period actuals labor hours
ACTUALS_LABOR_HOURS_PTD		NUMBER	Period-to-date actuals labor hours
BILLABLE_LABOR_HOURS_ITD		NUMBER	Inception-to-date billable labor hours
BILLABLE_LABOR_HOURS_YTD		NUMBER	Year-to-date billable labor hours
BILLABLE_LABOR_HOURS_PP		NUMBER	Prior period billable labor hours
BILLABLE_LABOR_HOURS_PTD		NUMBER	Period-to-date billable labor hours
REVENUE_ITD		NUMBER	Inception-to-date revenue
REVENUE_YTD		NUMBER	Year-to-date revenue
REVENUE_PP		NUMBER	Prior period revenue
REVENUE_PTD		NUMBER	Period-to-date revenue

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_ACCUM_WBS_CMT_V

PA_ACCUM_WBS_CMT_V is a view that returns current project- and task-level commitment summary amounts by the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date

For more information, see PA_ACCUM_WBS_CMT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_WBS_CMT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	The number of the project in Oracle Projects
PROJECT_NAME	NOT NULL	VARCHAR2(30)	The name of the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
TASK_NUMBER		VARCHAR2(25)	The number of the task
TASK_NAME		VARCHAR2(20)	The name of the task
INDENTED_TASK_NUMBER		VARCHAR2(2000)	Indented task number
INDENTED_TASK_NAME		VARCHAR2(2000)	Indented task name
WBS_LEVEL		NUMBER(3)	The level of the task in the project's work breakdown structure
TASK_START_DATE		DATE	Task start date
TASK_COMPLETION_DATE		DATE	Task completion date
TASK_MANAGER_PERSON_ID		NUMBER(9)	The person identification code of the task manager
TASK_MANAGER_NAME		VARCHAR2(240)	The name of the task manager
CMT_RAW_COST_ITD		NUMBER	Inception-to-date commitment raw cost
CMT_RAW_COST_YTD		NUMBER	Year-to-date commitment raw cost
CMT_RAW_COST_PP		NUMBER	Prior period commitment raw cost
CMT_RAW_COST_PTD		NUMBER	Period-to-date commitment raw cost
CMT_BURDENED_COST_ITD		NUMBER	Inception-to-date commitment burdened cost
CMT_BURDENED_COST_YTD		NUMBER	Year-to-date commitment burdened cost
CMT_BURDENED_COST_PP		NUMBER	Prior period commitment burdened cost

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CMT_BURDENED_COST_PTD	NUMBER	Period-to-date commitment burdened cost
CMT_QUANTITY_ITD	NUMBER	Inception-to-date commitment quantity
CMT_QUANTITY_YTD	NUMBER	Year-to-date commitment quantity
CMT_QUANTITY_PP	NUMBER	Prior period commitment quantity
CMT_QUANTITY_PTD	NUMBER	Period-to-date commitment quantity

PA_ACCUM_WBS_COST_BGT_V

PA_ACCUM_WBS_COST_BGT_V is a view that returns project- and task-level cost budget summary amounts for the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date
- Total

For more information, see PA_ACCUM_WBS_COST_BGT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_WBS_COST_BGT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	The number of the project in Oracle Projects
PROJECT_NAME	NOT NULL	VARCHAR2(30)	The name of the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
TASK_NUMBER		VARCHAR2(25)	The number of the task in Oracle Projects
TASK_NAME		VARCHAR2(20)	The name of the task in Oracle Projects
INDENTED_TASK_NUMBER		VARCHAR2(2000)	Indented task number
INDENTED_TASK_NAME		VARCHAR2(2000)	Indented task name
WBS_LEVEL		NUMBER(3)	The level of the task in the project's work breakdown structure
TASK_START_DATE		DATE	Task start date
TASK_COMPLETION_DATE		DATE	Task completion date

Oracle Proprietary, Confidential Information--Use Restricted by Contract

TASK_MANAGER_PERSON_ID	NUMBER(9)	The person identification code of the task manager
TASK_MANAGER_NAME	VARCHAR2(240)	The name of the task manager
BUDGET_TYPE_CODE	NOT NULL VARCHAR2(30)	The reference code that identifies the budget type
BASELINE_RAW_COST_ITD	NUMBER	Inception-to-date baseline raw cost
BASELINE_RAW_COST_YTD	NUMBER	Year-to-date baseline raw cost
BASELINE_RAW_COST_PP	NUMBER	Prior period baseline raw cost
BASELINE_RAW_COST_PTD	NUMBER	Period-to-date baseline raw cost
BASELINE_BURDENED_COST_ITD	NUMBER	Inception-to-date baseline burdened cost
BASELINE_BURDENED_COST_YTD	NUMBER	Year-to-date baseline burdened cost
BASELINE_BURDENED_COST_PP	NUMBER	Prior period baseline burdened cost
BASELINE_BURDENED_COST_PTD	NUMBER	Period-to-date baseline burdened cost
ORIGINAL_RAW_COST_ITD	NUMBER	Inception-to-date original raw cost
ORIGINAL_RAW_COST_YTD	NUMBER	Year-to-date original raw cost
ORIGINAL_RAW_COST_PP	NUMBER	Prior period original raw cost
ORIGINAL_RAW_COST_PTD	NUMBER	Period-to-date original raw cost
ORIGINAL_BURDENED_COST_ITD	NUMBER	Inception-to-date original burdened cost
ORIGINAL_BURDENED_COST_YTD	NUMBER	Year-to-date original burdened cost
ORIGINAL_BURDENED_COST_PP	NUMBER	Prior period original burdened cost
ORIGINAL_BURDENED_COST_PTD	NUMBER	Period-to-date original burdened cost
ORIGINAL_LABOR_HOURS_ITD	NUMBER	Inception-to-date original labor hours
ORIGINAL_LABOR_HOURS_YTD	NUMBER	Year-to-date original labor hours
ORIGINAL_LABOR_HOURS_PP	NUMBER	Prior period original labor hours
ORIGINAL_LABOR_HOURS_PTD	NUMBER	Period-to-date original labor hours
BASELINE_LABOR_HOURS_ITD	NUMBER	Inception-to-date baseline labor hours
BASELINE_LABOR_HOURS_YTD	NUMBER	Year-to-date baseline labor hours
BASELINE_LABOR_HOURS_PP	NUMBER	Prior period baseline labor hours
BASELINE_LABOR_HOURS_PTD	NUMBER	Period-to-date baseline labor hours
BASELINE_RAW_COST_TOT	NUMBER	Total baseline raw cost
BASELINE_BURDENED_COST_TOT	NUMBER	Total baseline burdened cost
ORIGINAL_RAW_COST_TOT	NUMBER	Total original raw cost
ORIGINAL_BURDENED_COST_TOT	NUMBER	Total original burdened cost
ORIGINAL_LABOR_HOURS_TOT	NUMBER	Total original labor hours
BASELINE_LABOR_HOURS_TOT	NUMBER	Total baseline labor hours

PA_ACCUM_WBS_REV_BGT_V

PA_ACCUM_WBS_REV_BGT_V is a view that returns project- and task-level revenue budget summary amounts for the following periods:

- Inception-to-date
- Year-to-date
- Prior period
- Period-to-date
- Total

For more information, see PA_ACCUM_WBS_REV_BGT_V in the *Oracle Projects Technical Reference Manual*.

Column descriptions for PA_ACCUM_WBS_REV_BGT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	The number of the project in Oracle Projects
PROJECT_NAME	NOT NULL	VARCHAR2(30)	The name of the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_NUMBER		VARCHAR2(25)	The number of the task in Oracle Projects
TASK_NAME		VARCHAR2(20)	The number of the task in Oracle Projects
INDENTED_TASK_NUMBER		VARCHAR2(2000)	Indented task number
INDENTED_TASK_NAME		VARCHAR2(2000)	Indented task name
WBS_LEVEL		NUMBER(3)	The level of the task in the project's work breakdown structure
TASK_START_DATE		DATE	Task start date
TASK_COMPLETION_DATE		DATE	Task completion date
TASK_MANAGER_PERSON_ID		NUMBER(9)	The person identification code of the task manager
TASK_MANAGER_NAME		VARCHAR2(240)	The name of the task manager
BUDGET_TYPE_CODE	NOT NULL	VARCHAR2(30)	The identification code of the budget type
BASELINE_REVENUE_ITD		NUMBER	Inception-to-date baseline revenue
BASELINE_REVENUE_YTD		NUMBER	Year-to-date baseline revenue
BASELINE_REVENUE_PP		NUMBER	Prior period baseline revenue
BASELINE_REVENUE_PTD		NUMBER	Period-to-date baseline revenue
ORIGINAL_REVENUE_ITD		NUMBER	Inception-to-date original revenue
ORIGINAL_REVENUE_YTD		NUMBER	Year-to-date original revenue
ORIGINAL_REVENUE_PP		NUMBER	Prior period original revenue

Oracle Proprietary, Confidential Information--Use Restricted by Contract

ORIGINAL_REVENUE_PTD	NUMBER	Period-to-date original revenue
ORIGINAL_LABOR_HOURS_ITD	NUMBER	Inception-to-date original labor hours
ORIGINAL_LABOR_HOURS_YTD	NUMBER	Year-to-date original labor hours
ORIGINAL_LABOR_HOURS_PP	NUMBER	Prior period original labor hours
ORIGINAL_LABOR_HOURS_PTD	NUMBER	Period-to-date original labor hours
BASELINE_LABOR_HOURS_ITD	NUMBER	Inception-to-date baseline labor hours
BASELINE_LABOR_HOURS_YTD	NUMBER	Year-to-date baseline labor hours
BASELINE_LABOR_HOURS_PP	NUMBER	Prior period baseline labor hours
BASELINE_LABOR_HOURS_PTD	NUMBER	Period-to-date baseline labor hours
BASELINE_REVENUE_TOT	NUMBER	Total baseline revenue
ORIGINAL_REVENUE_TOT	NUMBER	Total original revenue
ORIGINAL_LABOR_HOURS_TOT	NUMBER	Total original labor hours
BASELINE_LABOR_HOURS_TOT	NUMBER	Total baseline labor hours

PA_ACT_BY_GL_PERIOD_V

PA_ACT_BY_GL_PERIOD_V is a view that returns actual cost and revenue totals for lowest tasks and resources by GL periods.

Column descriptions for PA_ACT_BY_GL_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	GL period name
PERIOD_START_DATE	NOT NULL	DATE	Period start date
PERIOD_END_DATE	NOT NULL	DATE	Period end date
REVENUE		NUMBER	Revenue
RAW_COST		NUMBER	Raw cost
CAPITAL_RAW_COST		NUMBER	Capital raw cost
BURDENED_COST		NUMBER	Burdened cost
CAPITAL_BURDENED_COST		NUMBER	Capital burdened cost
QUANTITY		NUMBER	Quantity
LABOR_HOURS		NUMBER	Labor hours
BILLABLE_RAW_COST		NUMBER	Billable raw cost
BILLABLE_BURDENED_COST		NUMBER	Billable burdened cost
BILLABLE_QUANTITY		NUMBER	Billable quantity

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_ACT_BY_PA_PERIOD_V

PA_ACT_BY_PA_PERIOD_V is a view that returns actual cost and revenue totals for lowest tasks and resources by PA periods.

Column descriptions for PA_ACT_BY_PA_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
PERIOD_START_DATE	NOT NULL	DATE	Period start date
PERIOD_END_DATE	NOT NULL	DATE	Period end date
REVENUE		NUMBER	Revenue
RAW_COST		NUMBER	Raw cost
CAPITAL_RAW_COST		NUMBER	Capital raw cost
BURDENED_COST		NUMBER	Burdened cost
CAPITAL_BURDENED_COST		NUMBER	Capital burdened cost
QUANTITY		NUMBER	Quantity
LABOR_HOURS		NUMBER	Labor hours
BILLABLE_RAW_COST		NUMBER	Billable raw cost
BILLABLE_BURDENED_COST		NUMBER	Billable burdened cost
BILLABLE_QUANTITY		NUMBER	Billable quantity
BILLABLE_LABOR_HOURS		NUMBER	Billable labor hours

PA_BURDEN_COMPONENT_CMT_V

PA_BURDEN_COMPONENT_CMT_V is a view that returns commitment burden components by resource, PA period name, expenditure type, expenditure organization, and burden set for each transaction summarization record.

Column descriptions for PA_BURDEN_COMPONENT_CMT_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
RESOURCE_LIST_NAME		VARCHAR2(60)	The name of the resource list
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
ALIAS		VARCHAR2(30)	Alias
EXPENDITURE_ORGANIZATION_ID		NUMBER	The identification code of the expenditure organization
EXPENDITURE_ORGANIZATION	NOT NULL	VARCHAR2(60)	The name of the expenditure organization
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
CMT_COMPILED_SET_ID		NUMBER(15)	Cost burden set identifier
COST_BASE	NOT NULL	VARCHAR2(30)	Name of the cost base
PRECEDENCE	NOT NULL	NUMBER(15)	Precedence of the burden cost code
BURDEN_COST_CODE	NOT NULL	VARCHAR2(30)	Name of the burden cost code within the cost base
INPUT_MULTIPLIER	NOT NULL	NUMBER(22,5)	Original multiplier defined in the burden schedule revision
COMPILED_MULTIPLIER	NOT NULL	NUMBER(22,5)	Build-up multiplier
CMT_RAW_COST		NUMBER	PA_TXN_ACCUM transaction raw cost
CMT_BURDEN_COST		NUMBER	PA_TXN_ACCUM total raw cost multiplied by COMPILED_MULTIPLIER

PA_BURDEN_COMPONENT_COST_V

PA_BURDEN_COMPONENT_COST_V is a view that returns actual burden components by resource, PA period name, expenditure type, expenditure organization, and burden set for each transaction summarization record. This view returns burden cost components only for resources that have been burdened.

Column descriptions for PA_BURDEN_COMPONENT_COST_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects

Oracle Proprietary, Confidential Information--Use Restricted by Contract

RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
RESOURCE_LIST_NAME	NOT NULL	VARCHAR2(60)	The name of the resource list
RESOURCE_LIST_MEMBER_ID		NUMBER(15)	The identification code of the resource list member
ALIAS		VARCHAR2(30)	Alias
EXPENDITURE_ORGANIZATION_ID		NUMBER	The identification code of the expenditure organization
EXPENDITURE_ORGANIZATION	NOT NULL	VARCHAR2(60)	The name of the expenditure organization
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
COST_COMPILED_SET_ID		NUMBER(15)	Cost burden set identifier
COST_BASE	NOT NULL	VARCHAR2(30)	Name of the cost base
PRECEDENCE	NOT NULL	NUMBER(15)	Precedence of the burden cost code
BURDEN_COST_CODE	NOT NULL	VARCHAR2(30)	Name of the burden cost code within the cost base
INPUT_MULTIPLIER	NOT NULL	NUMBER(22,5)	Original multiplier defined in the burden schedule revision
COMPILED_MULTIPLIER	NOT NULL	NUMBER(22,5)	Build-up multiplier
RAW_COST		NUMBER	PA_TXN_ACCUM transaction raw cost
BURDEN_COST		NUMBER	PA_TXN_ACCUM total raw cost multiplied by COMPILED_MULTIPLIER

PA_CMT_BY_GL_PERIOD_V

PA_CMT_BY_GL_PERIOD_V is a view that returns current commitment totals for lowest tasks and resources by GL periods.

Column descriptions for PA_CMT_BY_GL_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	GL period name
PERIOD_START_DATE	NOT NULL	DATE	Period start date
PERIOD_END_DATE	NOT NULL	DATE	Period end date
CMT_QUANTITY		NUMBER	Quantity
CMT_RAW_COST		NUMBER	Raw cost
CMT_BURDENED_COST		NUMBER	Burdened cost

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_CMT_BY_PA_PERIOD_V

PA_CMT_BY_PA_PERIOD_V is a view that returns current commitment totals for lowest tasks and resources by PA periods.

Column descriptions for PA_CMT_BY_PA_PERIOD_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
PERIOD_START_DATE	NOT NULL	DATE	Period start date
PERIOD_END_DATE	NOT NULL	DATE	Period end date
CMT_QUANTITY		NUMBER	Quantity
CMT_RAW_COST		NUMBER	Raw cost
CMT_BURDENED_COST		NUMBER	Burdened cost

PA_GL_PERIODS_V

PA_GL_PERIODS_V is a view of the PA_PERIODS tables for GL periods and their start and end dates.

Column descriptions for PA_GL_PERIODS_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	GL period name
START_DATE	NOT NULL	DATE	Start date
END_DATE	NOT NULL	DATE	End date

PA_PA_PERIODS_V

PA_PA_PERIODS_V is a view of the PA_PERIODS tables for PA periods and their start and end dates.

Column descriptions for PA_PA_PERIODS_V are shown in the following table.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Column Descriptions

Name	Null?	Type	Description
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
CURRENT_PA_PERIOD_FLAG		VARCHAR2(1)	Current PA period flag
START_DATE	NOT NULL	DATE	Start date
END_DATE	NOT NULL	DATE	End date
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	GL period name

PA_PM_REFERENCE_V

PA_PM_REFERENCE_V is a view that retrieves Oracle Projects identifiers and reference codes from your external systems for projects and tasks.

Column descriptions for PA_PM_REFERENCE_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
PM_PROJECT_REFERENCE		VARCHAR2(25)	The reference code that uniquely identifies the project in the external system
PM_PRODUCT_CODE		VARCHAR2(10)	The code identifying the external system
DESCRIPTION		VARCHAR2(250)	Description
PROJECT_START_DATE		DATE	Project start date
PROJECT_COMPLETION_DATE		DATE	Project completion date
PROJECT_STATUS_CODE	NOT NULL	VARCHAR2(30)	The identification code of the project status
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
PARENT_TASK_ID		NUMBER(15)	The reference code that identifies the task's parent task within a project in Oracle Projects
TOP_TASK_ID	NOT NULL	NUMBER(15)	The reference code that identifies the top task above this task within a project in Oracle Projects
TASK_NUMBER	NOT NULL	VARCHAR2(25)	The number of the task
PM_TASK_REFERENCE		VARCHAR2(25)	The reference code that uniquely identifies the task in the external system
TASK_START_DATE		DATE	Task start date
TASK_COMPLETION_DATE		DATE	Task completion date
SERVICE_TYPE_CODE		VARCHAR2(30)	The identification code of the service type
CHARGEABLE_FLAG		VARCHAR2(1)	Chargeable flag
BILLABLE_FLAG		VARCHAR2(1)	Billable flag

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_TXN_ACCUM_V

PA_TXN_ACCUM_V is a view that shows detail information by various transaction attributes. Transaction attributes can include person, job, organization, vendor, expenditure type, event type, non-labor resource, expenditure category, revenue category, non-labor resource organization, event type classification, system linkage function, and week ending date.

Column descriptions for PA_TXN_ACCUM_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
PROJECT_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the project in Oracle Projects
TASK_ID	NOT NULL	NUMBER(15)	The reference code that uniquely identifies the task within a project in Oracle Projects
RESOURCE_LIST_MEMBER_ID	NOT NULL	NUMBER(15)	The identification code of the resource list member
RESOURCE_LIST_ID	NOT NULL	NUMBER(15)	The identification code of the resource list
PA_PERIOD_NAME	NOT NULL	VARCHAR2(20)	PA period name
GL_PERIOD_NAME	NOT NULL	VARCHAR2(15)	GL period name
PERSON_ID		NUMBER	The identification code of the person
PERSON_FULL_NAME		VARCHAR2(240)	The full name of the person
JOB_ID		NUMBER	The identification code of the job
JOB		VARCHAR2(240)	The name of the job
EXPENDITURE_ORGANIZATION_ID		NUMBER	The identification code of the expenditure organization
EXPENDITURE_ORGANIZATION	NOT NULL	VARCHAR2(60)	Expenditure organization
SUPPLIER_ID		NUMBER	The identification code of the supplier
SUPPLIER_NAME		VARCHAR2(80)	The name of the supplier
EXPENDITURE_TYPE		VARCHAR2(30)	Expenditure type
EVENT_TYPE		VARCHAR2(30)	Event type
NON_LABOR_RESOURCE		VARCHAR2(20)	Non-labor resource
EXPENDITURE_CATEGORY		VARCHAR2(30)	Expenditure category
REVENUE_CATEGORY		VARCHAR2(30)	Revenue category
NON_LABOR_RESOURCE_ORG_ID		NUMBER	The identification code of the non-labor resource organization
NON_LABOR_RESOURCE_ORG		VARCHAR2(60)	Non-labor resource organization
EVENT_TYPE_CLASSIFICATION		VARCHAR2(30)	Event type classification
SYSTEM_LINKAGE_FUNCTION		VARCHAR2(30)	Expenditure type class
WEEK_ENDING_DATE	NOT NULL	VARCHAR2(10)	Week ending date
REVENUE		NUMBER	Revenue
RAW_COST		NUMBER	Raw cost
CAPITAL_RAW_COST		NUMBER	Capital raw cost
BURDENED_COST		NUMBER	Burdened cost
CAPITAL_BURDENED_COST		NUMBER	Capital burdened cost
QUANTITY		NUMBER	Quantity

Oracle Proprietary, Confidential Information--Use Restricted by Contract

LABOR_HOURS	NUMBER	Labor hours
BILLABLE_RAW_COST	NUMBER	Billable raw cost
BILLABLE_BURDENED_COST	NUMBER	Billable burdened cost
BILLABLE_QUANTITY	NUMBER	Billable quantity
BILLABLE_LABOR_HOURS	NUMBER	Billable labor hours
CMT_RAW_COST	NUMBER	Commitment raw cost
CMT_BURDENED_COST	NUMBER	Commitment burdened cost
CMT_QUANTITY	NUMBER	Commitment quantity

Procedure Definitions

UPDATE_EARNED_VALUE

UPDATE_EARNED_VALUE is a PL/SQL procedure that updates earned value information in the PA_EARNED_VALUES table for lowest task–resource combinations. You can also use this procedure to update project–task rows.

Business Rules

- This procedure creates a new row in the table PA_EARNED_VALUES. CURRENT_FLAG is always set to Y for the last row inserted for each project, task, and resource combination. CURRENT_FLAG for all other rows is set to N.
- To create a project–task row, pass zero for the RESOURCE_LIST_MEMBER_ID parameter. To create a project row, pass zero for both the TASK_ID and RESOURCE_LIST_MEMBER_ID parameters.

Note: This API assumes that the vendor of the external system maintains the appropriate earned value data for all levels in any given project.

The following table shows the parameters for UPDATE_EARNED_VALUE .

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API version number
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Initial message table (default = 'F')
P_COMMIT	IN	VARCHAR2(1)	No	Commit (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		Return status
P_MSG_COUNT	OUT	NUMBER		Message count
P_MSG_DATA	OUT	VARCHAR2(2000)		Message
P_PROJECT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that identifies the project in the external system
P_TASK_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the task within a project in Oracle Projects

Name	Usage	Type	Req?	Description
P_PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_RESOURCE_LIST_MEMBER_ID	IN	NUMBER	Yes	The identification code of the resource list member
P_RESOURCE_ALIAS	IN	VARCHAR2(30)	No	The alias of the resource
P_RESOURCE_LIST_NAME	IN	VARCHAR2(60)	No	The name of the resource list
P_AS_OF_DATE	IN	DATE	Yes	As-of date
P_BCWS_CURRENT	IN	NUMBER	No	Budget cost of work performed
P_ACWP_CURRENT	IN	NUMBER	No	Actual cost of work performed
P_BCWP_CURRENT	IN	NUMBER	No	Budget cost of work performed
P_BAC_CURRENT	IN	NUMBER	No	Budget cost at completion
P_BCWS_ITD	IN	NUMBER	Yes	Inception-to-date budget cost of work performed
P_ACWP_ITD	IN	NUMBER	Yes	Inception-to-date actual cost of work performed
P_BCWP_ITD	IN	NUMBER	Yes	Inception-to-date budget cost of work performed
P_BAC_ITD ITD	IN	NUMBER	Yes	Inception-to-date budget cost at completion
P_BQWS_CURRENT	IN	NUMBER	No	Budget quantity of work performed
P_AQWP_CURRENT	IN	NUMBER	No	Actual quantity of work performed
P_BQWP_CURRENT	IN	NUMBER	No	Budget quantity of work performed
P_BAQ_CURRENT	IN	NUMBER	No	Budget quantity at completion
P_BQWS_ITD	IN	NUMBER	Yes	Inception-to-date budget quantity of work performed
P_AQWP_ITD	IN	NUMBER	Yes	Inception-to-date actual quantity of work performed
P_BQWP_ITD	IN	NUMBER	Yes	Inception-to-date budget quantity of work performed
P_BAQ_ITD	IN	NUMBER	Yes	Inception-to-date budget quantity at completion

UPDATE_PROGRESS

UPDATE_PROGRESS is a PL/SQL procedure that updates progress information in the PA_PERCENT_COMPLETES table as of a given date for all levels of the work breakdown structure.

For a given project, a task identifier of zero shows that the parameters apply to a project-level row. A task identifier greater than zero shows that the parameters apply to a task-level row.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Adding or deleting tasks from a project's work breakdown structure does not affect their corresponding rows in the PA_PERCENT_COMPLETES table. When executed, this API inserts a new row in the PA_PERCENT_COMPLETES table if a row for that project-task combination does not already exist.

Business Rules

- This procedure creates a new row in the table PA_PERCENT_COMPLETES. CURRENT_FLAG is always set to Y for the last row inserted for each project, task, and resource combination. CURRENT_FLAG for all other rows is set to N.
- To create a project row, you must pass zero in TASK_ID.

Note: This API assumes that vendor of the external system maintains the appropriate rollup of progress data for each level of the work breakdown structure for any given project. Providing progress information, however, is optional.

The following table shows the parameters for UPDATE_PROGRESS.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API version number
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	Initial message table (default = 'F')
P_COMMIT	IN	VARCHAR2(1)	No	Commit (default = 'F')
P_RETURN_STATUS	OUT	VARCHAR2(1)		Return status
P_MSG_COUNT	OUT	NUMBER		Message count
P_MSG_DATA	OUT	VARCHAR2(2000)		Message
P_PROJECT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the project in Oracle Projects
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the project in the external system
P_TASK_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the task within a project in Oracle Projects
P_PM_TASK_REFERENCE	IN	VARCHAR2(30)	No	The reference code that uniquely identifies the task in the external system
P_AS_OF_DATE	IN	DATE	Yes	As-of-date
P_PERCENT_COMPLETE	IN	NUMBER	Yes	Percent complete

Oracle Proprietary, Confidential Information--Use Restricted by Contract

CHAPTER

9

Detailed Design -- Agreement and Funding APIs

This chapter describes how to implement AMG APIs for agreements and funding, and provides detailed view and procedure definitions.

Agreement and Funding APIs

The agreement and funding APIs in the Activity Management Gateway (AMG) provide an open interface for external systems to insert, update, and delete agreements, as well as allocate funds from one agreement to any number of projects or top-level tasks.

Security

Actions performed using the APIs are subject to data level security (Control Actions). However, no function security is enforced. To maintain the same level of security as Oracle Projects, the APIs can only be executed either through AMG or Oracle Applications. This enables you to log in to the database, choose a valid responsibility, and only access the APIs that the responsibility allows.

These APIs provide the ability to copy components from the agreements and funding form to create and maintain agreements and fundings.

For an overview of the AMG views and APIs, see *Overview of Detailed Design*: page 4 – 2.

Control Actions

The following new Control Actions have been added for Agreement/Funding AMG functionality:

- Update Agreement
- Delete Agreement
- Add Funding
- Update Funding
- Delete Funding

For more information on the control actions, see *Control Actions Window*, *Oracle Projects User Guide*.

Views and Procedures

The views and procedures discussed in this section are listed below. The procedures are located in the public API package PA_AGREEMENT_PUB.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- Views
 - PA_AGREEMENT_TYPE_LOV_V: page 9 – 4
 - PA_TERMS_LOV_V: page 9 – 4
 - PA_OWNED_BY_LOV_V: page 9 – 5
 - PA_CUSTOMERS_LOV_V: page 9 – 5
- Agreement and Funding Procedures
 - CREATE_AGREEMENT: page 9 – 6
 - DELETE_AGREEMENT: page 9 – 7
 - UPDATE_AGREEMENT: page 9 – 8
 - ADD_FUNDING: page 9 – 9
 - DELETE_FUNDING: page 9 – 11
 - UPDATE_FUNDING: page 9 – 12
 - INIT_AGREEMENT: page 9 – 14
 - LOAD_AGREEMENT: page 9 – 14
 - LOAD_FUNDING: page 9 – 15
 - EXECUTE_CREATE_AGREEMENT: page 9 – 16
 - EXECUTE_UPDATE_AGREEMENT: page 9 – 17
 - FETCH_FUNDING: page 9 – 18
 - CLEAR_AGREEMENT: page 9 – 19
 - CHECK_DELETE_AGREEMENT_OK: page 9 – 19
 - CHECK_ADD_FUNDING_OK: page 9 – 20
 - CHECK_DELETE_FUNDING_OK: page 9 – 21
 - CHECK_UPDATE_FUNDING_OK: page 9 – 22

View Definitions

PA_AGREEMENT_TYPE_LOV_V

PA_AGREEMENT_TYPE_LOV_V retrieves valid agreement types.

Column descriptions for PA_AGREEMENT_TYPE_LOV_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
AGREEMENT_TYPE	NOT NULL	VARCHAR2(30)	The implementation-defined name that uniquely identifies the agreement type
REVENUE_LIMIT	NOT NULL	VARCHAR2(1)	Default flag to agreements that indicates whether revenue and invoices for projects funded by the agreement can exceed the allocated funding amount
TERM_ID	NOT NULL	NUMBER(15)	The identifier of the payment term that defaults to agreements and specifies the schedule on which invoices funded by the agreement are to be paid
TERM_NAME	NOT NULL	VARCHAR2(15)	The name of the payment term

PA_TERMS_LOV_V

PA_TERMS_LOV_V retrieves customer terms.

Column descriptions for PA_TERMS_LOV_V are shown in the following table.

Column Descriptions

Name	Null?	Type	Description
AGREEMENT_TYPE	NOT NULL	VARCHAR2(30)	The implementation-defined name that uniquely identifies the agreement type
REVENUE_LIMIT	NOT NULL	VARCHAR2(1)	Default flag to agreements that indicates whether revenue and invoices for projects funded by the agreement can exceed the allocated funding amount
TERM_ID	NOT NULL	NUMBER(15)	The identifier of the payment term that defaults to agreements and specifies the schedule on which invoices funded by the agreement are to be paid
TERM_NAME	NOT NULL	VARCHAR2(15)	The name of the payment term

Oracle Proprietary, Confidential Information--Use Restricted by Contract

PA_OWNED_BY_LOV_V

PA_OWNED_BY_LOV_V retrieves valid employees.

Column descriptions for PA_OWNED_BY_LOV_V are shown in the following table.

Column Descriptions

<u>Name</u>	<u>Null?</u>	<u>Type</u>	<u>Description</u>
PERSON_ID	NOT NULL	NUMBER(9)	Identifier of the person
NAME	NOT NULL	VARCHAR2(15)	Name of the person
EMPLOYEE_NUMBER	NOT NULL	VARCHAR2(9)	Employee number.

PA_CUSTOMERS_LOV_V

PA_CUSTOMERS_LOV_V retrieves valid customer names and numbers.

Column descriptions for PA_CUSTOMERS_LOV_V are shown in the following table.

Column Descriptions

<u>Name</u>	<u>Null?</u>	<u>Type</u>	<u>Description</u>
CODE	NOT NULL	NUMBER(15)	Identifier of the customer
DESCRIPTION	NOT NULL	VARCHAR2(50)	Customer name
CUSTOMER_NUMBER	NOT NULL	VARCHAR2(30)	Customer number

Procedure Definitions

CREATE_AGREEMENT

This API creates an agreement with associated funds.

Note: To use this API you must have a database environment that is capable of supporting the PL/SQL table and a user defined record (for example, Oracle Server 7.3 and PL/SQL 2.3). Otherwise, use the Load-Execute-Fetch APIs supplied in the pa_agreement_pub_package.

Business Rules

List of values

- Customer number
- Agreement type
- Agreement number
- Term name
- Revenue limit
- Valid Employee

The following table shows the parameters for CREATE_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_AGREEMENT_IN_REC	IN	AGREEMENT_REC_IN_TYPE	Yes	The reference code that uniquely identifies the agreement input record in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_AGREEMENT_OUT_REC	OUT	AGREEMENT_REC_OUT_TYPE		The reference code that uniquely identifies the agreement output record in Oracle Projects.
P_FUNDING_IN_TBL	IN	FUNDING_IN_TBL_TYPE	No	The reference code that uniquely identifies the funding input record in Oracle projects.
P_FUNDING_OUT_TBL	OUT	FUNDING_OUT_TBL_TYPE		The reference code that uniquely identifies the funding output record in Oracle Projects.

DELETE_AGREEMENT

This API deletes an agreement and associated funds.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.
- If the funding is baselined, the agreement cannot be deleted.
- Check accrued or billed amount:

agreement amount >= total funding amount >=0

AND

total funding amount >= amount accrued or billed

The following table shows the parameters for DELETE_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_PM_AGREEMENT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the agreement in the external system.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.

UPDATE_AGREEMENT

This API updates an agreement and associated funds.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.
- The agreement amount cannot be less than the sum of the total baselined amount and unbaselined amount.
- The customer cannot be changed if there is one fund for the agreement.
- List of Values
 - Customer number
 - Agreement type
 - Agreement number
 - Term name
 - Revenue limit
 - Valid employee

The following table shows the parameters for UPDATE_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	YES	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER	Yes	API standard
P_MSG_DATA	OUT	VARCHAR2(2000)	Yes	API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_RETURN_STATUS	OUT	VARCHAR2(1)	Yes	API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_AGREEMENT_IN_REC	IN	AGREEMENT_REC_IN_TYPE	Yes	The reference code that uniquely identifies the agreement input record in Oracle Projects.
P_AGREEMENT_OUT_REC	OUT	AGREEMENT_REC_OUT_TYPE		The reference code that uniquely identifies the agreement output record in Oracle Projects.
P_FUNDING_IN_TBL	IN	FUNDING_IN_TBL_TYPE	No	The reference code that uniquely identifies the funding input record in Oracle projects.
P_FUNDING_OUT_TBL	OUT	FUNDING_OUT_TBL_TYPE		The reference code that uniquely identifies the funding output record in Oracle Projects.

ADD_FUNDING

This API adds funding to an agreement.

Business Rules

- If the project is funded by multiple customers, funding cannot be done at the task level.
- If the project is funded by one customer, multiple agreements generate an error message.
- If the Project Type is not Contract, the fund amount must be zero.
- If the funding is baselined, the funding amount cannot be updated.
- Check funding level: If there is an existing Project Level Funding, there cannot also be a Top Task Level Funding. A project can only have one funding level.
- Check accrued or billed amount:
 agreement amount >= total funding amount >=0
 AND

Oracle Proprietary, Confidential Information--Use Restricted by Contract

total funding amount >= amount accrued or billed

The following table shows the parameters for ADD_FUNDING.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the funding in the external system.
P_FUNDING_ID	IN OUT	NUMBER(15)	Yes	The reference code that uniquely identifies the funding in Oracle Projects.
P_PA_PROJECT_ID	IN	NUMBER(15)	Yes	The reference code that uniquely identifies the project in Oracle Projects.
P_PA_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_ALLOCATED_AMOUNT	IN	NUMBER	Yes	The reference code that uniquely identifies the allocated funding amount within a project in Oracle Projects.
P_DATE_ALLOCATED	IN	DATE	No	The reference code that uniquely identifies the date allocated within a project in Oracle Projects.
P_DESC_FLEX_NAME	IN	VARCHAR2	No	Descriptive flexfield name
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Descriptive flexfield category
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Descriptive flexfield category

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Descriptive flexfield category
P_FUNDING_ID_OUT	OUT	NUMBER		The reference code that uniquely identifies the funding within a project in Oracle Projects.

DELETE_FUNDING

This API deletes a fund from an agreement.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.
- If the funding is baselined, the agreement cannot be deleted.
- Check accrued or billed amount:

agreement amount >= total funding amount >=0

AND

total funding amount >= amount accrued or billed

The following table shows the parameters for DELETE_FUNDING.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the supplier funding in the external system.
P_FUNDING_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the funding within a project in Oracle Projects.
P_FUNDING_ID_OUT	OUT	NUMBER		The reference code that uniquely identifies the funding (outflows) within a project in Oracle Projects.

UPDATE_FUNDING

This API updates a fund for an agreement.

Business Rules

- If the project is funded by multiple customers, task level funding is not allowed.
- If the project is funded by one customer, multiple agreements generate an error message.
- If the Project Type is not Contract, the fund amount must be zero.
- If the funding is baselined, the funding amount cannot be updated.
- Check funding level: If there is an existing Project Level Funding, there cannot also be a Top Task Level Funding. A project can only have one funding level.
- Check accrued or billed amount:
 agreement amount >= total funding amount >=0
 AND
 total funding amount >= amount accrued or billed

The following table shows the parameters for UPDATE_FUNDING.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the supplier funding in the external system.
P_FUNDING_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the funding within a project in Oracle Projects.
P_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies a project in Oracle Projects.
P_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies a task within a project in Oracle Projects.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_ALLOCATED_AMOUNT	IN	NUMBER	No	The reference code that uniquely identifies the amount of funding allocated within a project in Oracle Projects.
P_DATE_ALLOCATED	IN	DATE	No	The reference code that uniquely identifies the allocated date within a project in Oracle Projects.
P_DESCFLEX_NAME	IN	DATE	No	Descriptive flexfield name
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Descriptive flexfield category
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_FUNDING_ID_OUT	OUT	NUMBER		The reference code that uniquely identifies the funding (outflows) within a project in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

INIT_AGREEMENT

This API sets the global tables used by the Load–Execute–Fetch procedures that create a new agreement or update an existing agreement.

Parameters: None

LOAD_AGREEMENT

This API loads an agreement to a PL/SQL record.

The following table shows the parameters for LOAD_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_AGREEMENT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the agreement in the external system.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement within a project in Oracle Projects.
P_CUSTOMER_ID	IN	NUMBER	Yes	The identification code of the project's customer in Oracle Projects.
P_CUSTOMER_NAME	IN	VARCHAR2(50)	Yes	The identification name of the project's customer in Oracle Projects.
P_CUSTOMER_NUM	IN	VARCHAR2(30)	Yes	The identification number of the project's customer in Oracle Projects.
P_AGREEMENT_NUM	IN	VARCHAR2(20)	Yes	The reference code that uniquely identifies a agreement number within a project in Oracle Projects.
P_AGREEMENT_TYPE	IN	VARCHAR2(30)	Yes	The reference code that uniquely identifies a agreement type within a project in Oracle Projects.
P_AMOUNT	IN	NUMBER	Yes	The reference code that uniquely identifies the amount of the agreement within a project in Oracle Projects.
P_TERM_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the terms of the agreement within a project in Oracle Projects.
P_TERM_NAME	IN	VARCHAR2(15)	Yes	The name that uniquely identifies the term of the agreement within a project in Oracle Projects.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_REVENUE_LIMIT_FLAG	IN	VARCHAR2(1)	No	Indicates whether or not the revenue limit has been exceeded.
P_EXPIRATION_DATE	IN	DATE	No	Indicates the expiration date of the agreement within a project in Oracle Projects.
P_DESCRIPTION	IN	VARCHAR2(240)	No	Description of the agreement within a project in Oracle Projects.
P_OWNED_BY_PERSON_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the person who owns the agreement within a project in Oracle Projects.
P_OWNED_BY_PERSON_NAME	IN	VARCHAR2(240)	Yes	The name that uniquely identifies the person who owns the agreement within a project in Oracle Projects.
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Descriptive flexfield category
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_TEMPLATE_FLAG	IN	VARCHAR2(1)	No	Indicates whether or not the project is a template.
P_DESC_FLEX_NAME	IN	VARCHAR2(40)	No	Descriptive flexfield name

LOAD_FUNDING

This API loads funding to a PL/SQL table.

The following table shows the parameters for LOAD_FUNDING.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the funding in external system.
P_FUNDING_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the funding in Oracle Projects.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_PROJECT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the project in Oracle Projects.
P_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects.
P_ALLOCATED_AMOUNT	IN	NUMBER	Yes	The reference code that uniquely identifies the amount of funding allocated within a project in Oracle Projects.
P_DATE_ALLOCATED	IN	DATE	No	The reference code that uniquely identifies the date funding was allocated within a project in Oracle Projects.
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Descriptive flexfield category
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Descriptive flexfield attribute

EXECUTE_CREATE_AGREEMENT

This API creates an agreement with the funding using the data stored in the global tables during the Load phase.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Business Rules

List of values

- Customer number
- Agreement type
- Agreement number
- Term name
- Revenue limit
- Valid Employee

The following table shows the parameters for EXECUTE_CREATE_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_AGREEMENT_ID_OUT	OUT	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_CUSTOMER_ID_OUT	OUT	NUMBER	Yes	The reference code that uniquely identifies the customer in Oracle Projects.

EXECUTE_UPDATE_AGREEMENT

This API updates an agreement with the funding using the data stored in the global tables during the Load phase.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.

Oracle Proprietary, Confidential Information--Use Restricted by Contract

- The agreement amount cannot be less than the sum of the total baselined amount and unbaselined amount.
- The customer cannot be changed if there is one fund for the agreement.
- List of Values
 - Customer number
 - Agreement type
 - Agreement number
 - Term name
 - Revenue limit
 - Valid employee

The following table shows the parameters for EXECUTE_UPDATE_AGREEMENT.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.

FETCH_FUNDING

This API gets the return_status that was returned during creation of funds and stored in a global PL/SQL table.

The following table shows the parameters for FETCH_FUNDING.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	YES	API standard
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_FUNDING_INDEX	IN	NUMBER	Yes	Pointer to specific funding amount

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_FUNDING_ID	OUT	NUMBER		The reference code that uniquely identifies the funding in Oracle Projects.
P_PM_FUNDING_REFERENCE	OUT	VARCHAR2(35)		The reference code that uniquely identifies the funding in the external system.

CLEAR_AGREEMENT

This API clears the globals that were set up during initialization.

Parameters: None

CHECK_DELETE_AGREEMENT_OK

This API checks whether an agreement can be deleted.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.
- If the funding is baselined, the agreement cannot be deleted.
- Check accrued or billed amount:

agreement amount >= total funding amount >=0

AND

total funding amount >= amount accrued or billed

The following table shows the parameters for CHECK_DELETE_AGREEMENT_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_AGREEMENT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the agreement in the external system.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_DEL_AGREE_OK_FLAG	OUT	VARCHAR2(1)		Boolean flag for deleting agreement

CHECK_ADD_FUNDING_OK

This API checks whether a fund can be added.

Business Rules

- If the project is funded by multiple customers, task level funding is not allowed.
- If the project is funded by one customer, multiple agreements generate an error message.
- If the project type is not Contract, the fund amount must be zero.
- If the funding is baselined, the funding amount cannot be updated.
- The funding level must be valid: If there is an existing Project Level Funding, there cannot also be a Top Task Level Funding. A project can only have one funding level.
- The accrued/billed amount must be valid:
agreement amount >= total funding amount >=0

AND

total funding amount >= amount accrued or billed

The following table shows the parameters for CHECK_ADD_FUNDING_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_AGREEMENT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the agreement in the external system.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle Projects.
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the funding in the external system.
P_TASK_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the task within the project in Oracle Projects.
P_PROJECT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the project in Oracle Projects.
P_ADD_FUNDING_OK_FLAG	OUT	VARCHAR2(1)		Boolean flag for adding funding

CHECK_DELETE_FUNDING_OK

This API checks whether a fund can be deleted.

Business Rules

- If there is at least one summary project funding that exists where the sum of the baselined amount and total unbaselined amount is less than the revenue accrued or billed amount, the API does not allow the revenue limit to be changed.
- If the funding is baselined, the agreement cannot be deleted.
- Check accrued or billed amount:
 agreement amount >= total funding amount >=0
 AND
 total funding amount >= amount accrued or billed

The following table shows the parameters for CHECK_DELETE_FUNDING_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the funding in the external system.
P_FUNDING_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the funding in Oracle Projects.
P_DEL_FUNDING_OK_FLAG	OUT	VARCHAR2(1)		Boolean flag for deleting funding

CHECK_UPDATE_FUNDING_OK

This API checks whether a fund can be added.

Business Rules

- If the project is funded by multiple customers, task level funding is not allowed.
- If the project type is not Contract, the fund amount must be zero.
- If the funding is baselined, the funding amount cannot be updated.
- Funding level checks
 - If there is no task ID , there can be no task level funding.
 - If there is a task ID, there can be no project level funding.
- Check accrued/billed amount
 - agreement amount >= total funding amount >=0 AND
 - total funding amount >= amount accrued or billed

The following table shows the parameters for CHECK_UPDATE_FUNDING_OK.

Name	Usage	Type	Req?	Description
P_API_VERSION_NUMBER	IN	NUMBER	Yes	API standard
P_COMMIT	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_INIT_MSG_LIST	IN	VARCHAR2(1)	No	API standard (Default = 'F')
P_MSG_COUNT	OUT	NUMBER		API standard
P_MSG_DATA	OUT	VARCHAR2(2000)		API standard
P_RETURN_STATUS	OUT	VARCHAR2(1)		API standard
P_PM_PRODUCT_CODE	IN	VARCHAR2(30)	Yes	The identifier of the external project management system from which the project was imported.
P_PM_FUNDING_REFERENCE	IN	VARCHAR2(35)	Yes	The reference code that uniquely identifies the funding in the external system.
P_FUNDING_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the funding in Oracle Projects.
P_PM_PROJECT_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the project in the external system.
P_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects.
P_PM_TASK_REFERENCE	IN	VARCHAR2(25)	No	The reference code that uniquely identifies the task in the external system.
P_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task within a project in Oracle Projects.
P_PM_AGREEMENT_REFERENCE	IN	VARCHAR2(25)	Yes	The reference code that uniquely identifies the agreement in the external system.
P_AGREEMENT_ID	IN	NUMBER	Yes	The reference code that uniquely identifies the agreement in Oracle projects.
P_ALLOCATED_AMOUNT	IN	NUMBER	No	The reference code that uniquely identifies the amount of funding allocated within a project in Oracle Projects.
P_DATE_ALLOCATED	IN	DATE	No	The reference code that uniquely identifies the date funding was allocated within a project in Oracle Projects.
P_DESC_FLEX_NAME	IN	VARCHAR2(40)	No	Descriptive flexfield name
P_ATTRIBUTE_CATEGORY	IN	VARCHAR2(30)	No	Descriptive flexfield category
P_ATTRIBUTE1	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE2	IN	VARCHAR2(150)	No	Descriptive flexfield attribute

Oracle Proprietary, Confidential Information--Use Restricted by Contract

Name	Usage	Type	Req?	Description
P_ATTRIBUTE3	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE4	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE5	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE6	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE7	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE8	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE9	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_ATTRIBUTE10	IN	VARCHAR2(150)	No	Descriptive flexfield attribute
P_UPDATE_FUNDING_OK_FLAG	OUT	VARCHAR2(1)		Boolean flag for deleting funding

Oracle Proprietary, Confidential Information--Use Restricted by Contract

APPENDIX

A

Creating a Project Using Load–Execute–Fetch APIs

This appendix provides sample PL/SQL code that you can use to create a project using the Load–Execute–Fetch APIs.

Creating a Project Using the Load-Execute-Fetch APIs

The following sample PL/SQL code is a sample of a script that you can use to create a project using the Load-Execute-Fetch APIs.

The Load-Execute-Fetch APIs use parameters with standard datatypes (VARCHAR2, NUMBER, and DATE). They do not use composite datatypes.

```
DECLARE
--variables needed to create task hierarchy

level1                NUMBER;
level2                NUMBER;
level3                NUMBER;
a                     NUMBER := 0;
m                     NUMBER := 0;
parent_level1        VARCHAR2(30);
parent_level2        VARCHAR2(30);
parent_level3        VARCHAR2(30);
number_of_tasks1     NUMBER; --number of tasks/level
number_of_tasks2     NUMBER;
number_of_tasks3     NUMBER;
number_of_tasks4     NUMBER;

--variables needed for API standard parameters

l_api_version_number  NUMBER :=1.0      ;
l_commit              VARCHAR2(1) := 'F';
l_return_status       VARCHAR2(1);
l_init_msg_list       VARCHAR2(1);
l_msg_count           NUMBER;
l_msg_data            VARCHAR2(2000);
l_data               VARCHAR2(2000);
l_msg_entity          VARCHAR2(100);
l_msg_entity_index    NUMBER;
l_msg_index           NUMBER;
l_msg_index_out       NUMBER;
l_encoded             VARCHAR2(1);

--variables needed for Oracle Project specific parameters

l_created_from_project_id  NUMBER;
l_pm_product_code          VARCHAR2(10);
l_number_of_task_levels    NUMBER;
```

```

l_project_name          VARCHAR2(30);
l_pm_project_reference  VARCHAR2(25);
l_project_status_code   VARCHAR2(30);
l_distribution_rule     VARCHAR2(30);
l_public_sector_flag    VARCHAR2(1);
l_carrying_out_organization_id NUMBER;
l_start_date            DATE;
l_completion_date       DATE;
l_actual_start_date     DATE;
l_actual_finish_date    DATE;
l_early_start_date     DATE;
l_early_finish_date     DATE;
l_late_start_date       DATE;
l_late_finish_date      DATE;
l_person_id             NUMBER;
l_project_role_type     VARCHAR2(20);
l_class_category        VARCHAR2(30);
l_class_code            VARCHAR2(30);
l_project_id            NUMBER(15);
l_pa_project_number     VARCHAR2(25);
l_project_description   VARCHAR2(250);
l_customer_id           NUMBER;
l_project_relationship_code VARCHAR2(30);
l_task_id               NUMBER(15);
l_pm_task_reference     VARCHAR2(25);
l_task_index            NUMBER;
l_tasks_in              pa_project_pub.task_in_tbl_type;
l_task_rec              pa_project_pub.task_in_rec_type;
l_key_member_rec        pa_project_pub.project_role_rec_type;
l_key_member_tbl        pa_project_pub.project_role_tbl_type;
l_task_return_status    VARCHAR2(1);
API_ERROR               EXCEPTION;

```

```
BEGIN
```

```
--PRODUCT RELATED DATA
```

```
l_pm_product_code := 'SOMETHING';
```

```
--PROJECT DATA
```

```
l_created_from_project_id := 1040;
```

```
l_project_name := 'PROJECT_NAME';
```

```
l_pm_project_reference := 'PROJECT_NAME';
```

```
l_project_description := 'PROJECT_DESCRIPTION';
```

```
l_project_status_code := '';
```

```

--l_distribution_rule := 'COST/COST';
l_carrying_out_organization_id :=2;
l_start_date :='01-jan-94';
l_completion_date :='31-mar-99';
l_actual_start_date :='01-jan-93';
l_actual_finish_date :='01-apr-99';
l_early_start_date :='01-jan-94';
l_early_finish_date :='31-mar-99';
l_late_start_date :='01-jan-94';
l_late_finish_date :='31-mar-99';

--KEY MEMBERS DATA

m:= 1;

l_person_id :='29';
l_project_role_type :='PROJECT MANAGER';
l_key_member_rec.person_id :=29;
l_key_member_rec.project_role_type :='PROJECT MANAGER';
l_key_member_tbl(m) := l_key_member_rec;

m:=2;

l_key_member_rec.person_id :=30;
l_key_member_rec.project_role_type :='Project Coordinator';

l_key_member_tbl(m) := l_key_member_rec;

m:=3;

l_key_member_rec.person_id :=7;
l_key_member_rec.project_role_type :='Project Coordinator';

l_key_member_tbl(m) := l_key_member_rec;

--CLASS CATEGORIES DATA
l_class_category :='Funding Source';
l_class_code :='Federal';

--TASKS DATA

--Set the number of tasks for every level (there are 4 levels)
number_of_tasks1 := 10;

```

```

number_of_tasks2 := 1;
number_of_tasks3 := 1;
number_of_tasks4 := 0;

for level1 in 1..number_of_tasks1 loop

    a:= a + 1;
    l_task_rec.pm_task_reference :=a;
    l_task_rec.task_name :='TOP LEVEL '||a;
    l_task_rec.pm_parent_task_reference :='';
        l_task_rec.task_start_date := '09-MAR-95';
        l_task_rec.task_completion_date := '05-JUL-95';
        l_task_rec.actual_start_date := '10-MAR-95';
        l_task_rec.actual_finish_date := '06-JUL-95';
        l_task_rec.early_start_date := '09-MAR-95';
        l_task_rec.early_finish_date := '05-JUL-95';
        l_task_rec.late_start_date := '09-MAR-95';
        l_task_rec.late_finish_date := '05-JUL-95';

    --l_task_rec.address_id := 1012;
    l_tasks_in(a):= l_task_rec;
    parent_level1:= a;

FOR level2 IN 1..number_of_tasks2 LOOP
    a:= a + 1;
    l_task_rec.pm_task_reference :=a;
    l_task_rec.task_name :='2 LEVEL '||a;
    l_task_rec.pm_parent_task_reference := parent_level1;
    l_tasks_in(a) := l_task_rec;
    parent_level2 := a;

for level3 IN 1..number_of_tasks3 loop
    a := a + 1;
    l_task_rec.pm_task_reference := a;
    l_task_rec.task_name :='3 LEVEL '||a;
    l_task_rec.pm_parent_task_reference := parent_level2;
    l_tasks_in(a) := l_task_rec;
    parent_level3 := a;
    for level4 IN 1..number_of_tasks4 loop
        a := a + 1;
        l_task_rec.pm_task_reference := a;
        l_task_rec.task_name :='Fourth LEVEL '||a;
        l_task_rec.pm_parent_task_reference := parent_level3;
        l_tasks_in(a) := l_task_rec;

```

```

        end loop;

    end loop;

    END LOOP;
end loop;
-----
--INIT_CREATE_PROJECT
pa_project_pub.init_project;
-----
--LOAD_PROJECT
pa_project_pub.load_project( p_api_version_number => l_api_version_number
    ,p_return_status => l_return_status
    ,p_created_from_project_id => l_created_from_project_id
    ,p_project_name => l_project_name
    ,p_description => l_project_description
    ,p_pm_project_reference => l_pm_project_reference
    ,p_pa_project_number => 'rk-test-number'
    ,p_carrying_out_organization_id =>
        l_carrying_out_organization_id
    ,p_public_sector_flag => l_public_sector_flag
    ,p_customer_id => l_customer_id
    ,p_project_status_code => l_project_status_code
    ,p_start_date => l_start_date
    ,p_completion_date => l_completion_date
    ,p_actual_start_date => l_actual_start_date
    ,p_actual_finish_date => l_actual_finish_date
    ,p_early_start_date => l_early_start_date
    ,p_early_finish_date => l_early_finish_date
    ,p_late_start_date => l_late_start_date
    ,p_late_finish_date => l_late_finish_date
    ,p_distribution_rule => l_distribution_rule);

IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--LOAD_KEY_MEMBER    (loop for multiple key members)

FOR i in 1..1 LOOP
pa_project_pub.load_key_member( p_api_version_number => l_api_version_number
    ,p_return_status => l_return_status
    ,p_person_id => l_key_member_tbl(i).person_id

```

```

                ,p_project_role_type =>
                    l_key_member_tbl(i).project_role_type );
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;
-----
--LOAD_CLASS_CATEGORY (loop for multiple class categories-This example has
-- only one )
FOR i IN 1..1 LOOP
pa_project_pub.load_class_category(
                    p_api_version_number => l_api_version_number
                ,p_return_status => l_return_status
                ,p_class_category => l_class_category
                ,p_class_code => l_class_code );
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;
-----
--LOAD_TASK (loop for multiple tasks)
FOR i IN 1..a LOOP
pa_project_pub.load_task( p_api_version_number => l_api_version_number
                ,p_return_status => l_return_status
                ,p_pm_task_reference => l_tasks_in(i).pm_task_reference
                ,p_task_name => l_tasks_in(i).task_name
                ,p_pm_parent_task_reference =>
                    l_tasks_in(i).pm_parent_task_reference
                ,p_task_start_date => l_tasks_in(i).task_start_date
                ,p_task_completion_date =>
                    l_tasks_in(i).task_completion_date
                ,p_actual_start_date => l_tasks_in(i).actual_start_date
                ,p_actual_finish_date => l_tasks_in(i).actual_finish_date
                ,p_early_start_date => l_tasks_in(i).early_start_date
                ,p_early_finish_date => l_tasks_in(i).early_finish_date
                ,p_late_start_date => l_tasks_in(i).late_start_date
                ,p_late_finish_date => l_tasks_in(i).late_finish_date
                ,p_address_id => l_tasks_in(i).address_id);
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;

```

```

END LOOP;
-----
--EXECUTE_CREATE_PROJECT
pa_project_pub.execute_create_project(p_api_version_number =>
                                     l_api_version_number
                                     ,p_commit => l_commit
                                     ,p_init_msg_list => 'F'
                                     ,p_msg_count => l_msg_count
                                     ,p_msg_data => l_msg_data
                                     ,p_return_status => l_return_status
                                     ,p_pm_product_code => l_pm_product_code
                                     ,p_pa_project_id => l_project_id
                                     ,p_pa_project_number => l_pa_project_number);
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--FETCH_TASK
FOR l_task_index in 1..a LOOP
pa_project_pub.fetch_task( p_api_version_number => l_api_version_number
                          ,p_return_status => l_return_status
                          ,p_task_index => l_task_index
                          ,p_pa_task_id => l_task_id
                          ,p_pm_task_reference => l_pm_task_reference
                          ,p_task_return_status => l_task_return_status);
IF l_return_status != 'S'
OR l_task_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;
-----
--CLEAR_CREATE_PROJECT
pa_project_pub.clear_project;
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--HANDLE EXCEPTIONS
EXCEPTION
WHEN API_ERROR THEN
    for i in 1..l_msg_count loop

```

```

pa_interface_utils_pub.get_messages (
    p_msg_data          => l_msg_data
    ,p_data => l_data
    ,p_msg_count => l_msg_count
    ,p_msg_index_out => l_msg_index_out );
dbms_output.put_line ('error mesg '||l_data);

end loop;
WHEN OTHERS THEN
for i in 1..l_msg_count loop
    pa_interface_utils_pub.get_messages (
        p_msg_count => l_msg_count
        ,p_msg_data => l_msg_data
        ,p_data => l_data
        ,p_msg_index_out => l_msg_index_out);

        dbms_output.put_line ('error mesg '||l_data);
end loop;
END ;
/

```


APPENDIX

B

Creating a Budget Using Load–Execute–Fetch APIs

This appendix provides sample PL/SQL code for creating a budget using the Load–Execute–Fetch APIs.

Creating a Budget Using the Load-Execute-Fetch APIs

The following sample PL/SQL code is a sample of a script you can use to create a budget using the Load-Execute-Fetch APIs.

The Load-Execute-Fetch APIs use parameters with standard datatypes (VARCHAR2, NUMBER, and DATE). They do not use composite datatypes. If you create budgets using tools or products that support composite datatypes, see [Creating a Budget Using a Composite Datatype API](#): page C - 2.

```
DECLARE
  --variables needed for API standard parameters
  l_api_version_number      NUMBER :=1.0;
  l_commit                  VARCHAR2(1):= 'F';
  l_return_status           VARCHAR2(1);
  l_init_msg_list           VARCHAR2(1);
  l_msg_count               NUMBER;
  l_msg_data                VARCHAR2(2000);
  l_data                    VARCHAR2(2000);
  l_msg_entity              VARCHAR2(100);
  l_msg_entity_index        NUMBER;
  l_msg_index               NUMBER;
  l_msg_index_out           NUMBER;
  l_encoded                 VARCHAR2(1);
  i                         NUMBER;
  a                         NUMBER;
  --variables needed for Oracle Project specific parameters
  l_pm_product_code         VARCHAR2(10);
  l_pa_project_id           NUMBER;
  l_pm_project_reference    VARCHAR2(25);
  l_budget_type_code        VARCHAR2(30);
  l_change_reason_code      VARCHAR2(30);
  l_description              VARCHAR2(255);
  l_entry_method_code        VARCHAR2(30);
  l_resource_list_name      VARCHAR2(60);
  l_resource_list_id        NUMBER;
  l_budget_lines_in         pa_budget_pub.budget_line_in_tbl_type;
  l_budget_lines_in_rec     pa_budget_pub.budget_line_in_rec_type;
  l_budget_lines_out        pa_budget_pub.budget_line_out_tbl_type;
  l_line_index              NUMBER;
  l_line_return_status      VARCHAR2(1);

  API_ERROR                 EXCEPTION;
BEGIN
```

```

--PRODUCT RELATED DATA
  l_pm_product_code := 'SOMETHING';
--BUDGET DATA
--l_pa_project_id:= 1138;
l_pm_project_reference := 'PROJECT_NAME';
l_budget_type_code := 'AC';
l_change_reason_code := 'ESTIMATING ERROR';
l_description := 'New description -> 2';
l_entry_method_code := 'PA_LOWEST_TASK_BY_DATE_RANGE';
l_resource_list_id := 1014;
--BUDGET LINES DATA
a := 5;
FOR i IN 1..a LOOP
  if i = 1 THEN
    l_budget_lines_in_rec.pa_task_id := 2440;
    l_budget_lines_in_rec.resource_list_member_id := 1401;
  elsif i = 2 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1402;
    l_budget_lines_in_rec.pa_task_id := 2443;
  elsif i = 3 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1404;
    l_budget_lines_in_rec.pa_task_id := 2446;
  elsif i = 4 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1407;
    l_budget_lines_in_rec.pa_task_id := 2449;
  elsif i = 5 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1408;
    l_budget_lines_in_rec.pa_task_id := 2452;
  end if;
  l_budget_lines_in_rec.quantity :=93;
  l_budget_lines_in_rec.budget_start_date := '05-MAY-95';
  l_budget_lines_in_rec.budget_end_date := '09-MAY-95';
  l_budget_lines_in_rec.raw_cost :=300;
  l_budget_lines_in(i) := l_budget_lines_in_rec;
END LOOP;
-----
--INIT_BUDGET
pa_budget_pub.init_budget;
-----
--LOAD_BUDGET_LINE
FOR i IN 1..a LOOP
pa_budget_pub.load_budget_line( p_api_version_number => l_api_version_number
                              ,p_return_status => l_return_status
                              ,p_pa_task_id => l_budget_lines_in(i).pa_task_id

```

```

        ,p_pm_task_reference => l_budget_lines_in(i).pm_task_reference
        ,p_resource_alias    => l_budget_lines_in(i).resource_alias
        ,p_resource_list_member_id =>
            l_budget_lines_in(i).resource_list_member_id
        ,p_budget_start_date => l_budget_lines_in(i).budget_start_date
        ,p_budget_end_date  => l_budget_lines_in(i).budget_end_date
        ,p_period_name      => l_budget_lines_in(i).period_name
        ,p_description      => l_budget_lines_in(i).description
        ,p_raw_cost         => l_budget_lines_in(i).raw_cost
        ,p_burdened_cost    => l_budget_lines_in(i).burdened_cost
        ,p_revenue          => l_budget_lines_in(i).revenue
        ,p_quantity         => l_budget_lines_in(i).quantity );

END LOOP;
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--EXECUTE_CREATE_DRAFT_BUDGET
pa_budget_pub.execute_create_draft_budget
( p_api_version_number => l_api_version_number
  ,p_msg_count         => l_msg_count
  ,p_msg_data         => l_msg_data
  ,p_return_status    => l_return_status
  ,p_pm_product_code  => l_pm_product_code
  ,p_pa_project_id    => l_pa_project_id
  ,p_pm_project_reference => l_pm_project_reference
  ,p_budget_type_code => l_budget_type_code
  ,p_change_reason_code => l_change_reason_code
  ,p_description      => l_description
  ,p_entry_method_code => l_entry_method_code
  ,p_resource_list_name => l_resource_list_name
  ,p_resource_list_id => l_resource_list_id    );
IF l_return_status != 'S'
THEN
    null; --RAISE API_ERROR;
END IF;
-----
--FETCH_LINE
FOR l_line_index in 1..PA_BUDGET_PUB.G_budget_lines_tbl_count LOOP
pa_budget_pub.fetch_budget_line( p_api_version_number => l_api_version_number
    ,p_return_status => l_return_status
    ,p_line_index => l_line_index
    ,p_line_return_status => l_line_return_status);

```

```

IF l_return_status != 'S'
OR l_line_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;
-----
--CLEAR_BUDGET
pa_budget_pub.clear_budget;
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--HANDLE EXCEPTIONS
EXCEPTION
WHEN API_ERROR THEN
    for i in 1..l_msg_count loop
        pa_interface_utils_pub.get_messages (
            p_msg_data => l_msg_data
            ,p_data => l_data
            ,p_msg_count => l_msg_count
            ,p_msg_index_out => l_msg_index_out );
        dbms_output.put_line ('error msg '||l_data);
        dbms_output.put_line ('error msg '||l_msg_data);

    end loop;
WHEN OTHERS THEN
    for i in 1..l_msg_count loop
        pa_interface_utils_pub.get_messages (
            p_msg_data => l_msg_data
            ,p_data => l_data
            ,p_msg_count => l_msg_count
            ,p_msg_index_out => l_msg_index_out );
        dbms_output.put_line ('error msg '||l_data);

    end loop;
END;
/

```


C

Creating a Budget Using a Composite Datatype API

This appendix provides sample PL/SQL code for creating a budget using an AMG API that uses composite datatypes.

Creating a Budget Using a Composite Datatype API

The following sample PL/SQL code is a script that creates a budget using the API CREATE_DRAFT_BUDGET, which uses composite datatypes. If you create budgets using tools or products that do not support composite datatypes, see [Creating a Budget Using the Load-Execute-Fetch APIs: page B - 2](#).

```
DECLARE
--variables needed for API standard parameters
l_api_version_number    NUMBER :=1.0;
l_commit                VARCHAR2(1):= 'F';
l_return_status         VARCHAR2(1);
l_init_msg_list        VARCHAR2(1);
l_msg_count            NUMBER;
l_msg_data             VARCHAR2(2000);
l_data                 VARCHAR2(2000);
l_msg_entity           VARCHAR2(100);
l_msg_entity_index     NUMBER;
l_msg_index            NUMBER;
l_msg_index_out        NUMBER;
l_encoded              VARCHAR2(1);
i                      NUMBER;
a                      NUMBER;
--variables needed for Oracle Projects-specific parameters
l_pm_product_code      VARCHAR2(10);
l_pa_project_id        NUMBER;
l_pm_project_reference VARCHAR2(25);
l_budget_type_code     VARCHAR2(30);
l_version_name         VARCHAR2(30);
l_change_reason_code   VARCHAR2(30);
l_description          VARCHAR2(255);
l_entry_method_code    VARCHAR2(30);
l_resource_list_name    VARCHAR2(60);
l_resource_list_id     NUMBER;
l_budget_lines_in      pa_budget_pub.budget_line_in_tbl_type;
l_budget_lines_in_rec  pa_budget_pub.budget_line_in_rec_type;
l_budget_lines_out     pa_budget_pub.budget_line_out_tbl_type;
l_line_index           NUMBER;
l_line_return_status   VARCHAR2(1);

API_ERROR              EXCEPTION;
BEGIN
--PRODUCT RELATED DATA
```

```

l_pm_product_code := 'SOMETHING';
--BUDGET DATA
l_pm_project_reference := 'PROJECT_NAME';
l_budget_type_code := 'AC'; '--AR'; --
l_change_reason_code := 'ESTIMATING ERROR';
l_description := 'New description 2';
l_version_name := 'New version ';
l_entry_method_code := 'PA_LOWEST_TASK_BY_DATE_RANGE';
l_resource_list_id := 1014;
--BUDGET LINES DATA
a := 5;
FOR i IN 1..a LOOP
  if i = 1 THEN
    l_budget_lines_in_rec.pa_task_id := 2440;
    l_budget_lines_in_rec.resource_list_member_id := 1401;
  elsif i = 2 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1402;
    l_budget_lines_in_rec.pa_task_id := 2443;
  elsif i = 3 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1404;
    l_budget_lines_in_rec.pa_task_id := 2446;
  elsif i = 4 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1407;
    l_budget_lines_in_rec.pa_task_id := 2449;
  elsif i = 5 THEN
    l_budget_lines_in_rec.resource_list_member_id := 1408;
    l_budget_lines_in_rec.pa_task_id := 2452;
  end if;
  l_budget_lines_in_rec.quantity :=93;
  l_budget_lines_in_rec.budget_start_date := '05-MAY-95';
  l_budget_lines_in_rec.budget_end_date := '09-MAY-95';
  l_budget_lines_in_rec.raw_cost :=300;
  l_budget_lines_in(i) := l_budget_lines_in_rec;
END LOOP;
-----
--INIT_BUDGET
pa_budget_pub.init_budget;
-----
--CREATE_DRAFT_BUDGET
pa_budget_pub.create_draft_budget
( p_api_version_number => l_api_version_number
,p_msg_count => l_msg_count
,p_msg_data => l_msg_data
,p_return_status => l_return_status

```

```

,p_pm_product_code => l_pm_product_code
,p_pa_project_id => l_pa_project_id
,p_pm_project_reference => l_pm_project_reference
,p_budget_type_code => l_budget_type_code
,p_change_reason_code => l_change_reason_code
,p_budget_version_name => l_version_name
,p_description => l_description
,p_entry_method_code => l_entry_method_code
,p_resource_list_name => l_resource_list_name
,p_resource_list_id => l_resource_list_id
,p_budget_lines_in => l_budget_lines_in
,p_budget_lines_out => l_budget_lines_out );
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--CLEAR_BUDGET

pa_budget_pub.clear_budget;
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-----
--HANDLE EXCEPTIONS
EXCEPTION
WHEN API_ERROR THEN
    for i in 1..l_msg_count loop
        pa_interface_utils_pub.get_messages (
            p_msg_data => l_msg_data
            ,p_data => l_data
            ,p_msg_count => l_msg_count
            ,p_msg_index_out => l_msg_index_out );
        dbms_output.put_line ('error mesg '||l_data);
    end loop;
WHEN OTHERS THEN
    for i in 1..l_msg_count loop
        pa_interface_utils_pub.get_messages (
            p_msg_data => l_msg_data
            ,p_data => l_data
            ,p_msg_count => l_msg_count
            ,p_msg_index_out => l_msg_index_out );
        dbms_output.put_line ('error mesg '||l_data);

```

```
        end loop;  
END;  
/
```


APPENDIX

D

Creating an Agreement Using the Load-Execute-Fetch APIs

This appendix provides sample PL/SQL code for creating an agreement using the Load-Execute-Fetch APIs.

Creating an Agreement Using Load-Execute-Fetch APIs

The following sample PL/SQL code is a script that creates an agreement using the Load-Execute-Fetch APIs. The Load-Execute-Fetch APIs use parameters with standard datatypes (VARCHAR2, NUMBER, and DATE). These APIs do not use composite datatypes.

To create agreements using tools or products that support composite datatypes, see [Creating an Agreement Using a Composite Datatype API](#): page E - 2.

```
DECLARE
  --API standard parameters

  l_api_version_number          NUMBER :=1.0;
  l_commit                     VARCHAR2(1) := 'T';
  l_return_status              VARCHAR2(1)
  l_init_msg_list              VARCHAR2(1)
  l_msg_count                  NUMBER;
  l_msg_data                   VARCHAR2(2000);
  l_data                       VARCHAR2(2000);
  l_msg_entity                 VARCHAR2(100);
  l_msg_entity_index          NUMBER;
  l_msg_index                  NUMBER;
  l_msg_index_out             NUMBER;
  l_encoded                    VARCHAR2(1)
  l_agreement_id_out          NUMBER;
  l_customer_id_out           NUMBER;
  l_funding_id                NUMBER;

  --Oracle agreement specific variable

  l_pm_product_code           VARCHAR2(25);
  l_agreement_in_rec          pa_agreement_pub.Agreement_Rec_In_Type;
  l_agreement_out_rec         pa_agreement_pub.Agreement_Rec_Out_Type;

  ..--Oracle funding specific parameters
  l_funding_type              pa_agreement_pub.funding_rec_in_type;
  l_funding_in_tbl            pa_agreement_pub.funding_in_tbl_type;
  l_funding_out_tbl           pa_agreement_pub.funding_out_tbl_type;

  --Local agreement parameters

  l_early_start_date          DATE;
  l_pm_agreement_reference    VARCHAR2(25);
```

```

l_agreement_id          NUMBER;
l_customer_id          NUMBER;
l_customer_name        VARCHAR2(25);
l_customer_num         VARCHAR2(25);
l_agreement_num        VARCHAR2(25);
l_agreement_type       VARCHAR2(25);
l_amount               NUMBER;
l_term_id              NUMBER;
l_term_name            VARCHAR2(25);
l_revenue_limit_flag   VARCHAR2(25);
l_expiration_date      DATE;
l_description          VARCHAR2(25);
l_owned_by_person_id   NUMBER;
l_owned_by_person_name VARCHAR2(25);
l_attribute_category   VARCHAR2(25);
l_attribute1           VARCHAR2(25);
l_attribute2           VARCHAR2(25);
l_attribute3           VARCHAR2(25);
l_attribute4           VARCHAR2(25);
l_attribute5           VARCHAR2(25);
l_attribute6           VARCHAR2(25);
l_attribute7           VARCHAR2(25);
l_attribute8           VARCHAR2(25);
l_attribute9           VARCHAR2(25);
l_attribute10          VARCHAR2(25);
l_template_flag       VARCHAR2(25);

..---local funding variables
l_pm_funding_reference VARCHAR2(25);
l_funding_rec           pa_agreement_pub.funding_rec_in_type;
l_funding_in           pa_agreement_pub.funding_in_tbl_type;

    ---loop variables
a                          NUMBER:=0;
API_ERROR                 EXCEPTION;

BEGIN

    --- PRODUCT RELATED DATA
l_pm_product_code         := 'MSPROJECT';

    --- AGREEMENT RELATED DATA
l_pm_agreement-reference  := 'amg06';
l_agreement_id            := Null;

```

```

l_customer_id           :=1004;
l_customer_name        :='Universal Packaging';
l_customer_num         :='1004';
l_agreement_num        :='amg06';
l_agreement_type       :='Service Agreement';
l_amount               := 2000;
l_term_id              := 4;
l_term_name            := Null;
l_revenue_limit_flag   := N;
l_expiration_date      := Null;
l_description          := Null;
l_owned_by_person_id   := 53;
l_owned-by_person_name :=Null;
l_attribute_category   :=Null;
l_attribute1           :=Null;
l_attribute2           :=Null;
l_attribute3           :=Null;
l_attribute4           :=Null;
l_attribute5           :=Null;
l_attribute6           :=Null;
l_attribute7           :=Null;
l_attribute8           :=Null;
l_attribute9           :=Null;
l_attribute10          :=Null;
l_template_flag        :=N;

```

..---FUNDING RELATED DATA

```

a:= 1
l_funding_rec.pm_funding_reference := 'amg06fun';
l_funding_rec.project_funding_id   = Null;
l_funding_rec.agreement_id         := Null;
l_funding_rec.project_id           := 15353;
l_funding_rec.task_id              := Null;
l_funding_rec.allocated_amount     := 1000;
l_funding_rec.date_allocated       := '01-JAN-2000';
l_funding_rec.attribute_category   := Null;
l_funding_rec.attribute1           := Null;
l_funding_rec.attribute2           := Null;
l_funding_rec.attribute3           := Null;
l_funding_rec.attribute4           := Null;
l_funding_rec.attribute5           := Null;
l_funding_rec.attribute6           := Null;
l_funding_rec.attribute7           := Null;
l_funding_rec.attribute8           := Null;

```

```

l_funding_rec.attribute9           := Null;
l_funding_rec.attribute10          := Null;

-- LOOP CONSTRUCT
l_funding_in(a) := l_funding_rec;

    a:= 2;
l_funding_rec.pm_funding_reference := 'C1004';
l_funding_rec.project_funding_id   :=Null;
l_funding_rec.agreement_id         := Null;
l_funding_rec.project_id           := 1404;
l_funding_rec.task_id              := Null;
l_funding_rec.allocated_amount     := 1000;
l_funding_rec.date_allocated       := '01-JAN-2000';
l_funding_rec.attribute_category   := Null;
l_funding_rec.attribute1           := Null;
l_funding_rec.attribute2           := Null;
l_funding_rec.attribute3           := Null;
l_funding_rec.attribute4           := Null;
l_funding_rec.attribute5           := Null;
l_funding_rec.attribute6           := Null;
l_funding_rec.attribute7           := Null;
l_funding_rec.attribute8           := Null;
l_funding_rec.attribute9           := Null;
l_funding_rec.attribute10          := Null;

-- LOOP CONSTRUCT
l_funding_in(a) := l_funding_rec;
-----
--INIT_CREATE_AGREEMENT
pa_agreement_pub.init_agreement;
-----
--LOAD AGREEMENT

pa_agreement_pub.load_agreement
    (p_api_version_number           => l_api_version_number
    ,p_init_msg_list                 => l_init_msg_list
    ,p_return_status                 => l_return_status
    ,p_pm_agreement_reference       => l_pm_agreement_reference
    ,p_agreement_id                 => l_agreement_id
    ,p_customer_id                  => l_customer_id
    ,p_customer_name                 => l_customer_name
    ,p_customer_num                  => l_customer_num

```

```

    ,p_agreement_num          => l_agreement_num
    ,p_agreement_type        => l_agreement_type
    ,p_amount                 => l_amount
    ,p_term_id                => l_term_id
,p_term_name                 => l_term_name
,p_revenue_limit_flag       => l_revenue_limit_flag
,p_expiration_date          => l_expiration_date
,p_description               => l_description
,p_owned_by_person_id       => l_owned_by_person_id
,p_owned_by_person_name     => l_owned_by_person_name
,p_attribute_category        => l_attribute_category
,p_attributel               => l_attributel
,p_attribute2               => l_attribute2
,p_attribute3               => l_attribute3
,p_attribute4               => l_attribute4
,p_attribute5               => l_attribute5
,p_attribute6               => l_attribute6
,p_attribute7               => l_attribute7
,p_attribute8               => l_attribute8
,p_attribute9               => l_attribute9
,p_attributel0              => l_attributel0
,p_template_flag            => l_template_flag);

```

```

IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;

```

```

-- LOAD_FUNDING (loop for multiple Fundings )

```

```

FOR i IN 1..a LOOP
pa_agreement_pub.load_funding
    (p_api_version_number      => l_api_version_number
    ,p_init_msg_list           => l_init_msg_list
    ,p_return_status           => l_return_status
    ,p_pm_funding_reference    => l_funding_in(i).pm_funding_reference
    ,p_funding_id              => l_funding_in(i).project_funding_id
    ,p_agreement_id            => l_funding_in(i).agreement_id
    ,p_project_id              => l_funding_in(i).project_id
    ,p_task_id                 => l_funding_in(i).task_id
    ,p_allocated_amount        => l_funding_in(i).allocated_amount
    ,p_date_allocated          => l_funding_in(i).date_allocated
    ,p_attribute_category      => l_funding_in(i).attribute_category
    ,p_attributel              => l_funding_in(i).attributel

```

```

,p_attribute2          => l_funding_in(i).attribute2
,p_attribute3          => l_funding_in(i).attribute3
,p_attribute4          => l_funding_in(i).attribute4
,p_attribute5          => l_funding_in(i).attribute5
,p_attribute6          => l_funding_in(i).attribute6
,p_attribute7          => l_funding_in(i).attribute7
,p_attribute8          => l_funding_in(i).attribute8
,p_attribute9          => l_funding_in(i).attribute9
,p_attribute10         => l_funding_in(i).attribute10);

IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;

--EXECUTE_CREATE_AGREEMENT

pa_agreement_pub.execute_create_agreement

( p_api_version_number      => l_api_version_number,
  p_commit                  => l_commit,
  p_init_msg_list           => l_init_msg_list,
  p_msg_count               => l_msg_count,
  p_msg_data                => l_msg_data
  p_return_status           => l_return_status,
  p_pm_product_code         => l_pm_product_code,
  p_agreement_id_out        => l_agreement_id_out,
  p_customer_id_out         => l_customer_id_out);
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;

--FETCH_TASK

FOR l_funding_index in 1 ..a (loop for multiple Fundings)
LOOP
pa_agreement_pub.fetch_funding
    (p_api_version_number      => l_api_version_number
    ,p_init_msg_list           => l_init_msg_list
    ,p_return_status           => l_return_status
    ,p_funding_index          => l_funding_index
    ,p_funding_id             => l_funding_id

```

```

        ,p_pm_funding_reference      => l_pm_funding_reference);
IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
END LOOP;

-----

CLEAR_CREATE_AGREEMENT
pa_agreement_pub.clear_agreement;
-----

IF l_return_status != 'S'
THEN
    RAISE API_ERROR;
END IF;
-- HANDLE EXCEPTIONS
EXCEPTION
WHEN API_ERROR THEN
    for i in 1..l_msg_count
    loop
        pa_interface_utils_pub.get_messages
            (p_msg_data      => l_msg_data ,
             p_data          => l_data ,
             p_msg_count    => l_msg_count ,
             p_msg_index_out => l_msg_index_out);

        dbms_output.put_line ('error mesg ' || l_data);

    end loop;

WHEN OTHERS THEN
    for i in 1..l_msg_count
    loop
        pa_interface_utils_pub.get_messages
            (p_msg_data      => l_msg_data ,
             p_data          => l_data ,
             p_msg_count    => l_msg_count ,
             p_msg_index_out => l_msg_index_out);

        dbms_output.put_line ('error mesg ' || l_data),

    end loop;
END ;

```

APPENDIX

E

Creating an Agreement Using a Composite Datatype API

This appendix provides sample PL/SQL code for creating an agreement using an AMG API that uses composite datatypes.

Creating an Agreement Using a Composite Datatype API

The following sample PL/SQL code is a script that creates an agreement using the PA_AGREEMENT_PUB.CREATE_AGREEMENT, which uses composite datatypes.

If you create budgets using tools or products that do not support composite datatypes, see *Creating an Agreement Using the Load-Execute-Fetch APIs: page D - 2*.

```
DECLARE
  --variables needed for API standard parameters
  l_api_version_number    NUMBER :=1.0;
  l_commit                VARCHAR2(1) := 'F';
  l_return_status         VARCHAR2(1);
  l_init_msg_list        VARCHAR2(1);
  l_msg_count             NUMBER;
  l_msg_data              VARCHAR2(2000);
  l_data                  VARCHAR2(2000);
  l_msg_entity            VARCHAR2(100);
  l_msg_entity_index      NUMBER;
  l_msg_index             NUMBER;
  l_msg_index_out         NUMBER;
  l_encoded               VARCHAR2(1);
  l_agreement_id_out      NUMBER;
  l_customer_id-out       NUMBER;
  l_funding_id            NUMBER;

  --variables needed for Oracle Agreement specific parameters
  l_pm_product_code       VARCHAR2(25);
  p_agreement_in_rec      pa_agreement_pub.Agreement_Rec._In_type
  p_agreement_out_rec     pa_agreement_pub.Agreement_Rec_Out_type

  --variables needed for funding specific parameters
  l_funding_type          pa_agreement_pub.funding_rec_in_type;
  l_agreement_in_rec      pa_agreement_pub.funding_in_tbl_type;
  l_funding_out_tbl       pa_agreement_pub.funding_out_tbl_type;

  --Funding Variables
  l_pm_funding_reference  VARCHAR2(25);
  l_funding_rec           pa_agreement_pub.funding_rec_in_type;
  l_funding_in            pa_agreement_pub.funding_rec_in_type;
  l_funding_out           pa_agreement_pub.funding_rec_out_type;

  -- Loop Variables;
```

```

a NUMBER
API_ERROR          EXCEPTION

--BEGIN

-- PRODUCT RELATED DATA

l_pm_product_code:= 'MSPROJECT' ;

--AGREEMENT DATA

p_agreement_in_rec.pm_agreement_reference := 'AMGTEST1';
p_agreement_in_rec.agreement_id          := Null;
p_agreement_in_rec.customer_id           := 21491;
p_agreement_in_rec.customer_num          := '1086';
p_agreement_in_rec.agreement_num         := 'AMGTEST1';
p_agreement_in_rec.agreement_type        := 'Contract';
p_agreement_in_rec.amount                 := 2000;
p_agreement_in_rec.term_id                := 1000;
p_agreement_in_rec.term_name              := Null;
p_agreement_in_rec.revenue_limit_flag     := 'N';
p_agreement_in_rec.expiration_date        := Null;
p_agreement_in_rec.description            := Null;
p_agreement_in_rec.owned_by_person_id     := 1234;
p_agreement_in_rec.attribute_category     := Null;
p_agreement_in_rec.attribute1             := Null;
p_agreement_in_rec.attribute3             := Null;
p_agreement_in_rec.attribute4             := Null;
p_agreement_in_rec.attribute5             := Null;
p_agreement_in_rec.attribute6             := Null;
p_agreement_in_rec.attribute7             := Null;
p_agreement_in_rec.attribute8             := Null;
p_agreement_in_rec.attribute9             := Null;
p_agreement_in_rec.attribute10            := Null;
p_agreement_in_rec.template_flag         := 'N';

--FUNDING DATA

a:= 1;
l_funding_rec.pm_funding_reference       := 'AMGTEST1FUN'
l_funding_rec.project_funding_id         := Null;
l_funding_rec.agreement_id                := Null;
l_funding_rec.project_id                  := 7946;
l_funding_rec.task_id                     := 10273;
l_funding_rec.allocated_amount            := 200;

```

```

l_funding_rec.date_allocated      := '27-DEC-01';
l_funding_rec.desc_flex_name      := Null;
l_funding_rec.attribute_category  := Null;
l_funding_rec.attribute1         := Null;
l_funding_rec.attribute2         := Null;
l_funding_rec.attribute3         := Null;
l_funding_rec.attribute4         := Null;
l_funding_rec.attribute5         := Null;
l_funding_rec.attribute6         := Null;
l_funding_rec.attribute7         := Null;
l_funding_rec.attribute8         := Null;
l_funding_rec.attribute9         := Null;
l_funding_rec.attribute10        := Null;

-- LOOP CONSTRUCT
  l_funding_in(a) := l_funding_rec;

-- CONSTRUCTING THE FUNDING TABLE
FOR i IN 1..a LOOP
l_funding_in(i).pm_funding_reference := l_funding_rec.pm_funding_reference
l_funding_in(i).project_funding_id   :=l_funding_rec.funding_id;
l_funding_in(i).agreement_id         :=l_funding_rec.p_agreement_id;
l_funding_in(i).project_id           :=l_funding_rec.p_project_id;
l_funding_in(i).task_id              := l_funding_rec.p_task_id;
l_funding_in(i).allocated_amount     :=l_funding_rec.p_allocated_amount;
l_funding_in(i).date_allocated       :=l_funding_rec.p_date_allocated;
l_funding_in(i).desc_flex_name       :=l_funding_rec.p_desc_flex_name;
l_funding_in(i).attribute_category   :=l_funding_rec.p_attribute_category;
l_funding_in(i).attribute1           :=l_funding_rec.p_attribute1;
l_funding_in(i).attribute2           :=l_funding_rec.p_attribute2;
l_funding_in(i).attribute3           :=l_funding_rec.p_attribute3;
l_funding_in(i).attribute4           :=l_funding_rec.p_attribute4;
l_funding_in(i).attribute5           :=l_funding_rec.p_attribute5;
l_funding_in(i).attribute6           :=l_funding_rec.p_attribute6;
l_funding_in(i).attribute7           :=l_funding_rec.p_attribute7;
l_funding_in(i).attribute8           :=l_funding_rec.p_attribute8;
l_funding_in(i).attribute9           :=l_funding_rec.p_attribute9;
l_funding_in(i).attribute10          :=l_funding_rec.p_attribute10;
END LOOP;

-- 'CREATE_AGREEMENT

pa_agreement_pub.create_agreement
  ( p_api_version_number => l_api_version_number

```

```

,p_commit          => l_commit
,p_init_msg_list   => l_init_msg_list
,p_msg_count       => l_msg_count
,p_msg_data        => l_msg_data
,p_return_status   => l_return_status
,p_pm_product_code => l_pm_product_code
,p_agreement_in_rec => p_agreement_in_rec
,p_agreement_out_rec=> p_agreement_out_rec
,p_funding_in_tbl  => l_funding_in
,p_funding_out_tbl => l_funding_out);
IF l_return_status != 'S'
THEN
RAISE API_ERROR;
END IF;

--HANDLE EXCEPTIONS
EXCEPTION
WHEN API_ERROR THEN
for i in 1..l_msg_count
loop
    pa_interface_utils_pub.get_messages
        (p_msg_date          => l_msg_date
        ,p_data              => l_data
        ,p_msg_count         => l_msg_count
        ,p_msg_index_out     => l_msg_index_out)

        dbms_output.put_line ('error mesg' l_data)

    end loop;
if i = 1 THEN

WHEN OTHERS THEN
pa_interface_utils_pub.get_messages
        (p_msg_data          => l_msg_data
        ,p_data              => l_data
        ,p_msg_count         => l_msg_count
        ,p_msg_index_out     => l_msg_index_out);

        dbms_output.put_line ('error mesg' l_data)
END;
/

```


APPENDIX

F

Controlling Actions in Oracle Projects

This appendix lists the actions you can prevent Oracle Projects users from performing on projects that originated in external systems.

Controlling Actions in Oracle Projects

To ensure that information in your external systems remains consistent with information in Oracle Projects, you can restrict the changes users can make to data that originates in external systems. Use the Oracle Projects Control Actions window to select the actions that you want to restrict. You can restrict these actions:

- Add Task
- Baseline Budget
- Delete Project
- Delete Task
- Update Budget
- Update Project Dates
- Update Project Description
- Update Project Name
- Update Project Number
- Update Project Organization
- Update Project Status
- Update Task Dates
- Update Task Description
- Update Task Name
- Update Task Number
- Update Task Organization

You can base the restrictions on the external system in which the information originates or on the budget type (for budget-related actions).

For example, suppose you download a project from an external system. You have a business rule that the source system always maintains project and task dates. As an additional precaution, you want to prevent users from deleting from Oracle Projects any projects and tasks that originate in an external system. To fulfill these criteria, use the Control Actions window to specify the following actions:

- Delete Project
- Delete Task
- Update Project Dates

- Update Task Dates

After you specify these actions in the Control Actions window, Oracle Projects users who try to change the project and task dates on a project that originated in an external system sees the following error message:

The value for this field originated in an external system. You cannot change it.

A user who tries to delete the project or one of its tasks sees the following message:

The record originated in an external system. You cannot delete it.

Note: You can specify effective dates for the controls you select in the Control Actions window.

APPENDIX

G

Project and Task Date Client Extension API

This appendix describes the project and task date client extension API. You can customize the API to substitute dates used by external systems for the standard Oracle Projects project and task start and completion dates.

Project and Task Date Client Extension API

Oracle Projects supports the following project tracking dates through the AMG APIs. When you download a project from an external system, you can pass the values for these dates and store them in Oracle Projects as the project and task start and completion dates.

- Actual start date
- Actual finish date
- Early start date
- Early finish date
- Late start date
- Late finish date
- Scheduled start date
- Scheduled finish date

Oracle Projects provides a template package and procedure that contains the procedure you modify to implement the project date client extension API. The package is **pa_client_extn_pm** and the procedure is **customize_dates**. The template package contains default logic to return the date information that was passed to the API without substituting it for the Oracle Projects project or task start or completion date.

The Project Date Client Extension Package Body Template (PAPMGEB.pls) contains the procedure that you modify to implement the project date client extension API. Print and review the template before you begin to modify the project date client extension API. The template is in the SPA_TOP/admin/sql directory. Do not add or change any parameter names or datatypes in the API.

Package.Procedure

pa_client_extn_pm.customize_dates

The following table lists the parameters provided by Oracle Projects for the project date client extension API.

Name	Usage	Type	Req?	Description
P_PM_PROJECT_REFERENCE	IN	VARCHAR2	No	The reference code that uniquely identifies the project in the external system
P_PM_TASK_REFERENCE	IN	VARCHAR2	No	The reference code that uniquely identifies the task in the external system

Name	Usage	Type	Req?	Description
P_PROJECT_ID	IN	NUMBER	No	The reference code that uniquely identifies the project in Oracle Projects
P_TASK_ID	IN	NUMBER	No	The reference code that uniquely identifies the task in Oracle Projects
P_PM_PRODUCT_CODE	IN	VARCHAR2	Yes	The product code of the external system
P_IN_START_DATE	IN	DATE	Yes	The default start date
P_IN_COMPLETION_DATE	IN	DATE	Yes	The default completion date
P_ACTUAL_START_DATE	IN	DATE	No	The actual start date
P_ACTUAL_FINISH_DATE	IN	DATE	No	The actual finish date
P_EARLY_START_DATE	IN	DATE	No	The early start date
P_EARLY_FINISH_DATE	IN	DATE	No	The early finish date
P_LATE_START_DATE	IN	DATE	No	The late start date
P_LATE_FINISH_DATE	IN	DATE	No	The late finish date
P_SCHEDULED_START_DATE	IN	DATE	No	The scheduled start date
P_SCHEDULED_FINISH_DATE	IN	DATE	No	The scheduled finish date
P_OUT_START_DATE	OUT	DATE		
P_OUT_COMPLETION_DATE	OUT	DATE		
P_ERROR_CODE	OUT	NUMBER		
P_ERROR_MESSAGE	OUT	VARCHAR2		

You can customize this API to substitute a different set of project and task start dates for the standard Oracle Projects project and task start and completion dates. For example, you can define your own rules to determine which project and task dates in the external system correspond to the project and task start and completion dates in Oracle Projects.

The following code shows how to map the actual start and actual finish dates in an external system to the project and task start and completion dates in Oracle Projects.

Note: The parameters P_OUT_START_DATE and P_OUT_COMPLETION_DATE must return valid values. The public APIs read the values and will not execute properly if the date values are invalid.

```
-- Initialize the out variables
p_error_code := 0;
p_error_stage := NULL;
IF p_actual_start_date IS NOT NULL and p_actual_finish_date
  IS NOT NULL THEN
```

```

        p_out_start_date := p_actual_start_date;
        p_out_finish_date := p_actual_finish_date;
ELSE
        p_out_start_date := p_in_start_date;
        p_out_completion_date := p_in_completion_date;
END IF;
-- To specify conditions based on different external products that you
-- are importing from, use code that looks something like this
IF p_pm_product_code = <Your product code> THEN
    IF p_actual_start_date IS NOT NULL and p_actual_finish_date IS NOT NULL THEN
        p_out_start_date := p_actual_start_date;
        p_out_finish_date := p_actual_finish_date;
    ELSE
        p_out_start_date := p_in_start_date;
        p_out_completion_date := p_in_completion_date;
    END IF;
ELSIF p_pm_product_code = <differenct product code>
    IF p_early_start_date IS NOT NULL and p_early_finish_date IS NOT NULL THEN
        p_out_start_date := p_early_start_date;
        p_out_finish_date := p_early_finish_date;
    ELSE
        p_out_start_date := p_in_start_date;
        p_out_completion_date := p_in_completion_date;
    END IF;
ELSE
    p_out_start_date := p_in_start_date;
    p_out_completion_date := p_in_completion_date;
END IF;
-- If you want different mappings for projects and tasks then base your logic on
-- p_pm_task_reference or p_task_id
IF (p_pm_task_reference IS NOT NULL or p_task_id IS NOT NULL) THEN
    -- (this means this is for a task)
    -- place the logic for assigning one set of dates here
ELSE -- ( this means this is for a project)
    -- place the logic for assigning a different set of dates
END IF;
EXCEPTION
    WHEN OTHERS THEN
        p_error_code := -1;
        -- If ORACLE error then set p_error_code to SQLCODE
        -- Handle your exception here

```

APPENDIX

H

Alphabetical List of APIs

This appendix lists the AMG APIs and views in alphabetical order with a page reference to a complete description of each API and view.

Alphabetical List of APIs

The following table contains an alphabetical list of APIs. For a complete description of an API, refer to the indicated page.

<i>API Name</i>	<i>Description</i>
ADD_BUDGET_LINE: page 5 – 9	PL/SQL procedure. Adds a budget line to a working budget.
ADD_FUNDING: page 8 – 9	PL/SQL procedure. Adds a fund to an agreement.
ADD_RESOURCE_LIST_MEMBER: page 6 – 15	PL/SQL procedure. Adds a resource member to an existing resource list.
ADD_TASK: page 4 – 22	PL/SQL procedure. Adds new subtask to a task in Oracle Projects.
BASELINE_BUDGET: page 5 – 12	PL/SQL procedure. Baselines (approves) an existing budget.
CALCULATE_AMOUNTS: page 5 – 13	Calls the PA_CLIENT_EXTN_BUDGET extension to recalculate specified cost and revenue amounts and update the budget lines with the results.
CHECK_ADD_FUNDING_OK: page 8 – 20	PL/SQL procedure. Checks whether a fund can be added.
CHECK_ADD_SUBTASK_OK: page 4 – 61	Check procedure. Identifies conflicts with certain business rules.
CHECK_CHANGE_PARENT_OK: page 4 – 62	Check procedure. Shows whether a task can be moved from one parent to another.
CHECK_CHANGE_PROJECT_ORG_OK: page 4 – 63	Check procedure. Shows whether a project organization can be changed.
CHECK_DELETE_AGREEMENT_OK: page 8 – 19	PL/SQL procedure. Checks whether an agreement can be deleted.
CHECK_DELETE_FUNDING_OK: page 8 – 21	PL/SQL procedure. Checks whether a fund can be deleted.
CHECK_DELETE_PROJECT_OK: page 4 – 63	Check procedure. Shows whether a project can be deleted.
CHECK_DELETE_TASK_OK: page 4 – 64	Check procedure. Shows whether a task can be deleted.
CHECK_TASK_NUMBER_CHANGE_OK: page 4 – 65	Check procedure. Shows whether a task number can be changed.

<i>API Name</i>	<i>Description</i>
CHECK_UNIQUE_PROJECT_REFERENCE: page 4 – 65	Check procedure. Shows whether a project reference number assigned by an external system is unique in Oracle Projects.
CHECK_UNIQUE_TASK_NUMBER: page 4 – 66	Check procedure. Shows whether a task number in a project assigned by an external system is unique in Oracle Projects..
CHECK_UNIQUE_TASK_REFERENCE: page 4 – 66	Check procedure. Shows whether a task reference number is unique in Oracle Projects.
CHECK_UPDATE_FUNDING_OK: page 8 – 22	PL/SQL procedure. Checks whether a fund can be updated.
CLEAR_AGREEMENT: page 8 – 19	PL/SQL procedure. Clears the globals that were set up during initialization.
CLEAR_BUDGET: page 5 – 28	Load–Execute–Fetch procedure. Clears the global data structures set up during Initialize process.
CLEAR_CREATE_RESOURCE_LIST: page 6 – 23	Load–Execute–Fetch procedure. Clears the global data structures set up during Initialize process.
CLEAR_PROJECT: page 4 – 51	Load–Execute–Fetch procedure. Clears the global data structures set up during Load process.
CLEAR_UPDATE_MEMBERS: page 6 – 23	Load–Execute–Fetch procedure. Clears the global data structures set up during Initialize process.
CREATE_AGREEMENT: page 8 – 6	PL/SQL procedure. Creates an agreement with associated funds.
CREATE_DRAFT_BUDGET: page 5 – 14	PL/SQL procedure. Creates a draft budget for a given project, budget type, and budget entry method.
CREATE_PROJECT: page 4 – 26	PL/SQL procedure. Creates a project and associated tasks. Requires specification of a base project or template.
CREATE_RESOURCE_LIST: page 6 – 16	PL/SQL procedure. Creates a resource list and (optional) resource list members.
DELETE_AGREEMENT: page 8 – 7	PL/SQL procedure. Deletes an agreement and associated funds.

<i>API Name</i>	<i>Description</i>
DELETE_BUDGET_LINE: page 5 – 19	PL/SQL procedure. Deletes a budget line from a working budget for a given project and budget type.
DEETE_DRAFT_BUDGET: page 5 – 20	PL/SQL procedure. Deletes a working budget as well as associated budget lines and resource assignments.
DELETE_FUNDING: page 8 – 11	PL/SQL procedure. Deletes a fund from an agreement.
DELETE_PROJECT: page 4 – 37	PL/SQL procedure. Deletes a project and its associated tasks.
DELETE_RESOURCE_LIST: page 6 – 18	PL/SQL procedure. Deletes the specified resource list.
DELETE_RESOURCE_LIST_MEMBER: page 6 – 19	PL/SQL procedure. Deletes the specified member of a resource list.
DELETE_TASK: page 4 – 38	PL/SQL procedure. Deletes the specified task.
EXECUTE_CALCULATE_AMOUNTS: page 5 – 28	Load–Execute–Fetch procedure. Calculates raw cost, burdened cost, and revenue and updates the budget lines with the results.
EXECUTE_CREATE_AGREEMENT: page 5 – 30	PL/SQL procedure. Creates an agreement with funding using the data stored in the global tables during the Load phase.
EXECUTE_CREATE_DRAFT_BUDGET: page 4 – 51	Load–Execute–Fetch procedure. Creates a budget and its associated lines using data stored in the global tables during the Load process.
EXECUTE_CREATE_PROJECT: page 4 – 51	Load–Execute–Fetch procedure. Creates a project and tasks using data stored in the global tables during the Load process.
EXECUTE_UPDATE_AGREEMENT: page 8 – 17	PL/SQL procedure. Updates an agreement with funding using the data stored in the global tables during the Load phase.
EXECUTE_UPDATE_BUDGET: page 5 – 31	Load–Execute–Fetch procedure. Updates budget and budget lines using data stored in the global tables during the Load process.

<i>API Name</i>	<i>Description</i>
EXECUTE_UPDATE_PROJECT: page 4 – 52	Load–Execute–Fetch procedure. Updates projects and tasks using data stored in the global tables during the Load process. This procedure can add new tasks but cannot delete tasks.
EXEC_CREATE_RESOURCE_LIST: page 6 – 23	Load–Execute–Fetch procedure. Load–Execute–Fetch procedure. Executes the composite API CREATE_RESOURCE_LIST.
EXEC_UPDATE_RESOURCE_LIST: page 6 – 24	Load–Execute–Fetch procedure. Executes the composite API UPDATE_RESOURCE_LIST.
FETCH_BUDGET_LINE: page 5 – 32	Load–Execute–Fetch procedure. Retrieves the status returned when the budget line was created.
FETCH_CALCULATE_AMOUNTS: page 5 – 32	Load–Execute–Fetch procedure. Gets the raw cost, burdened cost, and revenue amounts by budget line from the global records updated by EXECUTE_CALCULATE AMOUNTS.
FETCH_FUNDING: page 8 – 18	PL/SQL procedure. Gets the return status that was returned during the creation of funds and stored in a global PL/SQL table.
FETCH_MEMBERS: page 6 – 24	Load–Execute–Fetch procedure. Fetches resource members from the global output structure for resource members.
FETCH_RESOURCE_LIST: page 6 – 25	Load–Execute–Fetch procedure. Fetches one resource identifier at a time from the global structure.
FETCH_TASK: page 4 – 52	Load–Execute–Fetch procedure. Fetches task parameters.
GET_ACCUM_PERIOD_INFO: page 1 – 21	PL/SQL procedure. Returns information about the last period through which the project is summarized in Oracle Projects, as well as the current reporting period.
GET_DEFAULTS: page 1 – 20	PL/SQL procedure. Returns the default values required to initialize the VARCHAR2, NUMBER, and DATE variables in your programs.

<i>API Name</i>	<i>Description</i>
GET_MESSAGES: page 1 – 18	PL/SQL procedure. Retrieves messages from the message stack.
INIT_AGREEMENT: page 8 – 14	PL/SQL procedure. Sets up the global tables that will be used by the Load–Execute Fetch programs.
INIT_BUDGET: page 5 – 33	Load–Execute–Fetch procedure. Sets up the global structures used by other Load–Execute–Fetch procedures.
INIT_CALCULATE_AMOUNTS: page 5 – 34	Load–Execute–Fetch procedure. Get the raw cost, burdened cost, and revenue amounts by budget line from the global records updated by the EXECUTE_CALCULATE_AMOUNTS API.
INIT_CREATE_RESOURCE_LIST: page 6 – 25	Load–Execute–Fetch procedure. Sets up the global data structures used by other Load–Execute–Fetch procedures.
INIT_PROJECT: page 4 – 53	Load–Execute–Fetch procedure. Sets up the global tables used by the other Load–Execute–Fetch procedures to create projects.
INIT_UPDATE_MEMBERS: page 6 – 25	Load–Execute–Fetch procedure. Sets up the global data structures used by other Load–Execute–Fetch procedures.
LOAD_AGREEMENT: page 8 – 14	PL/SQL procedure. Loads an agreement to a PL/SQL record
LOAD_BUDGET_LINE: page 5 – 34	Load–Execute–Fetch procedure. Loads a budget line to a global PL/SQL table.
LOAD_CLASS_CATEGORY: page 4 – 53	Load–Execute–Fetch procedure. Loads class categories to a global PL/SQL procedure.
LOAD_FUNDING: page 8 – 15	PL/SQL procedure. Loads funding to a PL/SQL table.
LOAD_KEY_MEMBER: page 4 – 53	Load–Execute–Fetch procedure. Loads key members to a global PL/SQL table.
LOAD_MEMBERS: page 6 – 25	Load–Execute–Fetch procedure. Loads the resource list member global input structure.
LOAD_PROJECT: page 4 – 54	Load–Execute–Fetch procedure. Loads a project to a global PL/SQL record.
LOAD_RESOURCE_LIST: page 6 – 27	Load–Execute–Fetch procedure. Loads the resource list global structure.

<i>API Name</i>	<i>Description</i>
LOAD_TASK: page 4 – 58	Load–Execute–Fetch procedure. Loads a task to a global PL/SQL table.
SET_GLOBAL_INFO: page 1 – 9	PL/SQL procedure. Sets the global variables necessary to access data in an environment that uses multi–organizations.
SORT_RESOURCE_LIST_MEMBERS: page 6 – 19	PL/SQL procedure. Updates the sort order for resource members in the given resource list.
UPDATE_AGREEMENT: page 8 – 8	PL/SQL procedure. Updates an agreement and associated funds.
UPDATE_BUDGET: page 5 – 21	PL/SQL procedure. Updates a working budget and its budget lines, and can insert new budget lines to a working budget. Updates an existing line in a working budget.
UPDATE_BUDGET_LINE: page 5 – 25	PL/SQL procedure. Updates an existing line in a working budget.
UPDATE_EARNED_VALUE: page 7 – 29	PL/SQL procedure. Updates earned value information for lowest task–resource combinations.
UPDATE_FUNDING: page 8 – 12	PL/SQL procedure. Updates a fund for an agreement.
UPDATE_PROGRESS: page 7 – 30	PL/SQL procedure. Updates progress information up to a given date for all levels of the work breakdown structure.
UPDATE_PROJECT: page 4 – 40	PL/SQL procedure. Modifies existing projects and tasks.
UPDATE_RESOURCE_LIST: page 6 – 20	PL/SQL procedure. Updates an existing resource list to modify information about existing members or add new members.
UPDATE_RESOURCE_LIST_MEMBER: page 6 – 22	PL/SQL procedure. Updates the alias and enables or disables the resource list members.
UPDATE_TASK: page 4 – 45	PL/SQL procedure. Updates existing tasks.

Alphabetical List of Views

The following table contains an alphabetical list of views. For a complete description of a view, refer to the indicated page.

<i>API Name</i>	<i>Page</i>	<i>Description</i>
PA_ACCUM_CMT_TXNS_V	7 – 8	View of project level, project–task level, and project–task–resource level commitments.
PA_ACCUM_RSRC_ACT_V	7 – 9	View of current project and task level resource actual cost and revenue summary amounts. Data includes inception–to–date, year–to–date, prior period, and period–to–date amounts.
PA_ACCUM_RSRC_CMT_V	7 – 10	View of current project and task level commitment summary amounts. Data includes inception–to–date, year–to–date, prior period, and period–to–date amounts.
PA_ACCUM_RSRC_COST_BGT_V	7 – 12	View of project and task level cost budget summary amounts.
PA_ACCUM_RSRC_REV_BGT_V	7 – 14	View of project and task level resource revenue budget summary amounts.
PA_ACCUM_WBS_ACT_V	7 – 15	View of project and task level revenue budget summary amounts.
PA_ACCUM_WBS_CMT_V	7 – 17	View of current project and task level commitment summary amounts.
PA_ACCUM_WBS_COST_BGT_V	7 – 18	View of project and task level cost budget summary amounts.
PA_ACCUM_WBS_REV_BGT_V	7 – 20	View of project and task level revenue budget summary amounts.
PA_ACT_BY_GL_PERIOD_V	7 – 21	View of actual cost and revenue totals for lowest tasks and resources by GL periods.
PA_ACT_BY_PA_PERIOD_V	7 – 22	View of actual cost and revenue totals for lowest tasks and resources by PA periods.
PA_AGREEMENT_TYPE_LOV_V	8 – 4	View of valid agreement types.
PA_AMG_RESOURCE_INFO_V	6 – 4	Customizable view to retrieve additional information about resource list members.
PA_BASE_BUDGET_BY_GL_PERIOD_V	6–4	View of the most recently baselined budget data by GL period. Data includes raw cost, burdened cost, revenue, quantity, and resource.
PA_BASE_BUDGET_BY_PA_PERIOD_V	5 – 4	View of the most recent baselined budget amounts by GL period.

<i>API Name</i>	<i>Page</i>	<i>Description</i>
PA_BUDGET_CHANGE_REASON_V	5 – 6	View of valid budget change reason codes.
PA_BUDGET_ENTRY_METHODS_V	5 – 6	View of valid budget entry methods.
PA_BUDGET_STATUS_CODES_V	5 – 6	View of the valid budget status codes.
PA_BUDGET_TYPES_V	5 – 7	View of the valid budget types.
PA_BURDEN_COMPONENT_CMT_V	7 – 22	View of commitment burden components by resource, PA period name, expenditure type, expenditure organization, and burden set for each transaction summarization record.
PA_BURDEN_COMPONENT_COST_V	7 – 23	View of actual burden components by resource, PA period name, expenditure type, expenditure organization, and burden set for each transaction summarization record. Returns components only for resources that have been burdened.
PA_CLASS_CATEGORIES_LOV_V	4 – 4	View of the class category, class code, and description of valid class codes set up in Oracle Projects.
PA_CMT_BY_GL_PERIOD_V	7 – 24	View of current commitment totals for lowest tasks and resources by GL periods.
PA_CMT_BY_PA_PERIOD_V	7 – 25	View of current commitment totals for lowest tasks and resources by PA periods.
PA_CUSTOMERS_LOV_V	8 – 5	View of the name and number of valid customers defined in Oracle Projects.
PA_DISTRIBUTION_RULES_LOV_V	4 – 4	View of the valid revenue distribution rules and associated project types.
PA_EMPLOYEES_RES_V	6 – 5	View of all employees defined in human resources. Any employee displayed on this list can be defined as a resource in Oracle Projects.
PA_EVENT_TYPES_RES_V	6 – 5	View of all of the event types set up in Oracle Projects.
PA_EXPENDITURE_TYPES_RES_V	6 – 6	View of all defined expenditure types.
PA_EXPEND_CATEGORIES_RES_V	6 – 6	View of all defined expenditure categories.
PA_GL_PERIODS_V	7 – 25	View of the GL periods and their start and end dates.
PA_JOBS_RES_V	7–6	View of all jobs defined in human resources. Any job displayed on this list can be defined as a resource in Oracle Projects.

<i>API Name</i>	<i>Page</i>	<i>Description</i>
PA_KEY_MEMBERS_LOV_V	4 – 5	View of the names and employee ID numbers of all valid key members in Oracle Projects.
PA_LOWEST_LEVEL_RESOURCES	6 – 8	View of resource list names and IDs and lowest level resource names and IDs.
PA_ORGANIZATIONS_LOV_V	4 – 5	View of name and type of all valid organizations defined in Oracle Projects.
PA_ORGANIZATIONS_RES_V	6 – 9	View of all organizations defined in human resources. Any organization displayed on this list can be defined as a resource in Oracle Projects.
PA_ORIG_BUDGET_BY_GL_PERIOD_V	5 – 7	View of the original budget by GL period.
PA_ORIG_BUDGET_BY_PA_PERIOD_V	5 – 8	View of the original budget by PA period.
PA_OVERRIDE_FIELDS_V	4 – 6	View of information about the Quick Entry field prompts for a particular template. Data includes Quick Entry field name, name displayed on the quick entry window, data type, mandatory flag, and the name of the list of values view.
PA_OVERRIDE_FIELD_VALUES_V	4 – 8	View of information about the values passed to the Quick Entry fields while creating a project. Data includes Quick Entry field name, name displayed on the quick entry window, data type, mandatory flag, codes, and field values.
PA_OWNED_BY_LOV_V	8 – 5	View of valid employees.
PA_PA_PERIODS_V	7 – 25	View of the PA periods and their start and end dates.
PA_PM_REFERENCE_V	7 – 26	View of project and task information, including unique reference codes used by external systems to identify the project and task.
PA_PROJECTS_AMG_V	4 – 9	View of all valid projects for the organization associated with the user's responsibility.
PA_PROJECT_STATUS_LOV_V	4 – 8	View of the valid project statuses set up in Oracle Projects.
PA_PROJ_ORG_STRUCTURES_V	6 – 9	View of retrieves the organization hierarchy, including the names of organizations and their parents.

<i>API Name</i>	<i>Page</i>	<i>Description</i>
PA_QRY_RESOURCE_LISTS_V	6 – 10	View of the defined resource lists. Data includes list name, description, and effective date range.
PA_QUERY_RES_LIST_MEMBERS_V	6 – 10	View of all members of a resource list. Data includes list name, resource name, alias, type, and information about the member.
PA_RESOURCE_LIST_GROUPS_V	6 – 11	View of all resource groups within a resource list.
PA_RESOURCE_LIST_V	6 – 12	View of all resource lists.
PA_RESOURCE_TYPES_ACTIVE_V	6 – 12	View of all active resource types.
PA_REVENUE_CATEGORIES_RES_V	6 – 13	View of all revenue categories.
PA_SELECT_TEMPLATE_V	4 – 14	View of projects and templates created in Oracle Projects. Data includes project and template ID, name, organization, distribution rule, and active date range.
PA_SERVICE_TYPE_LOV_V	4 – 15	View of the valid service type codes defined in Oracle Projects.
PA_TASKS_AMG_V	4 – 17	View of all valid tasks for the organization associated with the user's responsibility.
PA_TASK MANAGERS_LOV_V	4 – 15	View of the valid employees from Oracle Projects. Data includes both the employee number and the unique identifier person_id.
PA_TERMS_LOV_V	8 – 4	View of customer terms.
PA_TXN_ACCUM_V	7 – 27	View of cost, revenue, hours, and quantity amounts. Data includes person, job, organization, vendor, expenditure type, event type, non-labor resource, expenditure category, revenue category, non-labor resource organization, event type classification, system linkage function, and week ending date.
PA_USER_RESP_V	2–10	View of valid user responsibilities.
PA_VENDORS_RES_V	6 – 13	View of information pertaining to all vendors defined in Oracle Purchasing.

APPENDIX

I

List of AMG Messages

This appendix lists the AMG messages in alphabetical order.

AMG Messages

The following table shows the messages used in Activity Management Gateway.

New Message Code	Length	Description	Token(s)
PA_ALL_WARN_NO_EMPL_REC_AMG	27	This user is not yet registered as an employee.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_ALL_WARN_NO_EMPL_REC_AMG	27	This user is not yet registered as an employee	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_AMT_ALLOC_LT_ACCR_AMG	27	Total amount allocated cannot be less than amount accrued or billed.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_BASE_RES_LIST_EXISTS_AMG	30	You cannot change the resource list for a baselined budget	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_CORE_NO_VERSION_ID_AMG	28	A budget does not exist for this project with specified budget type.	PROJECT_NUMBER
PA_BU_INVALID_NEW_PERIOD_AMG	28	You cannot copy a budget to a period which is out of the range of system defined periods (for example, PA period or GL period).	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_NO_BUDGET_AMG	20	There are no budget lines in this draft budget. The budget must be entered before baseline.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_NO_PROJ_END_DATE_AMG	26	Project does not have a start date or a completion date.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_NO_TASK_PROJ_DATE_AMG	27	Task does not have a start date or a completion date.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_BU_UNBAL_PROJ_BUDG_AMG	25	Project funding is not equal to the budget total. To baseline a draft budget, the budget total must be as same as funding total.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE

Table 9 – 1 AMG Messages (Page 1 of 6)

New Message Code	Length	Description	Token(s)
PA_BU_UNBAL_TASK_BUDG_AMG	25	Task funding is not equal to the budget total of the task. To baseline a draft budget, the budget total must be as same as funding total.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_COPY_PROJECT_FAILED_AMG	26	Error occurred while creating the project.	PROJECT_NUMBER
PA_CREATE_CONTACTS_FAILED_AMG	29	Error occurred while creating Customer Contact information.	PROJECT_NUMBER
PA_CUST_NOT_OVERRIDABLE_AMG	27	You cannot override the Customer field while using this template.	PROJECT_NUMBER
PA_DESC_NOT_OVERRIDABLE_AMG	27	You cannot override the Description field while using this template.	PROJECT_NUMBER
PA_GET_CUST_INFO_FAILED_AMG	27	Error occurred while getting Customer information.	PROJECT_NUMBER
PA_HAS_REV/INV_AMG	18	Distribution rule cannot be changed since cost/revenue/invoices exist[\n] [\n] Cause:[\t]You cannot change the distribution rule because the project has costed items, revenue, or invoices.	PROJECT_NUMBER
PA_INVALID_DIST_RULE_AMG	24	Distribution Rule is invalid.	PROJECT_NUMBER
PA_INVALID_ORG_AMG	18	Organization is invalid.	PROJECT_NUMBER
PA_INVALID_PT_CLASS_ORG_AMG	27	Invalid organization. You cannot use the specified organization to create projects of this project type class. Choose a different organization or add the project type class to the current organization.	PROJECT_NUMBER
PA_NO_BILL_TO_ADDRESS_AMG	25	Active primary Bill To Address does not exist for the specified customer.	PROJECT_NUMBER
PA_NO_BILL_TO_CONTACT_AMG	25	Active primary billing contact does not exist for the specified customer.	PROJECT_NUMBER
PA_NO_CLIENT_EXISTS_AMG	23	The billing allocation across project client(s) is incomplete.	PROJECT_NUMBER
PA_NO_CONTACT_EXISTS_AMG	28	Billing contact not defined for each customer.	PROJECT_NUMBER
PA_NO_MANAGER_AMG	24	Project manager not currently defined for this project.	PROJECT_NUMBER
PA_NO_ORIG_PROJ_ID_AMG	22	Original project ID is not specified.	PROJECT_NUMBER
PA_NO_PROJ_CREATED_AMG	22	New project not created. No project information in the source project.	PROJECT_NUMBER
PA_NO_PROJ_ID_AMG	17	Project ID not specified.	PROJECT_NUMBER
PA_NO_REQ_CATEGORY_EXISTS_AMG	29	All mandatory class categories have not been classified.	PROJECT_NUMBER
PA_NO_SHIP_TO_ADDRESS_AMG	25	Active primary Ship To Address does not exist for the specified customer.	PROJECT_NUMBER

Table 9 – 1 AMG Messages (Page 2 of 6)

New Message Code	Length	Description	Token(s)
PA_NO_TASK_COPIED_AMG	21	No task is copied since there are tasks in the source project.	PROJECT_NUMBER
PA_NO_TASK_ID_D_AMG	19	You cannot delete this task since no task information has been provided.	PROJECT_NUMBER, TASK_NUMBER
PA_NO_TASK_ID_ST_AMG	20	You cannot create a subtask below this task since task information was not specified.	PROJECT_NUMBER, TASK_NUMBER
PA_NO_TOP_TASK_ID_ST_AMG	25	You cannot create a subtask below this task since task does not have top task ID.	PROJECT_NUMBER, TASK_NUMBER
PA_NO_UNIQUE_ID_AMG	19	Failed to generate unique project number. Action: Please contact your System Administrator to set up the Next Number field for Automatic Project Numbering in Implementation Options Window.	PROJECT_NUMBER
PA_PRODUCT_CODE_IS_MISSING_AMG	30	External product code required.	General
PA_PROJECT_NAME_IS_MISSING_AMG	30	Project name required.	PROJECT_NUMBER
PA_PROJECT_REF_IS_MISSING_AMG	29	External project reference required.	PROJECT_NUMBER
PA_PROJECT_STATUS_INVALID_AMG	29	The project status is invalid.	PROJECT_NUMBER
PA_PROJ_AP_INV_EXIST_D_AMG	26	You cannot delete this project since supplier invoices exist	PROJECT_NUMBER
PA_PROJ_BUDGET_EXIST_D_AMG	26	You cannot delete this project since budgets exist	PROJECT_NUMBER
PA_PROJ_BURDEN_SUM_DEST_D_AMG	29	The project is being used for the purpose of accumulating burden costs on project types.	PROJECT_NUMBER
PA_PROJ_CMT_TXN_EXIST_D_AMG	27	You cannot delete this project since project commitment transactions exist.	PROJECT_NUMBER
PA_PROJ_EVENT_EXIST_D_AMG	25	You cannot delete this project since events exist	PROJECT_NUMBER
PA_PROJ_EXP_ITEM_EXIST_D_AMG	28	You cannot delete this project since expenditure items exist.	PROJECT_NUMBER
PA_PROJ_FUND_EXIST_D_AMG	25	You cannot delete this project since funding exists.	PROJECT_NUMBER
PA_PROJ_INV_DIST_EXIST_D_AMG	28	You cannot delete this project since supplier invoice distribution lines exist	PROJECT_NUMBER
PA_PROJ_IN_USE_EXTERNAL_D_AMG	29	You cannot delete this project since project references exist	PROJECT_NUMBER
PA_PROJ_ORG_NOT_ACTIVE_AMG	26	This project organization is not active or is not within the current Project/Task owning organization hierarchy.	PROJECT_NUMBER
PA_PROJ_PO_DIST_EXIST_D_AMG	27	You cannot delete this project since purchase order distributions exist	PROJECT_NUMBER
PA_PR_COM_RUL_SET_EXIST_D_AMG	29	You cannot delete this project since compensation rules exist	PROJECT_NUMBER

Table 9 – 1 AMG Messages (Page 3 of 6)

New Message Code	Length	Description	Token(s)
PA_PR_CREATED_REF_EXIST_D_AMG	29	You cannot delete this project since compensation rule sets exist	PROJECT_NUMBER
PA_PR_INSUF_BILL_CONTACT_AMG	28	Billing contact not defined for each customer.	PROJECT_NUMBER
PA_PR_INSUF_CLASS_CODES_AMG	27	You must specify all mandatory class categories.	PROJECT_NUMBER
PA_PR_INSUF_PROJ_MGR_AMG	24	Project manager not currently defined for this project.	PROJECT_NUMBER
PA_PR_INVALID_START_DATE_AMG	28	Project start date must be earlier than all task start dates.	PROJECT_NUMBER
PA_PR_NAME_NOT_UNIQUE_AMG	27	Project name must be unique across all operating units in the Oracle Applications installation.	PROJECT_NUMBER
PA_PR_NAME_NOT_UNIQUE_A_AMG	25	Project name must be unique across all operating units in the Oracle Applications installation.	PROJECT_NUMBER
PA_PR_NO_PROJ_NAME_AMG	22	Project name not specified.	PROJECT_NUMBER
PA_PR_NO_PROJ_NUM_AMG	21	Project number ID not specified.	PROJECT_NUMBER
PA_PR_NO_UPD_SEGMENT1_EXP_AMG	29	You cannot change the project number since expenditure items exist	PROJECT_NUMBER
PA_PR_NUMERIC_NUM_REG_AMG	25	Please enter a numeric project number.	PROJECT_NUMBER
PA_PR_NUMERIC_NUM_REQ_AMG	25	Your implementation requires a numeric project number.	PROJECT_NUMBER
PA_PR_NUM_NOT_UNIQUE_AMG	26	Project number must be unique across all operating units in the Oracle Applications installation.	PROJECT_NUMBER
PA_PR_NUM_NOT_UNIQUE_A_AMG	24	Project number must be unique across all operating units in the Oracle Applications installation.	PROJECT_NUMBER
PA_PR_PO_REQ_DIST_EXIST_D_AMG	29	You cannot delete this project since purchase order requisitions exist.	PROJECT_NUMBER
PA_PR_START_DATE_NEEDED_AMG	27	The start date of the project is required if the completion date of the project is specified.	PROJECT_NUMBER
PA_PR_START_DATE_NEEDED_AMG	23	The start date of the project is required if the completion date of the project is specified.	PROJECT_NUMBER
PA_PUBLIC_SECTOR_INVALID_AMG	28	Invalid value for Public Sector flag.	PROJECT_NUMBER
PA_RE_ASSGMT_NOT_FOUND_AMG	26	Resource list assignment not found.	PROJECT_NUMBER
PA_RE_PROJ_NOT_FOUND_AMG	24	Specified project is invalid.	PROJECT_NUMBER
PA_RE_RL_INACTIVE_AMG	21	Resource list is not active.	PROJECT_NUMBER
PA_RE_RL_NOT_FOUND_AMG	22	Specified resource list is invalid.	PROJECT_NUMBER
PA_RE_USE_CODE_NOT_FOUND_AMG	28	Use code not found.	PROJECT_NUMBER

Table 9 – 1 AMG Messages (Page 4 of 6)

New Message Code	Length	Description	Token(s)
PA_SOURCE_TEMPLATE_INVALID_AMG	30	Source template ID is invalid.	PROJECT_NUMBER
PA_SOURCE_TEMP_IS_MISSING_AMG	30	Source template ID is required.	PROJECT_NUMBER
PA_SU_INVALID_DATES_AMG	23	From Date must be on or before the To Date.	PROJECT_NUMBER
PA_TASK_BURDEN_SUM_DEST_ST_AMG	30	The task is being used for the purpose of accumulating burden costs on project types.	PROJECT_NUMBER, TASK_NUMBER
PA_TASK_BURDEN_SUM_DEST_ST_AMG	29	The task is being used for the purpose of accumulating burden costs on project types.	PROJECT_NUMBER, TASK_NUMBER
PA_TASK_FUND_NO_PROJ_EVT_AMG	28	Task funding with project level events is not allowed.	PROJECT_NUMBER, TASK_NUMBER, BUDGET_TYPE, RESOURCE_NAME, START_DATE
PA_TASK_IN_USE_EXTERNAL_D_AMG	26	You cannot delete this task since task references exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_AP_INV_DIST_EXIST_D_AMG	30	You cannot delete this task since invoice distribution lines exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_AP_INV_DIST_EXIST_ST_AMG	30	You cannot create a subtask below this task since supplier invoice distribution lines exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_AP_INV_EXIST_D_AMG	25	You cannot delete this task since supplier invoices exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_AP_INV_EXIST_ST_AMG	26	You cannot create a subtask below this task since supplier invoices exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_ASSETASSIG_EXIST_ST_AMG	30	You cannot create a subtask below this task since assets have been assigned.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_BUDGET_EXIST_D_AMG	25	You cannot delete this task since budgets exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_BUDGET_EXIST_ST_AMG	26	You cannot create a subtask below this task since budgets exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_BUR_SCHOVR_EXIST_ST_AMG	30	You cannot create a subtask below this task since burden schedule overrides exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_CMT_TXN_EXIST_D_AMG	26	You cannot delete this task since commitment transactions exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_EBILL_RATE_EXIST_ST_AMG	30	You cannot create a subtask below this task since employee billing rate overrides exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_EVENT_EXIST_D_AMG	24	You cannot delete this task since events exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_EXP_ITEM_EXIST_D_AMG	27	You cannot delete this task since expenditure items exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_EXP_ITEM_EXIST_ST_AMG	28	You cannot create a subtask below this task since expenditure items exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_FUND_EXIST_D_AMG	27	You cannot delete this task since supplier invoice distribution line exist.	PROJECT_NUMBER, TASK_NUMBER

Table 9 – 1 AMG Messages (Page 5 of 6)

New Message Code	Length	Description	Token(s)
PA_TSK_JBILLTITLE_EXIST_ST_AMG	30	You cannot create a subtask below this task since job billing title overrides exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_JBILL_RATE_EXIST_ST_AMG	30	You cannot create a subtask below this task since job bill rate overrides exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_LAB_MULT_EXIST_ST_AMG	26	You cannot create a subtask below this task since there is labor multiplier for this task.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_L_COST_MUL_EXIST_ST_AMG	30	You cannot create a subtask below this task since labor cost multipliers exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_NL_BIL_RAT_EXIST_ST_AMG	30	You cannot create a subtask below this task since non-labor bill rate overrides exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_PO_DIST_EXIST_D_AMG	26	You cannot delete this task since supplier invoice distribution lines exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_PO_DIST_EXIST_ST_AMG	27	You cannot create a subtask below this task since supplier invoice distribution lines exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_PO_REQDIST_EXIST_ST_AMG	30	You cannot create a subtask below this task since purchase order requisitions exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_RULE_SET_EXIST_D_AMG	27	You cannot delete this task since compensation rule sets exist.	PROJECT_NUMBER, TASK_NUMBER
PA_TSK_TXN_CONT_EXIST_ST_AMG	28	You cannot create a subtask below this task since transaction controls exist.	PROJECT_NUMBER, TASK_NUMBER

Table 9 – 1 AMG Messages (Page 6 of 6)

Index

A

- accumulated period actuals, 2-21
- Activity Management Gateway. *See* AMG
- actuals, 2-21, 8-9, 8-15
- ADD_BUDGET_LINE, 6-9
- ADD_FUNDING, 9-9
- ADD_RESOURCE_LIST_MEMBER, 7-15
- ADD_TASK, 5-22
- Agreement
 - check delete, 9-19
 - clear, 9-19
 - create, 9-6, 9-16
 - delete, 9-7
 - international, 9-14
 - load, 9-14
 - update, 9-8, 9-17
- agreement and funding, APIs, 9-2
- agreements
 - examples, D-2, E-2
 - overview, 3-20
- agreements and funding, 9-3
 - views, 9-3
- AMG
 - defined, 1-1
 - licensing, 2-7
 - messages, I-2
 - overview, 2-2
- APIs
 - agreement and funding, 9-2
 - alphabetical list, H-2
 - budget, 3-10, 6-2
 - common, 2-18

- defined, 1-1
- project, 5-2
- resource, 7-2
- standard parameters, 2-13
- status, 8-2

Application Programming Interfaces. *See* APIs

B

- BASELINE_BUDGET, 6-12
- baselining budgets, 6-9
- BEM. *See* budgets, entry methods
- budget lines
 - adding, 6-10, 6-22, 6-26
 - deleting, 6-19
 - flexfield parameters, 6-23
 - saving data, 6-9
 - updating, 6-13, 6-25, 6-29
- budgets
 - APIs, 6-2
 - baselining, 6-9, 6-12
 - creating, 6-14
 - dates, start and end, 6-15, 6-19
 - deleting, 6-20
 - entry methods, 3-12, 6-19
 - examples, 3-10, B-2, C-2
 - flexfield parameters, 6-23
 - overview, 3-10
 - period names, 6-10, 6-15, 6-19, 6-22, 6-26
 - procedures and views, 6-2
 - saving data, 6-9
 - submitting, 6-12
 - task level of BEM, 6-10, 6-15, 6-22, 6-26
 - types, predefined, 3-10

- updating, 6-21
- business rules
 - pa_agreement_pub.add_funding, 9-9
 - pa_agreement_pub.check_add_funding_ok, 9-20
 - pa_agreement_pub.check_delete_agreement_ok, 9-19
 - pa_agreement_pub.check_delete_funding_ok, 9-21
 - pa_agreement_pub.check_update_funding_ok, 9-22
 - pa_agreement_pub.create_agreement, 9-6
 - pa_agreement_pub.delete_agreement, 9-7
 - pa_agreement_pub.delete_funding, 9-11
 - pa_agreement_pub.execute_create_agreement, 9-17
 - pa_agreement_pub.execute_update_agreement, 9-17
 - pa_agreement_pub.update_agreement, 9-8
 - pa_agreement_pub.update_funding, 9-12

C

- CALCULATE_AMOUNTS, 6-13
- Check procedures, 4-3, 5-3
- CHECK_ADD_FUNDING_OK, 9-20
- CHECK_ADD_SUBTASK_OK, 5-61
- CHECK_CHANGE_PARENT_OK, 5-62
- CHECK_CHANGE_PROJECT_ORG_OK, 5-63
- CHECK_DELETE_AGREEMENT_OK, 9-19
- CHECK_DELETE_FUNDING_OK, 9-21
- CHECK_DELETE_PROJECT_OK, 5-63
- CHECK_DELETE_TASK_OK, 5-64
- CHECK_TASK_NUMBER_CHANGE_OK, 5-65
- CHECK_UNIQUE_PROJECT_REFERENCE, 5-65
- CHECK_UNIQUE_TASK_NUMBER, 5-66
- CHECK_UNIQUE_TASK_REFERENCE, 5-66
- CHECK_UPDATE_FUNDING_OK, 9-22
- CLASS_CATEGORY_TBL_TYPE datatype, 5-32
- CLEAR_AGREEMENT, 9-19

- CLEAR_BUDGET, 6-28
- CLEAR_CREATE_RESOURCE_LIST, 7-23
- CLEAR_PROJECT, 5-51
- CLEAR_UPDATE_MEMBERS, 7-23
- commitments, 8-8
- common APIs, 2-18
- composite datatypes, 2-13, 4-2, C-2, E-2
 - See also* record and table datatypes; standard datatypes
- CREATE_AGREEMENT, 9-6
- CREATE_DRAFT_BUDGET, 6-14
- CREATE_PROJECT, 5-26
- CREATE_RESOURCE_LIST, 7-16

D

- datatypes
 - composite, 2-13, 4-2, C-2, E-2
 - record and table, 5-28 to 5-34
 - standard, 4-2
- dates for
 - budgets, 6-15, 6-19
 - projects, 5-37, 5-41, G-2
 - tasks, 5-37, 5-44, G-2
- defaults, initialization, 2-20
- DELETE_AGREEMENT, 9-7
- DELETE_BUDGET_LINE, 6-19
- DELETE_DRAFT_BUDGET, 6-20
- DELETE_FUNDING, 9-11
- DELETE_PROJECT, 5-37
- DELETE_RESOURCE_LIST, 7-18
- DELETE_RESOURCE_LIST_MEMBER, 7-19
- DELETE_TASK, 5-38

E

- error messages, retrieving, 2-18, 3-9, 3-18
- error messages. *See* messages
- event types, 7-5
- examples
 - composite datatypes, C-2, E-2
 - integrating data, 3-2, 3-10, 3-20
 - Load-Execute-Fetch procedures, A-2, B-2, D-2

- restricting actions, F-2
- tracking dates, G-2
- EXEC_CREATE_RESOURCE_LIST, 7-23
- EXEC_UPDATE_RESOURCE_LIST, 7-24
- EXECUTE_CALCULATE_AMOUNTS, 6-28
- EXECUTE_CREATE_AGREEMENT, 9-16
- EXECUTE_CREATE_DRAFT_BUDGET, 6-30
- EXECUTE_CREATE_PROJECT, 5-51
- EXECUTE_UPDATE_AGREEMENT, 9-17
- EXECUTE_UPDATE_BUDGET, 6-31
- EXECUTE_UPDATE_PROJECT, 5-52
- expenditures, 7-6
- exporting database, 2-7
- extensions, project and task date, G-2
- external systems, overview, 2-2

F

- FETCH_BUDGET_LINE, 6-32
- FETCH_CALCULATE_AMOUNTS, 6-32
- FETCH_FUNDING, 9-18
- FETCH_MEMBERS, 7-24
- FETCH_RESOURCE_LIST, 7-25
- FETCH_TASK, 5-52
- files, output
 - pacrole.lst, 2-5
 - pacruser.lst, 2-6
- flexfields, 6-23
- Funding
 - add, 9-9
 - check adding, 9-20
 - check deleting, 9-21
 - check update, 9-22
 - create, 9-9
 - delete, 9-11
 - fetch, 9-18
 - load, 9-15
 - update, 9-12

G

- GET_ACCUM_PERIOD_INFO, 2-21

- GET_DEFAULTS, 2-20
- GET_MESSAGES, 2-18
- global variables, 2-9
- grants, 2-7

I

- importing database, 2-7
- INIT_AGREEMENT, 9-14
- INIT_BUDGET, 6-33
- INIT_CALCULATE_AMOUNTS, 6-34
- INIT_CREATE_RESOURCE_LIST, 7-25
- INIT_PROJECT, 5-53
- INIT_UPDATE_MEMBERS, 7-25
- integration
 - examples, 3-2, 3-10, 3-20
 - with other systems, 2-2
 - with project management systems, 2-5

L

- license to use AMG, 2-7
- Load-Execute-Fetch procedures
 - budgets, 6-3
 - examples, A-2, B-2, D-2
 - overview, 4-2
 - projects, 5-3
 - resources, 7-3
- LOAD_AGREEMENT, 9-14
- LOAD_BUDGET_LINE, 6-34
- LOAD_CLASS_CATEGORY, 5-53
- LOAD_FUNDING, 9-15
- LOAD_KEY_MEMBER, 5-53
- LOAD_MEMBERS, 7-25
- LOAD_PROJECT, 5-54
- LOAD_RESOURCE_LIST, 7-27
- LOAD_TASK, 5-58

M

- messages, 2-11
 - displaying, 2-12

handling, 2-11
list, 1-2

N

named notation, 2-16
names, user, 2-8
naming projects, 5-41
numbering
 projects, 5-36
 tasks, 5-46

O

output files
 pacrole.lst, 2-5
 pacruser.lst, 2-6

P

PA_ACCUM_CMT_TXNS_V, 8-8
PA_ACCUM_RSRC_ACT_V, 8-9
PA_ACCUM_RSRC_CMT_V, 8-10
PA_ACCUM_RSRC_COST_BGT_V, 8-12
PA_ACCUM_RSRC_REV_BGT_V, 8-14
PA_ACCUM_WBS_ACT_V, 8-15
PA_ACCUM_WBS_CMT_V, 8-17
PA_ACCUM_WBS_COST_BGT_V, 8-18
PA_ACCUM_WBS_REV_BGT_V, 8-20
PA_ACT_BY_GL_PERIOD_V, 8-21
PA_ACT_BY_PA_PERIOD_V, 8-22
PA_AGREEMENT_PUB.UPDATE_AGREEMENT,
 9-8
PA_AGREEMENT_TYPE_LOV_V, 9-4
PA_AMG_RESOURCE_INFO_V, 7-4
PA_BASE_BUDGET_BY_GL_PERIOD_V, 6-4
PA_BASE_BUDGET_BY_PA_PERIOD_V, 6-4
PA_BUDGET_CHANGE_REASON_V, 6-5
PA_BUDGET_ENTRY_METHODS_V, 6-6
PA_BUDGET_STATUS_CODES_V, 6-6
PA_BUDGET_TYPES_V, 6-7

PA_BURDEN_COMPONENT_CMT_V, 8-22
PA_BURDEN_COMPONENT_COST_V, 8-23
PA_CLASS_CATEGORIES_LOV_V, 5-4
pa_client_extn_pm, G-2
PA_CMT_BY_GL_PERIOD_V, 8-24
PA_CMT_BY_PA_PERIOD_V, 8-25
PA_CUSTOMERS_LOV_V, 5-4, 9-5
PA_DISTRIBUTION_RULES_LOV_V, 5-4
PA_EMPLOYEES_RES_V, 7-5
PA_EVENT_TYPES_RES_V, 7-5
PA_EXPEND_CATEGORIES_RES_V, 7-6
PA_EXPENDITURE_TYPES_RES_V, 7-6
PA_GL_PERIODS_V, 8-25
PA_INTERFACE_UTILS_PUB, 2-7, 2-9
PA_JOBS_RES_V, 7-7
PA_KEY_MEMBERS_LOV_V, 5-5
PA_LOWEST_LEVEL_RESOURCES, 7-8
PA_ORGANIZATIONS_LOV_V, 5-5
PA_ORGANIZATIONS_RES_V, 7-9
PA_ORIG_BUDGET_BY_GL_PERIOD_V, 6-7
PA_ORIG_BUDGET_BY_PA_PERIOD_V, 6-8
PA_OVERRIDE_FIELD_VALUES_V, 5-8
PA_OVERRIDE_FIELDS_V, 5-6
PA_OWNED_BY_LOV_V, 9-5
PA_PA_PERIODS_V, 8-25
PA_PM_REFERENCE_V, 8-26
PA_PROJ_ORG_STRUCTURES_V, 7-9
PA_PROJECT_STATUS_LOV_V, 5-8
PA_PROJECTS_AMG_V, 5-9
PA_QRY_RESOURCE_LISTS_V, 7-10
PA_QUERY_RES_LIST_MEMBERS_V, 7-10
PA_RESOURCE_LIST_GROUPS_V, 7-11
PA_RESOURCE_LIST_V, 7-12
PA_RESOURCE_TYPES_ACTIVE_V, 7-12
PA_REVENUE_CATEGORIES_RES_V, 7-13
PA_SELECT_TEMPLATE_V, 5-14
PA_SERVICE_TYPE_LOV_V, 5-15
PA_TASK MANAGERS_LOV_V, 5-15
PA_TASKS_AMG_V, 5-17
PA_TERMS_LOV_A, 9-4
PA_TXN_ACCUM_V, 8-27
PA_USER_RESP_V, 2-8

- PA_VENDORS_RES_V, 7-13
- pacrgran.sql, 2-7
- pacrrole.sql, 2-5
- pacruser.sql, 2-6
- PAPMGEb.pls, G-2
- parameters, standard datatypes, 4-2, 5-22
- period accumulation, 2-21
- procedure definitions
 - budget, 6-2
 - check project and task, 5-3
 - Load-Execute-Fetch, 5-3, 6-3, 7-3
 - overview, 4-2
 - project and task, 5-2
 - resource list, 7-3
 - status, 8-3
- product codes, 6-15
- prohibiting actions, F-2
- Project Status Inquiry. *See* PSI
- PROJECT_IN_REC_TYPE datatype, 5-28
- PROJECT_OUT_REC_TYPE datatype, 5-31
- PROJECT_ROLE_TBL_TYPE datatype, 5-31
- projects
 - See also* tasks
 - APIs, 5-2
 - attributes, changing, 5-40
 - commitments, 8-8
 - creating, 5-26
 - dates, start and finish, 5-37, 5-41, G-3
 - defining, 5-2
 - deleting, 5-37
 - examples, A-2
 - numbering, 5-36
 - overview, 3-2
 - procedures and views, 5-2
 - updating, 5-40
- PSI, 2-4, 8-2

R

- record and table datatypes, 5-28 to 5-34
- resource list members
 - See also* resource lists; resources
 - adding, 7-15

- deleting, 7-19
- retrieving, 7-10
- sorting, 7-19
- updating, 7-22
- resource lists
 - See also* resource list members; resources
 - creating, 7-16
 - deleting, 7-18
 - names, 7-27
 - retrieving, 7-10, 7-11, 7-12
 - updating, 7-15, 7-20, 7-22
- resources
 - See also* resource list members; resource lists
 - APIs, 7-2
 - commitments, 8-8
 - defining, 7-5, 7-7, 7-8, 7-9, 7-13
 - procedures and views, 7-2
 - retrieving, 7-12
 - tracked as labor, 8-4
 - values predefined by Oracle Projects, 8-4
 - vendors, 7-13
- restricting actions, F-2
- retrieving error messages, 2-18, 3-9, 3-18
- revenue categories, 7-13

S

- scripts
 - pacrgran.sql, 2-7
 - pacrrole.sql, 2-5
 - pacruser.sql, 2-6
- security requirements, 2-8
- SET_GLOBAL_INFO, 2-7, 2-9
- site license, 2-7
- SORT_RESOURCE_LIST_MEMBERS, 7-19
- standard API parameters, 2-13
- standard datatypes, 4-2, A-2, D-2
- status APIs, 8-2
- submitting, budgets, 6-12
- summary amounts
 - commitment, 8-17
 - cost budget, 8-18
 - resource commitment, 8-10
 - resource cost budget, 8-12

- resource revenue, 8-9
- resource revenue budget, 8-14
- revenue, 8-15
- revenue budget, 8-20
- systems, external, 2-2

T

table and record datatypes, 5-28 to 5-34

TABLE_IN_TBL_TYPE datatype, 5-32

TASK_OUT_TBL_TYPE datatype, 5-36

tasks

See also projects

adding, 5-22

attributes, changing, 5-43, 5-44, 5-46, 5-47

commitments, 8-8

dates, start and finish, 5-37, 5-44, 5-46, G-2,
G-3

deleting, 5-38

fields, 5-44, 5-47

interface to Oracle Projects, 5-42, 5-46

loading task information, 5-58

moving within the WBS, 5-44, 5-47

numbering, 5-22, 5-42, 5-46

subtasks, creating, 5-22

updating, 5-45

U

UPDATE_BUDGET, 6-21

UPDATE_BUDGET_LINE, 6-25

UPDATE_EARNED_VALUE, 8-29

UPDATE_FUNDING, 9-12

UPDATE_PROGRESS, 8-30

UPDATE_PROJECT, 5-40

UPDATE_RESOURCE_LIST, 7-20

UPDATE_RESOURCE_LIST_MEMBER, 7-22

UPDATE_TASK, 5-45

usernames, 2-8

V

variables, global, 2-9

vendors, 7-13

verifying data, 4-3

view definitions

 budget, 6-2

 overview, 4-2

 project, 5-2

 resource, 7-2

 status, 8-2

views, alphabetical list, H-8

W

WBS, 5-2, 5-39, 5-44, 5-47

work breakdown structure. *See* WBS

Reader's Comment Form

Oracle Activity Management Gateway Technical Reference Manual A83510-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use 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 about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (650) 506-7000 Fax: (650) 506-7200

If you would like a reply, please give your name, address, and telephone number below:

Thank you for helping us improve our documentation.



CAUTION!

This document contains proprietary, confidential information that is the exclusive property of Oracle Corporation.

If you are an Oracle customer, this numbered copy is registered in your site's name. Use, duplication, and disclosure are severely restricted and subject to restrictions stated in your contract with Oracle Corporation.

If you are an Oracle employee, this numbered copy is registered in your name and must be returned if you leave the company. It is your responsibility not to disclose this information to anyone outside of Oracle who is not licensed to use this product. If you have a prospect who needs to see this information, contact mailid APPSAPP to obtain a special nondisclosure agreement.

If you are neither an Oracle customer with a valid contract for the use of this document nor an Oracle employee, then you received this document in an unauthorized manner and are not legally entitled to possess or read it. Please return it immediately to Oracle. You may call collect to the office of the Senior Vice President, Applications Division, to arrange its return.

Copy #

