

Oracle® Incentive Compensation

API Reference Guide

Release 11*i*

Part No. B12405-01

March 2004

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

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
------------------------------------	------------

Preface.....	ix
---------------------	-----------

How To Use This Guide	x
Documentation Accessibility	x
Other Information Sources	xi
Training and Support.....	xvi
Do Not Use Database Tools to Modify Oracle Applications Data	xvii
About Oracle	xvii

1 Oracle Incentive Compensation

API Reference

1.1	Oracle Incentive Compensation Public Packages.....	1-1
1.2	Package CN_CALC_SUBMISSION_PUB	1-10
1.2.1	Data Structure Specifications.....	1-11
1.2.2	Create Calculation Submission	1-14
1.2.3	Update Calculation Submission.....	1-16
1.3	Package CN_COMP_PLAN_PUB.....	1-19
1.3.1	Data Structure Specifications.....	1-20
1.3.2	Create Comp Plan	1-22
1.4	Package CN_MULTI_RATE_SCHEDULES_PUB	1-23
1.4.1	Data Structure Specifications.....	1-23
1.4.2	Create Schedule	1-25
1.4.3	Update Schedule	1-26

1.4.4	Delete Schedule.....	1-27
1.4.5	Create Dimension Assign.....	1-28
1.4.6	Update Dimension Assign	1-30
1.4.7	Delete Dimension Assign.....	1-31
1.4.8	Update Rate.....	1-32
1.4.9	Create Dimension.....	1-34
1.4.10	Create Dimension.....	1-35
1.4.11	Delete Dimension	1-36
1.4.12	Create Tier	1-37
1.4.13	Update Tier.....	1-39
1.4.14	Delete Tier	1-40
1.5	Package CN_ROLE_PLANS_PUB	1-41
1.5.1	Data Structure Specifications.....	1-42
1.5.2	Create Role Plans.....	1-43
1.5.3	Update Role Plans	1-44
1.5.4	Delete Role Plans	1-46
1.6	Package CN_SRP_CUSTOMIZE_PUB	1-47
1.6.1	Data Structure Specifications.....	1-47
1.6.2	Update SRP Quota Assigns.....	1-47
1.6.3	Update SRP Quota Rules.....	1-49
1.6.4	Update SRP Rule Uplifts	1-50
1.6.5	Change SRP Quota Custom Flag	1-52
1.7	Package CN_COMMISSION_CALC_PUB	1-53
1.7.1	Data Structure Specifications.....	1-53
1.7.2	Calculate Commission.....	1-54
1.8	Package CN_PLAN_ELEMENT_PUB.....	1-55
1.8.1	Data Structure Specifications.....	1-56
1.8.2	Create Plan Element.....	1-60
1.8.3	Delete Plan Elements	1-62
1.8.4	Duplicate Plan Element	1-64
1.8.5	Update Plan Element	1-65
1.9	Package CN_SCA_CREDITS_BATCH_PUB	1-68
1.9.1	Data Structure Specifications.....	1-68
1.9.2	Get Sales Credits.....	1-68
1.10	Package CN_SCA_CREDITS_ONLINE_PUB	1-69

1.10.1	Data Structure Specifications.....	1-69
1.10.2	Get Sales Credits.....	1-70
1.11	Package CN_SCA_WF_PKG.....	1-71
1.11.1	Data Structure Specifications.....	1-71
1.11.2	START_PROCESS (1).....	1-71
1.11.3	START_PROCESS (2).....	1-72
1.11.4	START_PROCESS (3).....	1-73
1.12	Package CN_SCA_WF_CUST_PKG	1-74
1.12.1	Data Structure Specifications.....	1-74
1.12.2	CUST_REV_DIST	1-75
1.12.3	CUST_REV_DIST	1-76
1.13	Package CN_POST_COLLECTION_TAE_PUB.....	1-77
1.13.1	Data Structure Specifications.....	1-77
1.13.2	Get_Assignments	1-77
1.14	Package CN_PROCESS_TAE_TRX_PUB.....	1-78
1.14.1	Data Structure Specifications.....	1-78
1.14.2	Process_Trx_Records	1-78
1.15	Package CN_PRD_QUOTA_PUB	1-79
1.15.1	Data Structure Specifications.....	1-79
1.15.2	Distribute_Prd_Quota	1-80
1.16	Package CN_SRP_PRD_QUOTA_PUB.....	1-82
1.16.1	Data Structure Specifications.....	1-82
1.16.2	Distribute_Srp_Prd_Quota	1-83
1.17	Package CN_PAYGROUP_PUB.....	1-85
1.17.1	Data Structure Specifications.....	1-86
1.17.2	Get_Pay_Group_Sum	1-89
1.17.3	CreatePayGroup	1-91
1.17.4	Update_PayGroup	1-93
1.17.5	Delete_PayGroup	1-94
1.17.6	Assign_Salesreps	1-96
1.17.7	Update_Salesrep_Assignment	1-97
1.18	Package CN_PMT_PLAN_GET_PUB	1-99
1.18.1	Data Structure Specifications.....	1-99
1.18.2	Get_payment_plans	1-101
1.19	Package CN_PMTPLAN_PUB	1-103

1.19.1	Data Structure Specifications.....	1-103
1.19.2	Create_PmtPlan	1-105
1.19.3	Update_PmtPlan.....	1-106
1.19.4	Delete_PmtPlan	1-108
1.20	Package CN_PMTSUB_PUB	1-110
1.20.1	Data Structure Specifications.....	1-110
1.20.2	Pay	1-110
1.20.3	Pay_Payrun_conc	1-111
1.20.4	submit_request	1-112
1.21	Package CN_SRP_PMT_PLANS_PUB	1-113
1.21.1	Data Structure Specifications.....	1-114
1.21.2	Create_Srp_Pmt_Plan	1-117
1.21.3	Create_Mass_Asgn_Srp_Pmt_Plan.....	1-118
1.21.4	Update_Srp_Pmt_Plan	1-119
1.21.5	Update_Mass_Asgn_Srp_Pmt_Plan.....	1-121
1.21.6	Delete_Srp_Pmt_Plan	1-122
1.21.7	Delete_Mass_Asgn_Srp_Pmt_Plan.....	1-124
1.22	Package CN_WKSHT_CT_UP_PUB.....	1-125
1.22.1	Data Structure Specifications.....	1-125
1.22.2	Create_delete_Wrkhst.....	1-128
1.22.3	Apply_payment_plan_upd.....	1-129
1.22.4	Apply_payment_plan_cre.....	1-131
1.22.5	Apply_payment_plan_del	1-132
1.23	Package CN_WKSHT_GET_PUB.....	1-133
1.23.1	Data Structure Specifications.....	1-134
1.23.2	get_srp_wksht.....	1-135
1.24	Package CN_SRP_PAYGROUP_PUB.....	1-137
1.24.1	Data Structure Specifications.....	1-137
1.24.2	Assign_Salesreps	1-140
1.24.3	Create_Mass_Asgn_Srp_Pay_Group	1-141
1.24.4	Update_salesrep_assignment	1-142
1.24.5	Update_Mass_Asgn_Srp_Pay	1-144
1.25	Messages and Notifications	1-145
1.25.1	Common Messages	1-146
1.25.2	CN_COMP_PLAN_PUB Messages	1-147

1.25.3	CN_ROLE_PLANS_PUB Messages.....	1-148
1.25.4	CN_COMMISSION_CALC_PUB Messages	1-150
1.25.5	CN_PLAN_ELEMENT_PUB Messages	1-150
1.25.6	CN_SCA_CREDITS_ONLINE_PUB Messages.....	1-152
1.25.7	CN_SCA_WF_PXG Messages	1-152
1.25.8	CN_PRD_QUOTA_PUB Messages	1-152
1.25.9	CN_SRP_PRD_QUOTA_PUB Messages	1-153
1.25.10	CN_PAYGROUP_PUB Messages.....	1-154
1.25.11	CN_PMTPLAN_PUB Messages.....	1-156
1.25.12	CN_PMTSUB_PUB Messages	1-157
1.25.13	CN_SRP_PMT_PLANS_PUB Messages	1-157
1.25.14	CN_WKSHT_GET_PUB Messages.....	1-158
1.25.15	CN_SRP_PAYGROUP_PUB Messages.....	1-158

A API Examples

A.1	Assigning Payment Plans to Resources	A-1
A.2	Updating Resources' Pay Group Assignment	A-5
A.3	Assigning Compensation Plans to Roles	A-10

Send Us Your Comments

Oracle Incentive Compensation API Reference Guide, Release 11i

Part No. B12405-01

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

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

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

- Electronic mail: appsdoc_us@oracle.com
- FAX: (650) 506-7200 Attn: Oracle Applications Documentation Manager
- Postal service:
Oracle Corporation
Oracle Applications Documentation Manager
500 Oracle Parkway
Redwood Shores, CA 94065
USA

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

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

Preface

Welcome to the Oracle Incentive Compensation API Reference Guide, Release 11*i*.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle Incentive Compensation.

If you have never used Oracle Incentive Compensation, Oracle suggests you attend one or more of the Oracle Applications training classes available through Oracle University.

- Oracle Self-Service Web Applications.

To learn more about Oracle Self-Service Web Applications, read the *Oracle Self-Service Web Applications Implementation Manual*.

- The Oracle Applications graphical user interface.

To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See [Other Information Sources](#) for more information about Oracle Applications product information.

How To Use This Guide

The Oracle Incentive Compensation Oracle Incentive Compensation API Reference Guide contains the information you need to understand and use APIs in Oracle Incentive Compensation. This guide contains one chapter with 25 sections

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Other Information Sources

You can choose from many sources of information, including documentation, training, and support services, to increase your knowledge and understanding of Oracle Incentive Compensation.

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

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **PDF Documentation**- See the Documentation CD provided with each release for current PDF documentation for your product. This Documentation CD is also available on *OracleMetaLink* and is updated frequently.
- **Online Help** - You can refer to Oracle iHelp for current HTML online help for your product. Oracle provides patchable online help, which you can apply to your system for updated implementation and end user documentation. No system downtime is required to apply online help.
- **11i Release Content Document** - Refer to the Release Content Document for new features listed release. The Release Content Document is available on *OracleMetaLink*.
- **About document** - Refer to the About document for patches that you have installed to learn about new documentation or documentation patches that you can download. The new About document is available on *OracleMetaLink*.

Related Guides

Oracle Incentive Compensation shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other guides when you set up and use Oracle Incentive Compensation.

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at <http://oraclestore.oracle.com>.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

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

Guides Related to This Product

Oracle Incentive Compensation User Guide

Use this user guide to manage Oracle Incentive Compensation on a day-to-day basis. Learn how to distribute quota, create compensation plans, collect transactions, and calculate and pay commission. See how to use Incentive Planning and modeling. The new Sales Credit Allocation feature is presented in detail.

Oracle Incentive Compensation Implementation Guide

This guide is designed to provide guidance during implementation of Oracle Incentive Compensation. The procedures are presented in the order that they must be performed for successful implementation. Appendixes are included that list system profiles, lookups, and other useful information.

Oracle Incentive Compensation Technical Reference Manual

Use this manual to obtain the specific technical details used in setting up Oracle Incentive Compensation.

Installation and System Administration

Oracle Applications Concepts

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

Installing Oracle Applications

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

Oracle Applications Implementation Wizard User Guide

If you are implementing more than one Oracle product, you can use the Oracle Applications Implementation Wizard to coordinate your setup activities. This guide describes how to use the wizard.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

“About” Document

For information about implementation and user documentation, instructions for applying patches, new and changed setup steps, and descriptions of software updates, refer to the “About” document for your product. “About” documents are available on Oracle *MetaLink* for most products starting with Release 11.5.8.

Maintaining Oracle Applications

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

Oracle Applications System Administrator’s Guide

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

Oracle Alert User's Guide

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

Oracle Applications Developer's Guide

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

Oracle Applications User Interface Standards for Forms-Based Products

This guide 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.

Other Implementation Documentation

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11i. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Applications Flexfields Guide

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

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*Metalink*

Oracle Applications Message Manual

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11*i*.

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Incentive Compensation and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

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

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Incentive Compensation working for you. This team includes your technical representative, account manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle server, and your hardware and software environment.

Oracle*MetaLink*

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

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

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

Do Not Use Database Tools to Modify Oracle Applications Data

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

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

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

About Oracle

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

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

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

Oracle Incentive Compensation API Reference

This appendix contains public APIs for Oracle Incentive Compensation 11i. These APIs are used for submitting calculation, creating compensation plans, building rate tables, creating and modifying roles, working with quotas, calculating commissions, working with plan elements, many aspects of payment and payruns, and for Sales Credit Allocation.

Section A.1 below lists the APIs and contains links to the tables that describes their content. The complete APIs appear in detail in sections A.2 through A.24, followed by section A.25, which contains Messages and Notifications.

1.1 Oracle Incentive Compensation Public Packages

These are the public packages covered in this appendix.

- CN_CALC_SUBMISSION_PUB, see [Table 1–1, "Calculation Submission APIs"](#).
- CN_COMP_PLAN_PUB, see [Table 1–2, "Compensation Plan API"](#).
- CN_MULTI_RATE_SCHEDULES_PUB, see [Table 1–3, "Rate Table APIs"](#).
- CN_ROLE_PLANS_PUB, see [Table 1–4, "Role-Plans APIs"](#).
- CN_SRP_CUSTOMIZE_PUB, see [Table 1–5, "Srp-Customize APIs"](#).
- CN_COMMISSION_CALC_PUB, see [Table 1–6, "Projected Compensation API"](#).
- CN_PLAN_ELEMENT_PUB, see [Table 1–7, "Plan Element APIs"](#).
- CN_SCA_CREDITS_BATCH_PUB, see [Table 1–8, "Sales Credit Allocation Rules Engine \(Batch Mode\) API"](#).

- CN_SCA_CREDITS_ONLINE_PUB, see [Table 1–9, "Sales Credit Allocation Online Rules Engine API"](#).
- CN_SCA_WF_PKG, see [Table 1–10, "Sales Credit Allocation Standard Workflow APIs"](#).
- CN_SCA_WF_CUST_PKG, see [Table 1–11, "Sales Credit Allocation Custom Workflow APIs"](#).
- CN_POST_COLLECTION_TAE_PUB, see [Table 1–12, "Territory Assignment Engine Collection API"](#).
- CN_PROCESS_TAE_TRX_PUB, see [Table 1–13, "Territory Assignment Engine Population API"](#).
- CN_PRD_QUOTA_PUB, see [Table 1–14, "Period Quotas Distribution API"](#).
- CN_SRP_PRD_QUOTA_PUB, see [Table 1–15, "Salesperson Period Quotas Distribution API"](#).
- CN_PAYGROUP_PUB, see [Table 1–16, "Pay Group APIs"](#).
- CN_PMT_PLAN_GET_PUB, see [Table 1–17, "Get Payment Plan API"](#).
- CN_PMTPLAN_PUB, see [Table 1–18, "Payment Plan APIs"](#).
- CN_PMTSUB_PUB, see [Table 1–19, "Submit Payment APIs"](#).
- CN_SRP_PMT_PLANS_PUB, see [Table 1–20, "Assign Payment Plans APIs"](#).
- CN_WKSHT_CT_UP_PUB, see [Table 1–21, "Create and Update Worksheet APIs"](#).
- CN_WKSHT_GET_PUB, see [Table 1–22, "Get Worksheet API"](#).
- CN_SRP_PAYGROUP_PUB, see [Table 1–23, "Salesrep Paygroup APIs"](#).

The following tables describe the public APIs that are discussed in this appendix.

Table 1–1 Calculation Submission APIs

Name	Description
Create_Calc_Submission	This procedure creates a new calculation submission batch with the given specifications.
Update_Calc_Submission	This procedure updates an existing calculation submission batch if applicable.

Table 1–2 Compensation Plan API

Name	Description
Create_Comp_Plan	This procedure creates a compensation plan with the given specifications. It also lets the user to create a plan element assignment.

Table 1–3 Rate Table APIs

Name	Description
Create_Rate_Schedule	This procedure creates a rate schedule with the given specifications. It also lets the user to assign rate dimensions at the same time.
Update_Rate_Schedule	This procedure updates a rate schedule according to the given specifications. It also lets the user to change the assignment of rate dimensions at the same time.
Delete_Rate_Schedule	This procedure deletes a rate schedule.
Create_Dimension_Assign	This procedure assigns a new rate dimension to the given rate schedule.
Update_Dimension_Assign	This procedure updates a given rate dimension assignment.
Delete_Dimension_Assign	This procedure deletes a rate dimension assignment.
Update_Rate	This procedure updates a commission rate in the given rate schedule.
Create_Dimension	This procedure creates a new rate dimension.
Update_Dimension	This procedure updates a given rate dimension.
Delete_Dimension	This procedure deletes a rate dimension.
Create_Tier	This procedure adds a new tier to a given dimension.
Update_Tier	This procedure updates a tier assigned to a dimension.
Delete_Tier	This procedure removes a rate tier from a given dimension.

Table 1–4 Role-Plans APIs

Name	Description
Create_Role_Plan	This procedure creates Role-Plan assignment. For all salesreps, with this Role-Plan assignment records are created in Salesrep-Plan Assigns, Salesrep-Quota Assigns, Salesrep-Periods Quotas, Salesrep-Rate Tiers, Salesrep-Rule Uplifts.
Update_Role_Plan	This procedure updates Role-Plan assignment. For all salesreps, with this Role-Plan assignment records are updated in Salesrep-Plan Assigns, Salesrep-Quota Assigns, Salesrep-Periods Quotas, Salesrep-Rate Tiers, Salesrep-Rule Uplifts.
Delete_Role_Plan	This procedure deletes Role-Plan assignment. For all salesreps, with this Role-Plan assignment records are deleted in Salesrep-Plan Assigns, Salesrep-Quota Assigns, Salesrep-Periods Quotas, Salesrep-Rate Tiers, Salesrep-Rule Uplifts.

Table 1–5 Srp-Customize APIs

Name	Description
Update_srp_quota_assign	This procedure updates Salesrep-Quota Assigns.
Update_srp_quota_rules	This procedure updates Salesrep-Quota Rules.
Update_srp_rule_uplifts	This procedure updates Salesrep-Rule Uplifts.
Change_srp_quota_custom_flag	This procedure updates customized flag of cn_Srp_quota_assigns.

Table 1–6 Projected Compensation API

Name	Description
Calculate_Commission	Calculate_commission procedure in cn_commission_calc_pub is used for calculating projected compensation for the given salesrep, projection identifier, calculation date, sales credit amount. These details are inserted into the 'cn_proj_compensation_gtt', a Global Temporary Table, by the calling program.

Table 1–7 Plan Element APIs

Name	Description
Create_Plan_Element	Create_Plan_Element in cn_plan_element_pub creates a plan element and create entries in the child tables. The possible child table entries are for: ■ Quota Rules ■ Rule Uplifts ■ Trx Factors ■ Rollover Quotas ■ Period_quotas
Delete_Plan_Element	Delete_Plan_Element in cn_plan_element_pub deletes a plan element and deletes respective entries in the child tables. The possible child table entries are for: ■ Quota Rules ■ Rule Uplifts ■ Trx Factors ■ Rollover Quotas If the plan element belongs to a compensation plan, the status is set to <i>Incomplete</i>. If the plan element has been assigned to the salesreps, then the records are also deleted from srp quota assigns, srp rate assigns, srp period quotas, srp quota rules and srp rollover quotas.
Duplicate_Plan_Element	Duplicate_Plan_Element in cn_plan_element_pub copies the information of an existing plan element. The child table entries are also copied from the source.
Update_Plan_Element	Update_Plan_Element in cn_plan_element_pub updates a plan element and updates respective entries in the child tables. The possible child table entries are for: Quota Rules Rule Uplifts Trx Factors Roll over quotas If the plan element belongs to a compensation plan, the status is set to <i>Incomplete</i>. If the plan element has been assigned to the salesreps, then the records are also updated in srp quota assigns, srp rate assigns, srp period quotas, srp quota rules and srp rollover quotas only if the plan element to salesrep assignment is not customizable.

Table 1–8 Sales Credit Allocation Rules Engine (Batch Mode) API

Name	Description
Get_Sales_Credits	The get sales credits procedure is used to retrieve sales credit allocation distribution for the transactions passed as input to the procedure. The input to this API is via interface tables (CN_SCA_HEADERS_INTERFACE_ALL and CN_SCA_LINES_INTERFACE_ALL). The input data is processed and the revenue and non-revenue allocation percentages are decided based on the Credit Rules setup data. The allocation percentages, along with the Resource and Role information, are stored in the output interface tables (CN_SCA_OUTPUT_LINES_ALL).

Table 1–9 Sales Credit Allocation Online Rules Engine API

Name	Description
get_sales_credits	The get sales credit procedure is used to retrieve sales credit allocation distribution for the transactions passed as input to the procedure. The input to this API is through a Global Temporary table (cn_sca_headers_interface_gtt, cn_sca_lines_interface_gtt). The input data is processed and the revenue and non-revenue allocation percentages are decided based on the rules setup data. The allocation percentages, along with the resource information, are returned to the caller using a Global temporary table (cn_sca_lines_output_gtt).

Table 1–10 Sales Credit Allocation Standard Workflow APIs

Name	Description
START_PROCESS (1)	This procedure creates a workflow process with the given specifications that distributes revenue between transactions processed in Batch mode.
START_PROCESS (2)	This procedure creates a workflow process with the given specifications that distributes revenue between transactions processed in Online mode.

Table 1–10 Sales Credit Allocation Standard Workflow APIs

Name	Description
START_PROCESS (3)	This procedure creates a deferred workflow process with the given specifications that transfers transactions from the Sales Credit Allocation batch interface tables to the Oracle Incentive Compensation Transaction Interface table, or to custom table(s).

Table 1–11 Sales Credit Allocation Custom Workflow APIs

Name	Description
CUST_REV_DIST	This procedure is an empty stub, allowing the customer to implement a custom revenue distribution process.
CUST_TRX_LOAD	This procedure is an empty stub, allowing the customer to implement a custom transfer of transactions from the Sales Credit Allocation batch interface tables to custom table(s).

Table 1–12 Territory Assignment Engine Collection API

Name	Description
get_assignments	This procedure provides a code template that can be used to integrate with the Territory Assignment Engine in the post collection process. This procedure contains the following 3 steps: 1. Populates records from <code>cn_comm_lines_api</code> into the input interface table for Territory Assignment Engine. 2. Calls the Territory Assignment Engine to process the records in the input interface table. 3. Populates results from the Territory Assignment Engine output interface table back to <code>cn_comm_lines_api</code>

Table 1–13 Territory Assignment Engine Population API

Name	Description
Process_Trx_Records	This procedure incorporates results from the Territory Assignment Engine output interface table into Oracle Incentive Compensation by negating the original transactions and creating new transactions in <code>cn_comm_lines_api</code> .

Table 1–14 Period Quotas Distribution API

Name	Description
Distribute_Prd_Quota	This procedure distributes the target for a plan element across the periods for which the plan element has been defined.

Table 1–15 Salesperson Period Quotas Distribution API

Name	Description
Distribute_Srp_Prd_Quota	This procedure distributes the target for a plan element across the periods for which the plan element has been defined for a given salesperson.

Table 1–16 Pay Group APIs

Name	Description
Get_Pay_Group_Sum	This procedure gets the pay group information.
Create_PayGroup	This procedure validates the input for a pay group and create one if all validations are passed.
Update_PayGroup	This procedure is used to update pay groups.
Delete_PayGroup	This procedure is used to delete pay groups.
Assign_salesreps	This procedure is used to create entry into cn_pay_groups.
Update_salesrep_assignment	This procedure updates the salesrep pay group assignment.

Table 1–17 Get Payment Plan API

Name	Description
Get_payment_plans	This procedure is used to get a salesrep's payment plan for the given role or just to get the payment plans.

Table 1–18 Payment Plan APIs

Name	Description
Create_PmtPlan	This procedure validates the input for a payment plan and creates one if all validations are passed.

Table 1–18 Payment Plan APIs

Name	Description
Update_PmtPlan	This procedure is used to update payment plans.
Delete_PmtPlan	This procedure is used to delete payment plans.

Table 1–19 Submit Payment APIs

Name	Description
Pay	This procedure is used to pay a payrun and update the subledger.
Pay_Payrun_conc	This procedure is used as the executable for the concurrent program.
submit_request	This procedure uses a concurrent program to pay a payrun.

Table 1–20 Assign Payment Plans APIs

Name	Description
Create_Srp_Pmt_Plan	This procedure is used to create a new payment plan assignment to a salesrep.
Create_Mass_Asgn_Srp_Pmt_Plan	This procedure is used to create a new <i>mass</i> payment plan assignment to a salesrep.
Update_Srp_Pmt_Plan	This procedure is used to update payment plan assignment of a salesrep.
Update_Mass_Asgn_Srp_Pmt_plan	This procedure is used to update <i>mass</i> payment plan assignment of a salesrep.
Delete_Srp_Pmt_Plan	This procedure is used to delete a payment plan assignment to a salesrep.
Delete_Mass_Asgn_Srp_Pmt_Plan	This procedure is used to delete <i>mass</i> payment plan assignments to a salesrep.

Table 1–21 Create and Update Worksheet APIs

Name	Description
Create_delete_Wrkhst	This procedure is used to refresh the worksheet with the srp_pmt_asgn_id.
Apply_payment_plan_upd	This procedure is used to update a payment plan at salesrep level.
Apply_payment_plan_cre	This procedure is used to create a payment plan at the salesrep level.
Apply_payment_plan_del	This procedure is used to delete a payment plan at the salesrep level.

Table 1–22 Get Worksheet API

Name	Description
get_srp_wksht	This procedure is used to get a salesrep's worksheet information.

Table 1–23 Salesrep Paygroup APIs

Name	Description
Assign_salesreps	This procedure is used to assign salesreps to a pay group.
Create_Mass_Asgn_Srp_Pay	This procedure is used to create a new mass pay group assignment to a salesrep.
Update_srp_assignment	This procedure is used to update the salesrep assignment to a pay group.
Update_Mass_Asgn_Srp_Pay	This procedure is used to mass update the salesrep assignment to a pay group.

1.2 Package CN_CALC_SUBMISSION_PUB

This package contains two APIs:

- Create_Calc_Submission
- Update_Calc_Submission

1.2.1 Data Structure Specifications

The following data structures are used in Package CN_CALC_SUBMISSION_PUB:

1.2.1.1 Data Structure 1

```
TYPE salesrep_rec_type IS RECORD
  (employee_number
    cn_salesreps.employee_number%TYPE := FND_API.G_MISS_CHAR, type
    cn_salesreps.type%TYPE := FND_API.G_MISS_CHAR
  );
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–24 Parameters

Parameter	Data Type	Description
Employee Number	VARCHAR2 (30)	Employee Number
Type	VARCHAR2 (30)	Employee Type

1.2.1.2 Data Structure 2

```
TYPE salesrep_tbl_type IS TABLE OF salesrep_rec_type
  INDEX BY BINARY_INTEGER;
```

```
TYPE plan_element_tbl_type IS TABLE OF cn_quotas.name%TYPE
  INDEX BY BINARY_INTEGER;
```

```
TYPE app_user_resp_rec_type IS RECORD
  ( user_name          fnd_user.user_name%TYPE := FND_API.G_MISS_CHAR,
    responsibility_name fnd_responsibility_vl.responsibility_name%TYPE := FND_
    API.G_MISS_CHAR
  );
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–25 Parameters

Parameter	Data Type	Description
User Name	VARCHAR2 (100)	User Name
Responsibility Type	VARCHAR2 (100)	Responsibility Type

1.2.1.3 Data Structure 3

```

TYPE calc_submission_rec_type IS RECORD
( batch_name                cn_calc_submission_batches.name%TYPE                :=
FND_API.G_MISS_CHAR,
  start_date                cn_calc_submission_batches.start_date%TYPE          :=
FND_API.G_MISS_DATE,
  end_date                  cn_calc_submission_batches.end_date%TYPE            :=
FND_API.G_MISS_DATE,
  calculation_type          cn_calc_submission_batches.calc_type%TYPE           :=
FND_API.G_MISS_CHAR,
  salesrep_option           cn_calc_submission_batches.salesrep_option%TYPE      :=
FND_API.G_MISS_CHAR,
  entire_hierarchy          cn_calc_submission_batches.hierarchy_flag%TYPE       :=
FND_API.G_MISS_CHAR,
  concurrent_calculation    cn_calc_submission_batches.concurrent_flag%TYPE     :=
FND_API.G_MISS_CHAR,
  incremental_calculation   cn_calc_submission_batches.intelligent_flag%TYPE    :=
FND_API.G_MISS_CHAR,
  interval_type             cn_interval_types.name%type                        :=
FND_API.G_MISS_CHAR,
  attribute_category        cn_comp_plans.attribute_category%TYPE := FND_API.G_
MISS_CHAR,
  attribute1                cn_comp_plans.attribute1%TYPE              := FND_API.G_
MISS_CHAR,
  attribute2                cn_comp_plans.attribute2%TYPE              := FND_API.G_
MISS_CHAR,
  attribute3                cn_comp_plans.attribute3%TYPE              := FND_API.G_
MISS_CHAR,
  attribute4                cn_comp_plans.attribute4%TYPE              := FND_API.G_
MISS_CHAR,
  attribute5                cn_comp_plans.attribute5%TYPE              := FND_API.G_
MISS_CHAR,
  attribute6                cn_comp_plans.attribute6%TYPE              := FND_API.G_
MISS_CHAR,
  attribute7                cn_comp_plans.attribute7%TYPE              := FND_API.G_
MISS_CHAR,
  attribute8                cn_comp_plans.attribute8%TYPE              := FND_API.G_

```

```

MISS_CHAR,
    attribute9          cn_comp_plans.attribute9%TYPE          := FND_API.G_
MISS_CHAR,
    attribute10         cn_comp_plans.attribute10%TYPE         := FND_API.G_
MISS_CHAR,
    attribute11         cn_comp_plans.attribute11%TYPE         := FND_API.G_
MISS_CHAR,
    attribute12         cn_comp_plans.attribute12%TYPE         := FND_API.G_
MISS_CHAR,
    attribute13         cn_comp_plans.attribute13%TYPE         := FND_API.G_
MISS_CHAR,
    attribute14         cn_comp_plans.attribute14%TYPE         := FND_API.G_
MISS_CHAR,
    attribute15         cn_comp_plans.attribute15%TYPE         := FND_API.G_
MISS_CHAR
);

```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–26 Parameters

Parameter	Data Type	Description
Batch_Name	VARCHAR2 (30)	Calculation submission batch name. Should uniquely identify the batch.
Start_Date	DATE	Start Date. Must be within opened period.
End_Date	DATE	End Date
Calculation_Type	VARCHAR2 (30)	Calculation type (Commission or Bonus)
Salesrep_Option	VARCHAR2 (20)	Valid values: ALL_REPS/USER_SPECIFY/REPS_IN_NOTIFY_LOG. If calc_type = BONUS, REPS_IN_NOTIFY_LOG is not valid.
Entire_Hierarchy	VARCHAR2 (1)	Entire hierarchy or not. Valid values: Y/N. If salesrep_option = ALL_REPS or REPS_IN_NOTIFY_LOG, hierarchy_flag should be set to N.
Concurrent_Calculation	VARCHAR2 (1)	Concurrent calculation or not. Valid values: Y/N.

Table 1–26 Parameters

Parameter	Data Type	Description
Incremental_Calculation	VARCHAR2 (1)	Incremental calculation or not. Valid values: Y/N. If salesrep_option = REPS_IN_NOTIFY_LOG, intelligent_flag should be set to Y.
Interval_Type	VARCHAR2 (80)	Interval type name. Optional. Valid values: PERIOD/QUARTER/YEAR/ALL. Mandatory when calc_type = BONUS.
Attribute_Category	VARCHAR2 (30)	Attribute category
Attribute1	VARCHAR2 (150)	Standard Attribute Columns
Attribute2	VARCHAR2 (150)	Standard Attribute Columns
Attribute3	VARCHAR2 (150)	Standard Attribute Columns
Attribute4	VARCHAR2 (150)	Standard Attribute Columns
Attribute5	VARCHAR2 (150)	Standard Attribute Columns
Attribute6	VARCHAR2 (150)	Standard Attribute Columns
Attribute7	VARCHAR2 (150)	Standard Attribute Columns
Attribute8	VARCHAR2 (150)	Standard Attribute Columns
Attribute9	VARCHAR2 (150)	Standard Attribute Columns
Attribute10	VARCHAR2 (150)	Standard Attribute Columns
Attribute11	VARCHAR2 (150)	Standard Attribute Columns
Attribute12	VARCHAR2 (150)	Standard Attribute Columns
Attribute13	VARCHAR2 (150)	Standard Attribute Columns
Attribute14	VARCHAR2 (150)	Standard Attribute Columns
Attribute15	VARCHAR2 (150)	Standard Attribute Columns

1.2.2 Create Calculation Submission

Use this procedure to create a new calculation submission batch with passed_in salesreps/ passed_in bonus plan elements.

Procedure Specification

PROCEDURE Create_Calc_Submission

```
(
  p_api_version      IN      NUMBER,
  p_init_msg_list    IN      VARCHAR2 := FND_API.G_FALSE,
  p_commit           IN      VARCHAR2 := FND_API.G_FALSE,
  p_validation_level  IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status     OUT     VARCHAR2,
  x_msg_count        OUT     NUMBER,
  x_msg_data         OUT     VARCHAR2,
  p_calc_submission_rec IN    calc_submission_rec_type := g_miss_calc_
submission_rec,
  p_app_user_resp_rec IN      app_user_resp_rec_type  := g_miss_app_user_resp_
rec,
  p_salesrep_tbl      IN      salesrep_tbl_type       := g_miss_salesrep_tbl,
  p_bonus_pe_tbl      IN      plan_element_tbl_type   := g_miss_pe_tbl,
  x_loading_status    OUT     VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–27 Create_Calc_Submission IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	NUMBER	N	Validation level (default Full).
p_calc_submission_rec	calc_submission_rec_type		Calculation submission record type.
p_app_user_resp_rec	app_user_resp_rec_type	N	Information required to submit concurrent calculation. Valid when concurrent_calculation = Y. User_name should be a valid application user name.

Table 1–27 Create_Calc_Submission IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_salesrep_tbl	salesrep_tbl_type	N	List of salesreps' employee number and employee type. Valid when salesrep_option = USER SPECIFY. salesrep_rec_type.employee number cannot be missing or null. salesrep_rec_type.type cannot be missing or null. Salespeople listed currently have or previously had a compensation plan assigned.
p_bonus_pe_tbl	plan_element_tbl_type	N	List of bonus plan elements. Valid when calc_type = BONUS. Plan elements listed should be BONUS type and their interval type should match the value of p_calc_submission_rec.interval_type. Or, if p_calc_submission_rec.interval_type = ALL, then the interval type can be PERIOD, QUARTER, or YEAR.

Note: If p_commit is not fnd_api.g_true, then the calculation will not be submitted even if all of the validations are successful.

The following table describes the OUT parameters associated with this API.

Table 1–28 Create_Calc_Submission OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1
x_loading_status	VARCHAR2	Loading Status

1.2.3 Update Calculation Submission

Use this API to update a calculation submission with passed_in salesreps and passed_in bonus plan elements.

Procedure Specification

PROCEDURE Update_Calc_Submission

```
(
  p_api_version           IN  NUMBER,
  p_init_msg_list         IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit                IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level      IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status         OUT  VARCHAR2,
  x_msg_count             OUT  NUMBER,
  x_msg_data              OUT  VARCHAR2,
  p_calc_submission_rec_old IN  calc_submission_rec_type := g_miss_calc_
submission_rec,
  p_calc_submission_rec_new IN  calc_submission_rec_type := g_miss_calc_
submission_rec,
  p_app_user_resp_rec     IN  app_user_resp_rec_type  := g_miss_app_user_
resp_rec,
  p_salesrep_tbl          IN  salesrep_tbl_type       := g_miss_salesrep_tbl,
  p_salesrep_tbl_action   IN  VARCHAR2,
  p_bonus_pe_tbl          IN  plan_element_tbl_type   := g_miss_pe_tbl,
  p_bonus_pe_tbl_action   IN  VARCHAR2,
  x_loading_status        OUT  VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–29 Update_Calc_Submission IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F). See Note below table.
p_validation_level	NUMBER	N	Validation level (default 100).

Table 1–29 Update_Calc_Submission IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_calc_submission_rec_old	calc_submission_rec_type	Y	The old calculation submission batch must be found based on p_calc_submission_rec_old.batch_name. If the old calculation submission batch is either completed or in progress, then it cannot be updated.
p_calc_submission_rec_new	calc_submission_rec_type	Y	All the validation rules in create_calc_submission holds here.
p_app_user_resp_rec	app_user_resp_rec_type	N	Information required to submit concurrent calculation. Valid when concurrent_calculation = 'Y'. The user_name should be a valid application user name. The responsibility_name should be a valid responsibility name.
p_salesrep_tbl	salesrep_tbl_type	N	A list of salesreps' employee number /employee type. Valid when salesrep_option = 'USER_SPECIFY'. The salesrep_rec_type.employee number cannot be missing or null. The salesrep_rec_type.type cannot be missing or null. The salespeople listed currently have or previously had compensation plans assigned.
p_salesrep_tbl_action	VARCHAR2	Y	Valid Values: ADD/DELETE. Either adds the listed salespeople to the table or deletes the listed salespeople from the table. If the salesperson already exists or there are duplicates in p_salesrep_tbl, it displays a message without failing the call.

Table 1–29 Update_Calc_Submission IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_bonus_pe_tbl	plan_element_tbl_type	N	A list of bonus plan element names. Valid when calc_type = BONUS. Plan elements listed should be BONUS type and their interval type should match the value of p_calc_submission_rec.interval_type. Or, if p_calc_submission_rec.interval_type = ALL, then their interval type can be any of PERIOD/QUARTER/YEAR.
p_bonus_pe_tbl_action	VARCHAR2	Y	Valid Values: ADD/DELETE. Either adds the listed bonus plan elements to table or deletes the listed bonus plan elements from the table. If the plan element already exists or there are duplicates in p_bonus_pe_tbl, it displays a message without failing the call.

Note: If p_commit is not fnd_api.g_true, then the calculation will not be submitted even if all of the validations are successful.

The following table describes the OUT parameters associated with this API.

Table 1–30 Update_Calc_Submission OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1
x_loading_status	VARCHAR2	Loading Status

1.3 Package CN_COMP_PLAN_PUB

Description of CN_COMP_PLAN_PUB goes here.

- Create Comp Plan

1.3.1 Data Structure Specifications

The following data structures are used in Package CN_COMP_PLAN_PUB:

1.3.1.1 Data Structure

TYPE comp_plan_rec_type **IS** RECORD

```
(
    name                cn_comp_plans.name%TYPE           := CN_API.G_MISS_CHAR,
    description          cn_comp_plans.description%TYPE    := CN_API.G_MISS_CHAR,
    start_date           cn_comp_plans.start_date%TYPE     := CN_API.G_MISS_DATE,
    end_date             cn_comp_plans.end_date%TYPE       := CN_API.G_MISS_DATE,
    status               cn_comp_plans.status_code%TYPE    := CN_API.G_MISS_CHAR,
    rc_overlap           cn_comp_plans.ALLOW_REV_CLASS_OVERLAP%TYPE
                                                                := CN_API.G_MISS_CHAR,
    plan_element_name    cn_quotas.name%TYPE              := CN_API.G_MISS_CHAR,
    attribute_category    cn_comp_plans.attribute_category%TYPE
                                                                := CN_API.G_MISS_CHAR,
    attribute1           cn_comp_plans.attribute1%TYPE     := CN_API.G_MISS_CHAR,
    attribute2           cn_comp_plans.attribute2%TYPE     := CN_API.G_MISS_CHAR,
    attribute3           cn_comp_plans.attribute3%TYPE     := CN_API.G_MISS_CHAR,
    attribute4           cn_comp_plans.attribute4%TYPE     := CN_API.G_MISS_CHAR,
    attribute5           cn_comp_plans.attribute5%TYPE     := CN_API.G_MISS_CHAR,
    attribute6           cn_comp_plans.attribute6%TYPE     := CN_API.G_MISS_CHAR,
    attribute7           cn_comp_plans.attribute7%TYPE     := CN_API.G_MISS_CHAR,
    attribute8           cn_comp_plans.attribute8%TYPE     := CN_API.G_MISS_CHAR,
    attribute9           cn_comp_plans.attribute9%TYPE     := CN_API.G_MISS_CHAR,
    attribute10          cn_comp_plans.attribute10%TYPE    := CN_API.G_MISS_CHAR,
    attribute11          cn_comp_plans.attribute11%TYPE    := CN_API.G_MISS_CHAR,
    attribute12          cn_comp_plans.attribute12%TYPE    := CN_API.G_MISS_CHAR,
    attribute13          cn_comp_plans.attribute13%TYPE    := CN_API.G_MISS_CHAR,
    attribute14          cn_comp_plans.attribute14%TYPE    := CN_API.G_MISS_CHAR,
    attribute15          cn_comp_plans.attribute15%TYPE    := CN_API.G_MISS_CHAR
);
```

TYPE plan_element_tbl_type **IS** TABLE OF cn_quotas.name%TYPE
INDEX BY BINARY_INTEGER;

TYPE comp_plan_list_rec_type **IS** RECORD

```
(
    name                cn_comp_plans.name%TYPE           := CN_API.G_MISS_CHAR,
    description          cn_comp_plans.description%TYPE    := CN_API.G_MISS_CHAR,
    start_date           cn_comp_plans.start_date%TYPE     := CN_API.G_MISS_DATE,
    end_date             cn_comp_plans.end_date%TYPE       := CN_API.G_MISS_DATE,
    status               cn_comp_plans.status_code%TYPE    := CN_API.G_MISS_CHAR,
```

```

rc_overlap          cn_comp_plans.ALLOW_REV_CLASS_OVERLAP%TYPE
                                := CN_API.G_MISS_CHAR,
plan_element_list plan_element_tbl_type      := G_MISS_PE_LIST,
attribute_category  cn_comp_plans.attribute_category%TYPE := CN_API.G_MISS_
CHAR,
attribute1          cn_comp_plans.attribute1%TYPE      := CN_API.G_MISS_CHAR,
attribute2          cn_comp_plans.attribute2%TYPE      := CN_API.G_MISS_CHAR,
attribute3          cn_comp_plans.attribute3%TYPE      := CN_API.G_MISS_CHAR,
attribute4          cn_comp_plans.attribute4%TYPE      := CN_API.G_MISS_CHAR,
attribute5          cn_comp_plans.attribute5%TYPE      := CN_API.G_MISS_CHAR,
attribute6          cn_comp_plans.attribute6%TYPE      := CN_API.G_MISS_CHAR,
attribute7          cn_comp_plans.attribute7%TYPE      := CN_API.G_MISS_CHAR,
attribute8          cn_comp_plans.attribute8%TYPE      := CN_API.G_MISS_CHAR,
attribute9          cn_comp_plans.attribute9%TYPE      := CN_API.G_MISS_CHAR,
attribute10         cn_comp_plans.attribute10%TYPE     := CN_API.G_MISS_CHAR,
attribute11         cn_comp_plans.attribute11%TYPE     := CN_API.G_MISS_CHAR,
attribute12         cn_comp_plans.attribute12%TYPE     := CN_API.G_MISS_CHAR,
attribute13         cn_comp_plans.attribute13%TYPE     := CN_API.G_MISS_CHAR,
attribute14         cn_comp_plans.attribute14%TYPE     := CN_API.G_MISS_CHAR,
attribute15         cn_comp_plans.attribute15%TYPE     := CN_API.G_MISS_CHAR

```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–31 Parameters

Parameter	Data Type	Description
name	VARCHAR2	Comp Plan Name
description	VARCHAR2	Employee Type
start_date	Date	Start Date
end_date	Date	End Date
status	VARCHAR2	Status
rc_overlap	VARCHAR2	Y or N if revenue class overlap is allowed
plan_element_name	VARCHAR2	Name of plan element being assigned
attribute_category	VARCHAR2	Flexfield category
attribute 1-15	VARCHAR2	Flexfields

Parameter Descriptions - comp_plan_list_rec_type

This is the same as the parameter description for comp_plan_rec_type, except that plan_element_name is replaced with plan_element_list, which is a table of plan element names (all Varchar2).

1.3.2 Create Comp Plan

Description of Create Comp Plan goes here.

Procedure Specification

```
PROCEDURE Create_Comp_Plan
(
  p_api_version          IN    NUMBER,
  p_init_msg_list        IN    VARCHAR2 := CN_API.G_FALSE,
  p_commit               IN    VARCHAR2 := CN_API.G_FALSE,
  p_validation_level     IN    NUMBER   := CN_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT   VARCHAR2,
  x_msg_count            OUT   NUMBER,
  x_msg_data             OUT   VARCHAR2,
  p_comp_plan_rec        IN    comp_plan_rec_type := G_MISS_COMP_PLAN_REC,
  x_loading_status       OUT   VARCHAR2,
  x_comp_plan_id         OUT   NUMBER
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–32 Create Comp Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_comp_plan_rec	comp_plan_rec_type		Contents of comp plan

The following table describes the OUT parameters associated with this API.

Table 1–33 Create Comp Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1
x_loading_status	VARCHAR2	Loading Status
x_comp_plan_id	NUMBER	ID of newly created comp plan

1.4 Package CN_MULTI_RATE_SCHEDULES_PUB

Description of CN_MULTI_RATE_SCHEDULES_PUB goes here.

- Create Schedule
- Update Schedule
- Delete Schedule
- Create Dimension Assign
- Update Dimension Assign
- Delete Dimension Assign
- Update Rate
- Create Dimension
- Update Dimension
- Delete Dimension
- Create Tier
- Update Tier
- Delete Tier

1.4.1 Data Structure Specifications

The following data structures are used in Package CN_MULTI_RATE_SCHEDULES_PUB:

1.4.1.1 Data Structure

```
TYPE dim_assign_rec_type IS RECORD
  (rate_schedule_name      CN_RATE_SCHEDULES.NAME%TYPE,
   rate_dim_name           CN_RATE_DIMENSIONS.NAME%TYPE,
   rate_dim_sequence       CN_RATE_SCH_DIMS.RATE_DIM_SEQUENCE%TYPE,
   object_version_number   CN_RATE_DIMENSIONS.OBJECT_VERSION_NUMBER%TYPE);

TYPE dim_assign_tbl_type IS TABLE OF dim_assign_rec_type INDEX BY BINARY_
INTEGER;

TYPE rate_tier_rec_type IS RECORD
  (tier_sequence           CN_RATE_DIM_TIERS.TIER_SEQUENCE%TYPE,
   value1                  VARCHAR2(80),
   value2                  VARCHAR2(80),
   object_version_number   CN_RATE_DIM_TIERS.OBJECT_VERSION_NUMBER%TYPE);

TYPE rate_tier_tbl_type IS TABLE OF rate_tier_rec_type INDEX BY BINARY_INTEGER;

TYPE tier_coordinates_tbl IS TABLE OF NUMBER                INDEX BY BINARY_INTEGER;
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–34 Parameters - dim_assign_rec_type

Parameter	Data Type	Description
rate_schedule_name	VARCHAR2	Rate schedule name
rate_dim_name	VARCHAR2	Rate dimension name
rate_dim_sequence	Number	Rate dimension sequence
object_version_number	Date	Object version

Table 1–35 Parameters - rate_tier_rec_type

Parameter	Data Type	Description
tier_sequence	Number	Rate tier sequence

Table 1–35 Parameters - rate_tier_rec_type

Parameter	Data Type	Description
value1	VARCHAR2	Start value of tier
value2	Number	End value of tier (ignored for string-based dimension)
object_version_number	Date	Object version

1.4.2 Create Schedule

Procedure Specification

```

PROCEDURE Create_Schedule
  (p_api_version           IN      NUMBER,
   p_init_msg_list         IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit                IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_name                  IN      CN_RATE_SCHEDULES.NAME%TYPE,
   p_commission_unit_code  IN      CN_RATE_SCHEDULES.COMMISSION_UNIT_
CODE%TYPE,
   p_dims_tbl              IN      dim_assign_tbl_type := g_miss_dim_
assign_tbl,
   x_return_status          OUT     VARCHAR2,
   x_msg_count              OUT     NUMBER,
   x_msg_data              OUT     VARCHAR2);

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–36 Create Schedule IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).

Table 1–36 Create Schedule IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_name	VARCHAR2	Y	Rate schedule name
p_commission_unit_code	VARCHAR2	Y	Commission unit (AMOUNT or PERCENT)
p_dims_tbl	dim_assign_tbl_type	N	Dimension assignment table (default empty table)

The following table describes the OUT parameters associated with this API.

Table 1–37 Create Schedule Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.3 Update Schedule

Procedure Specification

```
PROCEDURE Update_Schedule
  (p_api_version          IN      NUMBER,
   p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit               IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_original_name         IN      CN_RATE_SCHEDULES.NAME%TYPE,
   p_new_name              IN      CN_RATE_SCHEDULES.NAME%TYPE :=
                                     cn_api.g_miss_char,
   p_commission_unit_code  IN      CN_RATE_SCHEDULES.COMMISSION_UNIT_
CODE%TYPE :=
                                     cn_api.g_miss_char,
   p_object_version_number IN      CN_RATE_SCHEDULES.OBJECT_VERSION_
NUMBER%TYPE,
   p_dims_tbl              IN      dim_assign_tbl_type := g_miss_dim_
assign_tbl,
   x_return_status         OUT     VARCHAR2,
   x_msg_count             OUT     NUMBER,
   x_msg_data              OUT     VARCHAR2);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–38 Update Schedule IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_original_name	VARCHAR2	Y	Original rate schedule name
p_new_name	VARCHAR2	N	New name (leave off is no change)
p_commission_unit_code	VARCHAR2	Y	Commission unit (AMOUNT or PERCENT)
p_object_version_number	Number	Y	Object version
p_dims_tbl	dim_assign_tbl_type	N	Dimension assignment table (default empty table)

The following table describes the OUT parameters associated with this API.

Table 1–39 Update Schedule OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.4 Delete Schedule**Procedure Specification**

PROCEDURE Delete_Schedule

(p_api_version	IN	NUMBER,
p_init_msg_list	IN	VARCHAR2 := FND_API.G_FALSE,
p_commit	IN	VARCHAR2 := FND_API.G_FALSE,
p_validation_level	IN	NUMBER := FND_API.G_VALID_LEVEL_FULL,
p_name	IN	CN_RATE_SCHEDULES.NAME%TYPE,
x_return_status	OUT	VARCHAR2,
x_msg_count	OUT	NUMBER,
x_msg_data	OUT	VARCHAR2);

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–40 Delete Schedule IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_name	VARCHAR2	Y	Rate schedule name

The following table describes the OUT parameters associated with this API.

Table 1–41 Delete Schedule OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.5 Create Dimension Assign

Procedure Specification

PROCEDURE Create_Dimension_Assign

```

(p_api_version          IN      NUMBER,
 p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
 p_commit               IN      VARCHAR2 := FND_API.G_FALSE,
 p_validation_level     IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
 p_rate_schedule_name   IN      CN_RATE_SCHEDULES.NAME%TYPE,
 p_rate_dimension_name  IN      CN_RATE_DIMENSIONS.NAME%TYPE,
 p_rate_dim_sequence    IN      CN_RATE_SCH_DIMS.RATE_DIM_SEQUENCE%TYPE,
 x_return_status        OUT     VARCHAR2,
 x_msg_count            OUT     NUMBER,
 x_msg_data             OUT     VARCHAR2);

```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–42 Create Dimension Assign IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_rate_schedule_name	VARCHAR2	Y	Rate schedule name
p_rate_dimension_name	VARCHAR2	Y	Rate dimension name
p_rate_dim_sequence	NUMBER	Y	Rate dimension sequence

The following table describes the OUT parameters associated with this API.

Table 1–43 Create Dimension Assign OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned

Table 1–43 Create Dimension Assign OUT Parameters

Parameter	Data Type	Descriptions
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.6 Update Dimension Assign

Procedure Specification

```
PROCEDURE Update_Dimension_Assign
(p_api_version          IN          NUMBER,
 p_init_msg_list        IN          VARCHAR2 := FND_API.G_FALSE,
 p_commit               IN          VARCHAR2 := FND_API.G_FALSE,
 p_validation_level     IN          NUMBER   := FND_API.G_VALID_LEVEL_FULL,
 p_rate_schedule_name   IN          CN_RATE_SCHEDULES.NAME%TYPE,
 p_orig_rate_dim_name   IN          CN_RATE_DIMENSIONS.NAME%TYPE,
 p_new_rate_dim_name    IN          CN_RATE_DIMENSIONS.NAME%TYPE := cn_
api.g_miss_char,
 p_rate_dim_sequence    IN          CN_RATE_SCH_DIMS.RATE_DIM_SEQUENCE%TYPE
:=
                                cn_api.g_miss_num,
 p_object_version_number IN          CN_RATE_SCH_DIMS.OBJECT_VERSION_
NUMBER%TYPE,
 x_return_status         OUT         VARCHAR2,
 x_msg_count             OUT         NUMBER,
 x_msg_data              OUT         VARCHAR2);
```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–44 Update Dimension Assign IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).

Table 1–44 Update Dimension Assign IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_rate_schedule_name	VARCHAR2	Y	Rate schedule name
p_orig_rate_dim_name	VARCHAR2	Y	Original rate dimension
p_new_rate_dim_name	VARCHAR2	N	New rate dimension (if changing)
p_rate_dim_sequence	NUMBER	Y	Rate dimension sequence (if changing)
p_object_version_number	NUMBER	Y	Object version

The following table describes the OUT parameters associated with this API.

Table 1–45 Update Dimension Assign OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.7 Delete Dimension Assign

Procedure Specification

```

PROCEDURE Delete_Dimension_Assign
    (p_api_version          IN          NUMBER,
      p_init_msg_list       IN          VARCHAR2 := FND_API.G_FALSE,
      p_commit              IN          VARCHAR2 := FND_API.G_FALSE,
      p_validation_level    IN          NUMBER   := FND_API.G_VALID_LEVEL_FULL,
      p_rate_schedule_name  IN          CN_RATE_SCHEDULES.NAME%TYPE,
      p_rate_dimension_name IN          CN_RATE_DIMENSIONS.NAME%TYPE,
      x_return_status       OUT         VARCHAR2,
      x_msg_count           OUT         NUMBER,
      x_msg_data            OUT         VARCHAR2);

```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–46 Delete Dimension Assign IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_rate_schedule_name	VARCHAR2	Y	Rate schedule name
p_rate_dimension_name	VARCHAR2	Y	Rate dimension name

The following table describes the OUT parameters associated with this API.

Table 1–47 Delete Dimension Assign OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.8 Update Rate

Procedure Specification

```
PROCEDURE Update_Rate
  (p_api_version          IN      NUMBER,
   p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit               IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_rate_schedule_name   IN      CN_RATE_SCHEDULES.NAME%TYPE,
   p_tier_coordinates_tbl IN      tier_coordinates_tbl,
   p_commission_amount     IN      CN_RATE_TIERS.COMMISSION_AMOUNT%TYPE,
   p_object_version_number IN      CN_RATE_TIERS.OBJECT_VERSION_
NUMBER%TYPE,
```

```

x_return_status      OUT      VARCHAR2,
x_msg_count          OUT      NUMBER,
x_msg_data           OUT      VARCHAR2);

```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–48 Update Rate IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_rate_schedule_name	VARCHAR2	Y	Rate schedule name
p_tier_coordinates_tbl	tier_coordinates_tbl	Y	Coordinates of rate tier
p_commission_amount	Number	Y	Commission amount
p_object_version_number	Number	Y	Object version

The following table describes the OUT parameters associated with this API.

Table 1–49 Update Rate OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.9 Create Dimension

Procedure Specification

```
PROCEDURE Create_Dimension
  (p_api_version          IN      NUMBER,
   p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit               IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_name                  IN      CN_RATE_DIMENSIONS.NAME%TYPE,
   p_description           IN      CN_RATE_DIMENSIONS.DESCRPTION%TYPE :=
NULL,
   p_dim_unit_code         IN      CN_RATE_DIMENSIONS.DIM_UNIT_CODE%TYPE,
   p_tiers_tbl             IN      rate_tier_tbl_type := g_miss_rate_tier_
tbl,
   x_return_status         OUT     VARCHAR2,
   x_msg_count             OUT     NUMBER,
   x_msg_data              OUT     VARCHAR2);
```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–50 Create Dimension IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_name	VARCHAR2	Y	Rate dimension name
p_description	VARCHAR2	Y	Description
p_dim_unit_code	VARCHAR2	Y	Dimension unit (AMOUNT of PERCENT)
p_tiers_tbl	Rate_tier_tbl_type	N	Contents of rate tiers (default empty table)

The following table describes the OUT parameters associated with this API.

Table 1–51 Create Dimension OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.10 Create Dimension

Procedure Specification

PROCEDURE Update_Dimension

(p_api_version	IN	NUMBER,
p_init_msg_list	IN	VARCHAR2 := FND_API.G_FALSE,
p_commit	IN	VARCHAR2 := FND_API.G_FALSE,
p_validation_level	IN	NUMBER := FND_API.G_VALID_LEVEL_FULL,
p_original_name	IN	CN_RATE_DIMENSIONS.NAME%TYPE,
p_new_name	IN	CN_RATE_DIMENSIONS.NAME%TYPE :=
		cn_api.g_miss_char,
p_description	IN	CN_RATE_DIMENSIONS.DESCRPTION%TYPE :=
		cn_api.g_miss_char,
p_dim_unit_code	IN	CN_RATE_DIMENSIONS.DIM_UNIT_CODE%TYPE :=
		cn_api.g_miss_char,
p_tiers_tbl	IN	rate_tier_tbl_type :=
		g_miss_rate_tier_tbl,
p_object_version_number	IN	CN_RATE_DIMENSIONS.OBJECT_VERSION_
NUMBER%TYPE,		
x_return_status	OUT	VARCHAR2,
x_msg_count	OUT	NUMBER,
x_msg_data	OUT	VARCHAR2);

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–52 Update Dimension IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_original_name	VARCHAR2	Y	Original rate dimension name
p_new_name	VARCHAR2	Y	New rate dimension name
p_description	VARCHAR2	Y	Description
p_dim_unit_code	VARCHAR2	Y	Dimension unit (AMOUNT of PERCENT)
p_tiers_tbl	Rate_tier_tbl_type	N	Rate tiers (default empty table)
p_object_version_number	NUMBER	Y	Object version

The following table describes the OUT parameters associated with this API.

Table 1–53 Update Dimension OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.11 Delete Dimension

Procedure Specification

```
PROCEDURE Delete_Dimension
  (p_api_version          IN      NUMBER,
   p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit                IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_name                  IN      CN_RATE_DIMENSIONS.NAME%TYPE,
```

```

x_return_status      OUT      VARCHAR2,
x_msg_count          OUT      NUMBER,
x_msg_data           OUT      VARCHAR2);

```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–54 Delete Dimension IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_name	VARCHAR2	Y	Rate dimension name

The following table describes the OUT parameters associated with this API.

Table 1–55 Delete Dimension OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.12 Create Tier

Procedure Specification

```

PROCEDURE Create_Tier
(
  p_api_version          IN      NUMBER,
  p_init_msg_list        IN      VARCHAR2 := FND_API.G_FALSE,
  p_commit               IN      VARCHAR2 := FND_API.G_FALSE,
  p_validation_level     IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  p_dimension_name       IN      CN_RATE_DIMENSIONS.NAME%TYPE,

```

p_value1	IN	VARCHAR2,
p_value2	IN	VARCHAR2,
p_tier_sequence	IN	CN_RATE_DIM_TIERS.TIER_SEQUENCE%TYPE,
x_return_status	OUT	VARCHAR2,
x_msg_count	OUT	NUMBER,
x_msg_data	OUT	VARCHAR2);

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–56 Create Tier IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_dimension_name	VARCHAR2	Y	Rate dimension name
p_value1	VARCHAR2	Y	Lower tier value
p_value2	VARCHAR2	Y	Upper tier value (ignored for string based dimensions)

The following table describes the OUT parameters associated with this API.

Table 1–57 Create Tier OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.13 Update Tier

Procedure Specification

```

PROCEDURE Update_Tier
  (p_api_version           IN      NUMBER,
   p_init_msg_list         IN      VARCHAR2 := FND_API.G_FALSE,
   p_commit                IN      VARCHAR2 := FND_API.G_FALSE,
   p_validation_level      IN      NUMBER   := FND_API.G_VALID_LEVEL_FULL,
   p_dimension_name        IN      CN_RATE_DIMENSIONS.NAME%TYPE,
   p_tier_sequence         IN      CN_RATE_DIM_TIERS.TIER_SEQUENCE%TYPE,
   p_value1                IN      VARCHAR2,
   p_value2                IN      VARCHAR2,
   p_object_version_number IN      CN_RATE_DIM_TIERS.OBJECT_VERSION_
NUMBER%TYPE,
   x_return_status         OUT      VARCHAR2,
   x_msg_count             OUT      NUMBER,
   x_msg_data              OUT      VARCHAR2);

```

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–58 Update Tier IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_dimension_name	VARCHAR2	Y	Rate dimension name
p_tier_sequence	NUMBER	Y	Rate tier sequence
p_value1	VARCHAR2	Y	Lower tier value
p_value2	VARCHAR2	Y	Upper tier value (ignored for string based dimensions)

Table 1–58 Update Tier IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_object_version_number	NUMBER	Y	Object version

The following table describes the OUT parameters associated with this API.

Table 1–59 Update Tier OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.4.14 Delete Tier

Procedure Specification

PROCEDURE Delete_Tier

(p_api_version	IN	NUMBER,
p_init_msg_list	IN	VARCHAR2 := FND_API.G_FALSE,
p_commit	IN	VARCHAR2 := FND_API.G_FALSE,
p_validation_level	IN	NUMBER := FND_API.G_VALID_LEVEL_FULL,
p_dimension_name	IN	CN_RATE_DIMENSIONS.NAME%TYPE,
p_tier_sequence	IN	CN_RATE_DIM_TIERS.TIER_SEQUENCE%TYPE,
x_return_status	OUT	VARCHAR2,
x_msg_count	OUT	NUMBER,
x_msg_data	OUT	VARCHAR2);

Current Version

1.0.

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–60 Delete Tier IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	API version

Table 1–60 Delete Tier IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Initialize message list (default F)
p_commit	VARCHAR2	N	Commit flag (default F).
p_validation_level	VARCHAR2	N	Validation level (default 100).
p_dimension_name	VARCHAR2	Y	Rate dimension name
p_tier_sequence	NUMBER	Y	Rate tier sequence

The following table describes the OUT parameters associated with this API.

Table 1–61 Delete Tier OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Return Status
x_msg_count	NUMBER	Number of messages returned
x_msg_data	VARCHAR2	Contents of message if x_msg_count = 1

1.5 Package CN_ROLE_PLANS_PUB

There are three APIs for CN_CALC_SUBMISSION_PUB.

- **Create Role Plans:** This procedure creates a Role-Plan Assignment. Records are inserted into cn_role_plans. There is a call to procedure cn_srp_plan_assign_pvt.create_srp_plan_assigns. This inserts records into cn_srp_plan_Assigns for all salesreps with the role. This results in records being created in cn_srp_quota_assigns, cn_srp_period_quotas, cn_Srp_periods, cn_srp_rollover_quotas(if exists), cn_Srp_quota_rules, and cn_Srp_rate_assigns.
- **Update Role Plans:** This procedure updates Role-Plan Assignment. Records are updated in cn_role_plans. There is a call to procedure .cn_srp_plan_assign_pvt.update_srp_plan_assigns. This results in update to cn_srp_quota_assigns, cn_srp_rate_tiers, cn_srp_periods, cn_Srp_rollover, quotas.
- **Delete Role Plans:** This procedure deletes the Role-Plan assignment from cn_role_plans. This results in records being deleted from cn_Srp_plan_assigns, cn_Srp_quota_Assigns, and cn_Srp_rate_tiers

1.5.1 Data Structure Specifications

The following data structures are used in Package CN_ROLE_PLANS_PUB:

1.5.1.1 Data Structure

```
TYPE role_plan_rec_type IS RECORD
(
    role_name          cn_roles.name%TYPE          := cn_api.G_MISS_
CHAR,
    comp_plan_name     cn_comp_plans.name%TYPE      := cn_api.G_MISS_CHAR,
    start_date         cn_role_plans.start_date%TYPE := cn_api.G_MISS_DATE,
    end_date           cn_role_plans.end_date%TYPE   := cn_api.G_MISS_DATE,
    attribute_category  cn_role_plans.attribute_category%TYPE := cn_api.G_MISS_
CHAR,
    attribute1         cn_role_plans.attribute1%TYPE := cn_api.G_MISS_CHAR,
    attribute2         cn_role_plans.attribute2%TYPE := cn_api.G_MISS_CHAR,
    attribute3         cn_role_plans.attribute3%TYPE := cn_api.G_MISS_CHAR,
    attribute4         cn_role_plans.attribute4%TYPE := cn_api.G_MISS_CHAR,
    attribute5         cn_role_plans.attribute5%TYPE := cn_api.G_MISS_CHAR,
    attribute6         cn_role_plans.attribute6%TYPE := cn_api.G_MISS_CHAR,
    attribute7         cn_role_plans.attribute7%TYPE := cn_api.G_MISS_CHAR,
    attribute8         cn_role_plans.attribute8%TYPE := cn_api.G_MISS_CHAR,
    attribute9         cn_role_plans.attribute9%TYPE := cn_api.G_MISS_CHAR,
    attribute10        cn_role_plans.attribute10%TYPE := cn_api.G_MISS_CHAR,
    attribute11        cn_role_plans.attribute11%TYPE := cn_api.G_MISS_CHAR,
    attribute12        cn_role_plans.attribute12%TYPE := cn_api.G_MISS_CHAR,
    attribute13        cn_role_plans.attribute13%TYPE := cn_api.G_MISS_CHAR,
    attribute14        cn_role_plans.attribute14%TYPE := cn_api.G_MISS_CHAR,
    attribute15        cn_role_plans.attribute15%TYPE := cn_api.G_MISS_CHAR
)
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–62 *role_plan_rec_type Parameters*

Parameter	Data Type	Description
role_name	VARCHAR2	Role name
Comp_plan_name	VARCHAR2	Plan Name
Start_Date	DATE	Start Date
End_Date	DATE	End Date

Table 1–62 *role_plan_rec_type Parameters*

Parameter	Data Type	Description
Attribute_Category	VARCHAR2	Descriptive flexfield structure defining column
Attribute1	VARCHAR2	Standard Attribute Columns
Attribute2	VARCHAR2	Standard Attribute Columns
Attribute3	VARCHAR2	Standard Attribute Columns
Attribute4	VARCHAR2	Standard Attribute Columns
Attribute5	VARCHAR2	Standard Attribute Columns
Attribute6	VARCHAR2	Standard Attribute Columns
Attribute7	VARCHAR2	Standard Attribute Columns
Attribute8	VARCHAR2	Standard Attribute Columns
Attribute9	VARCHAR2	Standard Attribute Columns
Attribute10	VARCHAR2	Standard Attribute Columns
Attribute11	VARCHAR2	Standard Attribute Columns
Attribute12	VARCHAR2	Standard Attribute Columns
Attribute13	VARCHAR2	Standard Attribute Columns
Attribute14	VARCHAR2	Standard Attribute Columns
Attribute15	VARCHAR2	Standard Attribute Columns

1.5.2 Create Role Plans

This procedure creates a record in `cn_role_plans`. This also calls `cn_Srp_plan_assigns_pvt.create_srp_plan_assigns` to create records in `cn_Srp_plan_assigns`.

Procedure Specification

```

PROCEDURE Create_Role_Plan
(
  p_api_version      IN   NUMBER,
  p_init_msg_list    IN   VARCHAR2 := FND_API.G_FALSE,
  p_commit           IN   VARCHAR2 := FND_API.G_FALSE,
  p_validation_level IN   NUMBER := FND_API.g_valid_level_full,
  x_return_status    OUT  NOCOPY VARCHAR2,
  x_loading_status   OUT  NOCOPY VARCHAR2,
  x_msg_count        OUT  NOCOPY NUMBER,
  x_msg_data         OUT  NOCOPY VARCHAR2,

```

```
p_role_plan_rec          IN    role_plan_rec_type := G_MISS_ROLE_PLAN_REC
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–63 create_role_plans IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_role_plan_rec	role_plan_rec_type	Y	Record of type role_plan_rec_type
p_validation_level	VARCHAR2	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–64 create_role_plans OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.5.3 Update Role Plans

This procedure updates a record in cn_role_plans. This also calls cn_Srp_plan_assigns_pvt. update_srp_plan_assigns to create records in cn_Srp_plan_assigns.

Procedure Specification

```
PROCEDURE Update_Role_Plan
(
    p_api_version          IN    NUMBER,
```

```

    p_init_msg_list          IN      VARCHAR2 := FND_API.G_FALSE,
    p_commit                 IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level       IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_status          OUT NOCOPY VARCHAR2,
    x_loading_status         OUT NOCOPY VARCHAR2,
    x_msg_count              OUT NOCOPY NUMBER,
    x_msg_data               OUT NOCOPY VARCHAR2,
    p_role_plan_rec_old      IN      role_plan_rec_type := G_MISS_ROLE_PLAN_REC,
    p_ovn                    IN      cn_role_plans.object_version_number%TYPE,
    p_role_plan_rec_new      IN      role_plan_rec_type := G_MISS_ROLE_PLAN_REC
);

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–65 *update_role_plans IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_role_plan_rec_old	role_plan_rec_type	Y	Record of type role_plan_rec_type
p_role_plan_rec_new	role_plan_rec_type	Y	Record of type role_plan_rec_type
p_ovn	NUMBER	Y	Object Version Number
p_validation_level	VARCHAR2	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–66 *update_role_plans OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters

Table 1–66 *update_role_plans OUT Parameters*

Parameter	Data Type	Descriptions
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.5.4 Delete Role Plans

This procedure deletes a record in `cn_role_plans`. This also calls `cn_Srp_plan_assigns_pvt.delete_srp_plan_assigns` to create records in `cn_Srp_plan_assigns`.

Procedure Specification

```
PROCEDURE Delete_Role_Plan
(
    p_api_version          IN    NUMBER,
    p_init_msg_list        IN    VARCHAR2 := FND_API.G_FALSE,
    p_commit               IN    VARCHAR2 := FND_API.G_FALSE,
    p_validation_level     IN    NUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_status        OUT   NOCOPY  VARCHAR2,
    x_loading_status       OUT   NOCOPY  VARCHAR2,
    x_msg_count            OUT   NOCOPY  NUMBER,
    x_msg_data             OUT   NOCOPY  VARCHAR2,
    p_role_plan_rec       IN    role_plan_rec_type := G_MISS_ROLE_PLAN_REC
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–67 *delete_role_plans IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_role_plan_rec	role_plan_rec_type	Y	Record of type <code>role_plan_rec_type</code>

Table 1–67 delete_role_plans IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_validation_level	NUMBER	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–68 update_role_plans OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.6 Package CN_SRP_CUSTOMIZE_PUB

This package contains four APIs:

- Update SRP Quota Assigns: This procedure updates cn_srp_quota_assigns. This in turn calls procedure cn_srp_quota_assigns_pkg.update_record.
- Update SRP Quota Rules: This procedure updates cn_srp_quota_rules.
- Update SRP Rule Uplifts: This procedure updates cn_srp_rule_uplifts.
- Change SRP Quota Custom Flag: This procedure calls cn_srp_quota_assigns_pkg.update_record.

1.6.1 Data Structure Specifications

None.

1.6.2 Update SRP Quota Assigns

This procedure updates cn_srp_quota_assigns. This in turn calls procedure cn_srp_quota_assigns_pkg.update_record.

Procedure Specification

```
PROCEDURE Update_srp_quota_assign(
```

p_api_version	IN NUMBER,
p_init_msg_list	IN VARCHAR2,
p_commit	IN VARCHAR2,
p_validation_level	IN NUMBER,
p_srp_quota_assign_id	IN NUMBER,
p_customized_flag	IN VARCHAR2,
p_quota	IN NUMBER,
p_fixed_amount	IN NUMBER,
p_goal	IN NUMBER,
x_return_status	OUT NOCOPY VARCHAR2,
x_msg_count	OUT NOCOPY NUMBER,
x_msg_data	OUT NOCOPY VARCHAR2,
x_loading_status	OUT NOCOPY VARCHAR2,
x_status	OUT NOCOPY VARCHAR2)

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–69 Update_Srp_Quota_Assigns IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_srp_quota_assign_id	NUMBER	Y	Srp_quota_assign_id
p_customized_flag	VARCHAR2	Y	Customized Flag
p_quota	NUMBER	Y	Quota Id
p_fixed_amount	NUMBER	N	Payment_Amount of cn_srp_quota_assigns
p_goal	NUMBER	N	Performance Goal of cn_srp_quota_assigns
p_validation_level	NUMBER	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–70 Update_Srp_Quota_Assigns OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_status	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.6.3 Update SRP Quota Rules

This procedure updates cn_srp_quota_rules.

Procedure Specification

```
PROCEDURE Update_Srp_Quota_Rules (
    p_api_version          IN NUMBER,
    p_init_msg_list        IN VARCHAR2,
    p_commit               IN VARCHAR2,
    p_validation_level     IN NUMBER,
    p_quota_rule_id        IN NUMBER,
    p_srp_quota_rule_id    IN NUMBER,
    p_target               IN NUMBER,
    p_payment_amount       IN NUMBER,
    p_performance_goal     IN NUMBER,
    x_return_status        OUT NOCOPY VARCHAR2,
    x_msg_count            OUT NOCOPY NUMBER,
    x_msg_data             OUT NOCOPY VARCHAR2,
    x_loading_status       OUT NOCOPY VARCHAR2
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–71 Update_Srp_Quota_Rules IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_srp_quota_rule_id	NUMBER	Y	Srp_quota_assign_id
p_quota_rule_id	VARCHAR2	Y	Customized Flag
p_target	NUMBER	Y	Quota Id
p_payment_amount	NUMBER	N	Payment_Amount of cn_srp_quota_assigns
p_performance_goal	NUMBER	N	Performance Goal of cn_srp_quota_assigns
p_validation_level	NUMBER	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–72 Update_Srp_Quota_Rules OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.6.4 Update SRP Rule Uplifts

This procedure updates cn_srp_rule_uplifts.

Procedure Specification

```
PROCEDURE Update_Srp_Rule_Uplifts(  
    p_api_version          IN  NUMBER,  
    p_init_msg_list        IN  VARCHAR2,  
    p_commit               IN  VARCHAR2,  
    p_validation_level     IN  NUMBER,
```

```

p_srp_rule_uplift_id    IN  NUMBER,
p_payment_factor        IN  NUMBER,
p_quota_factor          IN  NUMBER,
x_return_status         OUT  NOCOPY VARCHAR2,
x_msg_count             OUT  NOCOPY NUMBER,
x_msg_data              OUT  NOCOPY VARCHAR2,
x_loading_status        OUT  NOCOPY VARCHAR2
)

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–73 Update_Srp_Rule_Uplifts IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_srp_rule_uplift_id	NUMBER	Y	Srp_rule_uplift_id
p_payment_factor	VARCHAR2	Y	Payment Factor
p_quota_factor	NUMBER	Y	Quota Factor
p_validation_level	NUMBER	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–74 Update_Srp_Rule_Uplifts OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.6.5 Change SRP Quota Custom Flag

This procedure calls cn_srp_quota_assigns_pkg.update_record.

Procedure Specification

```
PROCEDURE Change_srp_quota_custom_flag(  
    p_api_version          IN  NUMBER,  
    p_init_msg_list        IN  VARCHAR2,  
    p_commit               IN  VARCHAR2,  
    p_validation_level     IN  NUMBER,  
    p_srp_quota_assign_id  IN  NUMBER,  
    p_customized_flag      IN  VARCHAR2,  
    x_return_status        OUT NOCOPY VARCHAR2,  
    x_msg_count            OUT NOCOPY NUMBER,  
    x_msg_data             OUT NOCOPY VARCHAR2,  
    x_loading_status       OUT NOCOPY VARCHAR2  
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–75 Change_Srp_Quota_Custom_Flag IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_srp_quota_assign_id	NUMBER	Y	Srp_quota_assign_id
p_customized_flag	VARCHAR2	Y	Customized Flag
p_validation_level	NUMBER	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–76 Change_Srp_Quota_Custom_Flag OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
X_status	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.7 Package CN_COMMISSION_CALC_PUB

The calculate_Commission procedure in cn_commission_calc_pub is used for calculating projected compensation for a salesperson, including the projection identifier, calculation date, and sales credit amount. These details are inserted into a Global Temporary Table called *cn_proj_compensation_gtt* by the calling program. The projection identifier passed by the user is mapped to the plan element using the defined for plan element classification rules defined for the period defined in the *cn_proj_compensation_gtt*. Then the corresponding formula is used for finding the projected compensation of the projection identifier passed.

The projected compensation is calculated only if:

- The passed projection identifier is mapped to a plan element in the ruleset.
- The mapped plan element has got a valid assignment to the resource provided in the *cn_proj_compensation_gtt*.
- The plan element has got a valid formula with forecast expression defined.

1.7.1 Data Structure Specifications

1.7.1.1 Data Structure

None.

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–77 *commission_calc Parameters*

Parameter	Data Type	Description
p_api_version	NUMBER	Standard In Parameter

1.7.2 Calculate Commission

This procedure creates a record in cn_role_plans. This also calls Cn_Commission_Calc_Pvt. calculate_Commission to create records in cn_Srp_plan_assigns.

Procedure Specification

```
Procedure calculate_Commission
(
    p_api_version          IN  NUMBER,
    p_init_msg_list        IN  VARCHAR2 := FND_API.G_FALSE,
    x_inc_plnr_disclaimer  OUT NOCOPY cn_repositories.income_planner_
disclaimer%TYPE,
    x_return_status        OUT NOCOPY VARCHAR2,
    x_msg_count            OUT NOCOPY NUMBER,
    x_msg_data             OUT NOCOPY VARCHAR2
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–78 *commission_calc IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter

The following table describes the OUT parameters associated with this API.

Table 1–79 *commission_calc OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters

Table 1–79 commission_calc OUT Parameters

Parameter	Data Type	Descriptions
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_inc_plnr_disclaimer	VARCHAR2	The income planner disclaimer message is returned if the profile 'CN_CUST_DISCLAIMER' is set.

1.8 Package CN_PLAN_ELEMENT_PUB

Create_Plan_Element in cn_plan_element_pub creates a plan element and creates entries in the child tables. The possible child table entries are for:

- Quota Rules
- Rule Uplifts
- Trx Factors
- Period Quotas
- Roll Over Quotas

Delete_Plan_Element in cn_plan_element_pub deletes a plan element and delete respective entries in the child tables. The possible child table entries are for:

- Quota Rules
- Rule Uplifts
- Trx Factors
- Roll Over Quotas

If the plan element belongs to a compensation plan status is set to Incomplete. If the plan element has been assigned to the salesreps, then the records are also deleted from srp quota assigns, srp rate assigns, srp period quotas, srp quota rules and srp roll over quotas.

Duplicate_Plan_Element in cn_plan_element_pub copies the information of an existing plan element. The child table entries are also copied from the source.

Update_Plan_Element in cn_plan_element_pub updates a plan element and updates respective entries in the child tables. The possible child table entries are for:

- Quota Rules
- Rule Uplifts

- Trx Factors
- Roll Over Quotas

If the plan element belongs to a compensation plan, status is set to Incomplete. If the plan element has been assigned to the salesreps, then the records are also updated in srp quota assigns, srp rate assigns, srp period quotas, srp quota rules and srp role over quotas only if the plan element to salesrep assignment is not customizable.

1.8.1 Data Structure Specifications

Six data structures are used in Package CN_PLAN_ELEMENT_PUB:

1.8.1.1 Data Structure

```
plan_element_rec_type
period_quotas_rec_type
revenue_class_rec_type
rev_uplift_rec_type
trx_factor_rec_type
rt_quota_asgns_rec_type
```

1.8.1.1.1 plan_element_rec_type

```
TYPE plan_element_rec_type IS RECORD
(
    name                cn_quotas.name%TYPE           := cn_api.g_miss_char,
    description          cn_quotas.description%TYPE     := cn_api.g_miss_char,
    period_type          cn_lookups.meaning%TYPE        := cn_api.g_miss_char,
    element_type         cn_lookups.meaning%TYPE        := cn_api.g_miss_char,
    target               cn_quotas.target%TYPE         := cn_api.g_miss_num,
    incentive_type       cn_lookups.meaning%TYPE        := cn_api.g_miss_char,
    credit_type          cn_credit_types.name%TYPE      := cn_api.g_miss_char,
    calc_formula_name    cn_calc_formulas.name%TYPE    := cn_api.g_miss_char,
    rt_sched_custom_flag cn_quotas.rt_sched_custom_flag%TYPE := cn_api.g_miss_
char,
    package_name         cn_quotas.package_name%TYPE   := cn_api.g_miss_char,
    performance_goal     cn_quotas.performance_goal%TYPE:= cn_api.g_miss_num,
    payment_amount       cn_quotas.payment_amount%TYPE := cn_api.g_miss_num,
    start_date           cn_quotas.start_date%TYPE     := cn_api.g_miss_date,
    end_date             cn_quotas.end_date%TYPE       := cn_api.g_miss_date,
    status               cn_quotas.quota_status%TYPE   := cn_api.g_miss_char,
    interval_name        cn_interval_types.name%TYPE   := cn_api.g_miss_char,
    payee_assign_flag    cn_quotas.payee_assign_flag%TYPE:= cn_api.g_miss_char,
```

```

vesting_flag          cn_quotas.vesting_flag%TYPE      := cn_api.g_miss_char,
addup_from_rev_class_flag cn_quotas.addup_from_rev_class_flag%TYPE
:= cn_api.g_miss_char,
expense_account_id    cn_quotas_all.expense_account_id%TYPE:= cn_api.g_miss_
id,
liability_account_id  cn_quotas_all.liability_account_id%TYPE := cn_api.g_
miss_id,
quota_group_code      cn_quotas.quota_group_code%TYPE := cn_api.g_miss_char,
payment_group_code    cn_quotas.payment_group_code%TYPE:= cn_api.g_miss_char,
attribute_category    cn_quotas.attribute_category%TYPE:= cn_api.g_miss_char,
attribute1            cn_quotas.attribute1%TYPE       := cn_api.g_miss_char,
attribute2            cn_quotas.attribute2%TYPE       := cn_api.g_miss_char,
attribute3            cn_quotas.attribute3%TYPE       := cn_api.g_miss_char,
attribute4            cn_quotas.attribute4%TYPE       := cn_api.g_miss_char,
attribute5            cn_quotas.attribute5%TYPE       := cn_api.g_miss_char,
attribute6            cn_quotas.attribute6%TYPE       := cn_api.g_miss_char,
attribute7            cn_quotas.attribute7%TYPE       := cn_api.g_miss_char,
attribute8            cn_quotas.attribute8%TYPE       := cn_api.g_miss_char,
attribute9            cn_quotas.attribute9%TYPE       := cn_api.g_miss_char,
attribute10           cn_quotas.attribute10%TYPE      := cn_api.g_miss_char,
attribute11           cn_quotas.attribute11%TYPE      := cn_api.g_miss_char,
attribute12           cn_quotas.attribute12%TYPE      := cn_api.g_miss_char,
attribute13           cn_quotas.attribute13%TYPE      := cn_api.g_miss_char,
attribute14           cn_quotas.attribute14%TYPE      := cn_api.g_miss_char,
attribute15           cn_quotas.attribute15%TYPE      := cn_api.g_miss_char
)

```

1.8.1.1.2 period_quotas_rec_type

TYPE period_quotas_rec_type IS RECORD

```

( period_name          cn_periods.period_name%TYPE      := cn_api.g_miss_char,
period_target         cn_period_quotas.period_target%TYPE := cn_api.g_miss_num,
period_payment        cn_period_quotas.period_payment%TYPE:= cn_api.g_miss_num,
performance_goal      cn_period_quotas.performance_goal%TYPE:= cn_api.g_miss_
num,
attribute1            cn_period_quotas.attribute1%TYPE := cn_api.g_miss_char,
attribute2            cn_period_quotas.attribute2%TYPE := cn_api.g_miss_char,
attribute3            cn_period_quotas.attribute3%TYPE := cn_api.g_miss_char,
attribute4            cn_period_quotas.attribute4%TYPE := cn_api.g_miss_char,
attribute5            cn_period_quotas.attribute5%TYPE := cn_api.g_miss_char,
attribute6            cn_period_quotas.attribute6%TYPE := cn_api.g_miss_char,
attribute7            cn_period_quotas.attribute7%TYPE := cn_api.g_miss_char,
attribute8            cn_period_quotas.attribute8%TYPE := cn_api.g_miss_char,
attribute9            cn_period_quotas.attribute9%TYPE := cn_api.g_miss_char,
attribute10           cn_period_quotas.attribute10%TYPE := cn_api.g_miss_char,

```

```

        attribute11      cn_period_quotas.attribute11%TYPE := cn_api.g_miss_char,
        attribute12      cn_period_quotas.attribute12%TYPE := cn_api.g_miss_char,
        attribute13      cn_period_quotas.attribute13%TYPE := cn_api.g_miss_char,
        attribute14      cn_period_quotas.attribute14%TYPE := cn_api.g_miss_char,
        attribute15      cn_period_quotas.attribute15%TYPE := cn_api.g_miss_char,
        period_name_old  cn_periods.period_name%TYPE      := cn_api.g_miss_char
    )

```

1.8.1.1.3 revenue_class_rec_type

```

TYPE revenue_class_rec_type IS RECORD
(
    rev_class_name      cn_quota_rules.name%TYPE      := cn_api.g_miss_char,
    rev_class_target    cn_quota_rules.target%TYPE    := cn_api.g_miss_num,
    rev_class_payment_amount cn_quota_rules.payment_amount%TYPE:= cn_api.g_miss_
num,
    rev_class_performance_goal cn_quota_rules.performance_goal%TYPE:= cn_api.g_
miss_num,
    description        cn_quota_rules.description%TYPE := cn_api.g_miss_char,
    attribute_category cn_quotas.attribute_category%TYPE := cn_api.g_miss_char,
    attribute1         cn_quotas.attribute1%TYPE      := cn_api.g_miss_char,
    attribute2         cn_quotas.attribute2%TYPE      := cn_api.g_miss_char,
    attribute3         cn_quotas.attribute3%TYPE      := cn_api.g_miss_char,
    attribute4         cn_quotas.attribute4%TYPE      := cn_api.g_miss_char,
    attribute5         cn_quotas.attribute5%TYPE      := cn_api.g_miss_char,
    attribute6         cn_quotas.attribute6%TYPE      := cn_api.g_miss_char,
    attribute7         cn_quotas.attribute7%TYPE      := cn_api.g_miss_char,
    attribute8         cn_quotas.attribute8%TYPE      := cn_api.g_miss_char,
    attribute9         cn_quotas.attribute9%TYPE      := cn_api.g_miss_char,
    attribute10        cn_quotas.attribute10%TYPE     := cn_api.g_miss_char,
    attribute11        cn_quotas.attribute11%TYPE     := cn_api.g_miss_char,
    attribute12        cn_quotas.attribute12%TYPE     := cn_api.g_miss_char,
    attribute13        cn_quotas.attribute13%TYPE     := cn_api.g_miss_char,
    attribute14        cn_quotas.attribute14%TYPE     := cn_api.g_miss_char,
    attribute15        cn_quotas.attribute15%TYPE     := cn_api.g_miss_char,
    rev_class_name_old cn_quota_rules.name%TYPE      := cn_api.g_miss_char
)

```

1.8.1.1.4 rev_uplift_rec_type

```

TYPE rev_uplift_rec_type IS RECORD
(
    rev_class_name      cn_quota_rules.name%TYPE := cn_api.g_miss_char,
    start_date         cn_quota_rule_uplifts.start_date%TYPE:= cn_api.g_miss_date,
    end_date           cn_quota_rule_uplifts.end_date%TYPE:= cn_api.g_miss_date,
    rev_class_payment_uplift NUMBER              := cn_api.g_miss_num,

```

```

    rev_class_quota_uplift    NUMBER                               := cn_api.g_miss_num,
    attribute_category        cn_quota_rule_uplifts.attribute_category%TYPE
:=cn_api.g_miss_char,
    attribute_category        cn_quotas.attribute_category%TYPE:= cn_api.g_miss_char,
    attribute1                cn_quota_rule_uplifts.attribute1%TYPE:= cn_api.g_miss_char,
    attribute2                cn_quota_rule_uplifts.attribute2%TYPE:= cn_api.g_miss_char,
    attribute3                cn_quota_rule_uplifts.attribute3%TYPE:= cn_api.g_miss_char,
    attribute4                cn_quota_rule_uplifts.attribute4%TYPE:= cn_api.g_miss_char,
    attribute5                cn_quota_rule_uplifts.attribute5%TYPE:= cn_api.g_miss_char,
    attribute6                cn_quota_rule_uplifts.attribute6%TYPE:= cn_api.g_miss_char,
    attribute7                cn_quota_rule_uplifts.attribute7%TYPE:= cn_api.g_miss_char,
    attribute8                cn_quota_rule_uplifts.attribute8%TYPE:= cn_api.g_miss_char,
    attribute9                cn_quota_rule_uplifts.attribute9%TYPE:= cn_api.g_miss_char,
    attribute10               cn_quota_rule_uplifts.attribute10%TYPE:= cn_api.g_miss_char,
    attribute11               cn_quota_rule_uplifts.attribute11%TYPE:= cn_api.g_miss_char,
    attribute12               cn_quota_rule_uplifts.attribute12%TYPE:= cn_api.g_miss_char,
    attribute13               cn_quota_rule_uplifts.attribute13%TYPE:= cn_api.g_miss_char,
    attribute14               cn_quota_rule_uplifts.attribute14%TYPE:= cn_api.g_miss_char,
    attribute15               cn_quota_rule_uplifts.attribute14%TYPE:= cn_api.g_miss_char,
    rev_class_name_old        cn_quota_rules.name%TYPE           := cn_api.g_miss_char,
    start_date_old            cn_quota_rule_uplifts.start_date%TYPE:= cn_api.g_miss_date,
    end_date_old              cn_quota_rule_uplifts.end_date%TYPE:= cn_api.g_miss_date
)

```

1.8.1.15 trx_factor_rec_type

TYPE trx_factor_rec_type IS RECORD

```

(
    trx_type                cn_trx_factors.trx_type%TYPE          := cn_api.g_miss_char,
    event_factor            cn_trx_factors.event_factor%TYPE     := cn_api.g_miss_num,
    rev_class_name          cn_quota_rules.name%TYPE              := cn_api.g_miss_char
)

```

1.8.1.16 rt_quota_asgns_rec_type

TYPE rt_quota_asgns_rec_type IS RECORD

```

(
    rate_schedule_name      cn_rate_schedules.name%TYPE          := cn_api.g_miss_char,
    calc_formula_name       cn_calc_formulas.name%TYPE           := cn_api.g_miss_char,
    start_date              cn_rt_Quota_asgns.start_date%TYPE    := cn_api.g_miss_date,
    end_date                cn_rt_quota_asgns.end_date%TYPE      := cn_api.g_miss_date,
    attribute_category       cn_quotas.attribute_category%TYPE   := cn_api.g_miss_char,
    attribute1              cn_quotas.attribute1%TYPE            := cn_api.g_miss_char,
    attribute2              cn_quotas.attribute2%TYPE            := cn_api.g_miss_char,
    attribute3              cn_quotas.attribute3%TYPE            := cn_api.g_miss_char,

```

```
attribute4      cn_quotas.attribute4%TYPE      := cn_api.g_miss_char,
attribute5      cn_quotas.attribute5%TYPE      := cn_api.g_miss_char,
attribute6      cn_quotas.attribute6%TYPE      := cn_api.g_miss_char,
attribute7      cn_quotas.attribute7%TYPE      := cn_api.g_miss_char,
attribute8      cn_quotas.attribute8%TYPE      := cn_api.g_miss_char,
attribute9      cn_quotas.attribute9%TYPE      := cn_api.g_miss_char,
attribute10     cn_quotas.attribute10%TYPE     := cn_api.g_miss_char,
attribute11     cn_quotas.attribute11%TYPE     := cn_api.g_miss_char,
attribute12     cn_quotas.attribute12%TYPE     := cn_api.g_miss_char,
attribute13     cn_quotas.attribute13%TYPE     := cn_api.g_miss_char,
attribute14     cn_quotas.attribute14%TYPE     := cn_api.g_miss_char,
attribute15     cn_quotas.attribute15%TYPE     := cn_api.g_miss_char,
rate_schedule_name_old cn_rate_schedules.name%TYPE := cn_api.g_miss_char,
start_date_old  cn_rt_Quota_asgns.start_date%TYPE := cn_api.g_miss_date,
end_date_old    cn_rt_quota_asgns.end_date%TYPE  := cn_api.g_miss_date
)
```

1.8.2 Create Plan Element

This procedure creates a new plan element. The procedure calls the following external procedures also:

- cn_quotas_pkg.begin_record
- CN_PERIOD_QUOTAS_GRP.Create_period_quotas
- cn_rt_quota_asgns_pvt.create_rt_quota_asgns
- CN_QUOTA_RULES_GRP.Create_Quota_rules
- CN_API.get_rev_class_id
- CN_CHK_PLAN_ELEMENT_PKG.get_quota_rule_id
- CN_CHK_PLAN_ELEMENT_PKG.chk_trx_factor
- cn_quota_rule_uplifts_grp.Create_Quota_rule_uplift

Procedure Specification

PROCEDURE Create_Plan_Element

```
(
  p_api_version          IN  NUMBER:= CN_API.G_MISS_NUM,
  p_init_msg_list        IN  VARCHAR2:= CN_API.G_FALSE,
  p_commit               IN  VARCHAR2:= CN_API.G_FALSE,
  p_validation_level     IN  NUMBER:= CN_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count            OUT NOCOPY NUMBER,
```

```

x_msg_data                OUT  NOCOPY VARCHAR2,
p_plan_element_rec        IN   plan_element_rec_type
:= G_MISS_PLAN_ELEMENT_REC,
p_revenue_class_rec_tbl   IN   revenue_class_rec_tbl_type
:= G_MISS_REVENUE_CLASS_REC_TBL,
p_rev_uplift_rec_tbl      IN   rev_uplift_rec_tbl_type
:= G_MISS_REV_UPLIFT_REC_TBL,
p_trx_factor_rec_tbl      IN   trx_factor_rec_tbl_type
:= G_MISS_TRX_FACTOR_REC_TBL,
p_period_quotas_rec_tbl   IN   period_quotas_rec_tbl_type
:= g_miss_period_quotas_rec_tbl,
p_rt_quota_asgns_rec_tbl  IN   rt_quota_asgns_rec_tbl_type
:=g_miss_rt_quota_asgns_rec_tbl,
x_loading_status          OUT  NOCOPY  VARCHAR2
)

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–80 Create_Plan_Element IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_validation_level	NUMBER	Y	Standard IN parameter
p_plan_element_rec	plan_element_rec_type	Y	Plan element details
p_revenue_class_rec_tbl	rev_class_rec_tbl_type	N	Revenue class details
p_rev_uplift_rec_tbl	rev_uplift_rec_tbl_type	N	Revenue class uplift factor details
p_trx_factor_rec_tbl	trx_factor_rec_tbl_type	N	Transaction factors details

Table 1–80 Create_Plan_Element IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_period_quotas_rec_tbl	period_quotas_rec_tbl_type	N	Period Quotas details
p_rt_quota_asgns_rec_tbl	rt_quota_asgns_rec_tbl_type	N	Rate quota assigns details

The following table describes the OUT parameters associated with this API.

Table 1–81 Create_Plan_Element OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.8.3 Delete Plan Elements

This procedure deletes an existing plan element. This API calls the following external procedures also:

- cn_chk_plan_element_pkg.get_quota_id
- cn_quota_rule_uplifts_grp.Delete_Quota_rule_uplift
- cn_rt_quota_asgns_pvt.delete_rt_quota_asgns
- CN_quota_rules_GRP.delete_quota_rules
- cn_quotas_pkg.delete_record

Procedure Specification

```
PROCEDURE Delete_Plan_Element
(
    p_api_version           IN  NUMBER      := CN_API.G_MISS_NUM,
    p_init_msg_list         IN  VARCHAR2    := CN_API.G_FALSE,
    p_commit                IN  VARCHAR2    := CN_API.G_FALSE,
    p_validation_level       IN  NUMBER      := CN_API.G_VALID_LEVEL_FULL,
    x_return_status         OUT NOCOPY VARCHAR2,
    x_msg_count             OUT NOCOPY NUMBER,
    x_msg_data              OUT NOCOPY VARCHAR2,
```

```

p_quota_name          IN  VARCHAR2 := CN_API.G_MISS_CHAR,
p_revenue_class_rec_tbl IN revenue_class_rec_tbl_type
                        := g_miss_revenue_class_rec_tbl,
p_rev_uplift_rec_tbl   IN  rev_uplift_rec_tbl_type
                        := g_miss_rev_uplift_rec_tbl,
p_rt_quota_asgns_rec_tbl IN rt_quota_asgns_rec_tbl_type
                        := g_miss_rt_quota_asgns_rec_tbl,
x_loading_status       OUT  NOCOPY VARCHAR2
)

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–82 Delete_Plan_Elements IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_validation_level	NUMBER	Y	Standard IN parameter
p_revenue_class_rec_tbl	revenue_class_rec_tbl_type	N	Revenue class details
p_rev_uplift_rec_tbl	rev_uplift_rec_tbl_type	N	Revenue class uplift factor details
p_rt_quota_asgns_rec_tbl	rt_quota_asgns_rec_tbl_type	N	Rate quota assigns details

The following table describes the OUT parameters associated with this API.

Table 1–83 Delete_Plan_Elements OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters

Table 1–83 Delete_Plan_Elements OUT Parameters

Parameter	Data Type	Descriptions
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters

1.8.4 Duplicate Plan Element

This procedure creates a new plan element which is a copy of an existing plan element. This calls the following external procedures also:

- cn_quotas_pkg.begin_record
- CN_PERIOD_QUOTAS_GRP.Create_period_quotas
- cn_rt_quota_asgns_pvt.create_rt_quota_asgns
- CN_QUOTA_RULES_GRP.Create_Quota_rules
- CN_API.get_rev_class_id
- CN_CHK_PLAN_ELEMENT_PKG.get_quota_rule_id
- CN_CHK_PLAN_ELEMENT_PKG.chk_trx_factor
- cn_quota_rule_uplifts_grp.Create_Quota_rule_uplift

Procedure Specification

```
PROCEDURE Duplicate_Plan_Element
(
    p_api_version          IN  NUMBER      := CN_API.G_MISS_NUM,
    p_init_msg_list        IN  VARCHAR2    := CN_API.G_FALSE,
    p_commit               IN  VARCHAR2    := CN_API.G_FALSE,
    p_validation_level     IN  NUMBER      := CN_API.G_VALID_LEVEL_FULL,
    x_return_status        OUT NOCOPY VARCHAR2,
    x_msg_count            OUT NOCOPY NUMBER,
    x_msg_data             OUT NOCOPY VARCHAR2,
    p_plan_element_name    IN   cn_quotas.name%TYPE := CN_API.G_MISS_CHAR,
    x_plan_element_name    OUT NOCOPY  cn_quotas.name%TYPE,
    x_loading_status       OUT NOCOPY  VARCHAR2
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–84 Duplicate_Plan_Element IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_validation_level	NUMBER	Y	Standard IN parameter
p_plan_element_name	VARCHAR2	Y	Plan element name

The following table describes the OUT parameters associated with this API.

Table 1–85 Duplicate_Plan_Element OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters
x_plan_element_name	VARCHAR2	Plan element name

1.8.5 Update Plan Element

This procedure creates a new plan element. This calls the following external procedures also:

- cn_quotas_pkg.begin_record
- cn_period_quotas_pkg.delete_record
- cn_period_quotas_pkg.distribute_target(p_pe_rec.quota_id);
- CN_PERIOD_QUOTAS_GRP.Update_period_quotas
- CN_SRP_PERIOD_QUOTAS_PKG.populate_srp_period_quotas_ext

- cn_rt_quota_asgns_pkg.insert_record
- cn_rt_quota_asgns_pvt.Update_rt_quota_asgns
- CN_QUOTA_RULES_GRP.Update_Quota_rules
- CN_CHK_PLAN_ELEMENT_PKG.chk_trx_factor
- cn_quota_rule_uplifts_grp.Update_Quota_rule_uplift

Procedure Specification

```
PROCEDURE Update_Plan_Element
(
    p_api_version           IN  NUMBER      := CN_API.G_MISS_NUM,
    p_init_msg_list        IN  VARCHAR2    := CN_API.G_FALSE,
    p_commit                IN  VARCHAR2    := CN_API.G_FALSE,
    p_validation_level      IN  NUMBER      := CN_API.G_VALID_LEVEL_FULL,
    x_return_status         OUT NOCOPY VARCHAR2,
    x_msg_count             OUT NOCOPY NUMBER,
    x_msg_data             OUT NOCOPY VARCHAR2,
    p_new_plan_element_rec  IN  plan_element_rec_type
                                := g_miss_plan_element_rec,
    p_quota_name_old        IN  VARCHAR2,
    p_revenue_class_rec_tbl IN  revenue_class_rec_tbl_type
:= G_MISS_REVENUE_CLASS_REC_TBL,
    p_rev_uplift_rec_tbl    IN  rev_uplift_rec_tbl_type
:= G_MISS_REV_UPLIFT_REC_TBL,
    p_trx_factor_rec_tbl    IN  trx_factor_rec_tbl_type
                                := G_MISS_TRX_FACTOR_REC_TBL,
    p_period_quotas_rec_tbl IN  period_quotas_rec_tbl_type
                                := g_miss_period_quotas_rec_tbl,
    p_rt_quota_asgns_rec_tbl IN rt_quota_asgns_rec_tbl_type
:= g_miss_rt_quota_asgns_rec_tbl,
    x_loading_status        OUT NOCOPY VARCHAR2
)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–86 Update_Plan_Element IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
p_commit	VARCHAR2	Y	Standard IN parameter
p_validation_level	NUMBER	Y	Standard IN parameter
p_plan_element_rec	plan_element_rec_type	Y	Plan element details
p_revenue_class_rec_tbl	revenue_class_rec_tbl_type	N	Revenue class details
p_rev_uplift_rec_tbl	rev_uplift_rec_tbl_type	N	Revenue class uplift factor details
p_trx_factor_rec_tbl	trx_factor_rec_tbl_type	N	Transaction factors details
p_period_quotas_rec_tbl	period_quotas_rec_tbl_type	N	Period quotas details
p_rt_quota_asgns_rec_tbl	rt_quota_asgns_rec_tbl_type	N	Rate quota assigns details
p_quota_name_old	VARCHAR	Y	Old plan element name that needs to be updated.

The following table describes the OUT parameters associated with this API.

Table 1–87 Update_Plan_Element OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters
x_loading_status	VARCHAR2	Standard OUT parameters
x_plan_element_name	VARCHAR2	Plan element name

1.9 Package CN_SCA_CREDITS_BATCH_PUB

This package contains one API:

- Get Sales Credits

This package is accessed by users of the Sales Credit Allocation Module via the concurrent program. The package takes start and end dates and the Transaction Source as input and processes the transactions available in the Interface tables. After completing the processing it populates the results in the output Interface tables.

1.9.1 Data Structure Specifications

There are no data structures used.

1.9.2 Get Sales Credits

The GET_SALES_CREDITS procedure in CN_SCA_CREDITS_BATCH_PUB is used for determining the distribution of sales credit allocation percentages among the different resources and role combinations who took part in the sales transaction. The transactions are loaded into the CN_SCA_HEADERS_INTERFACE table before calling this procedure. This procedure processes the transactions and identifies the Sales Credit Rules based on the attribute information on the transaction and results are populated into the CN_SCA_OUTPUT_LINES table.

Procedure Specification

```
PROCEDURE get_sales_credits (  
    errbuf          OUT  NOCOPY  VARCHAR2,  
    retcode         OUT  NOCOPY  NUMBER,  
    p_transaction_source  IN  VARCHAR2,  
    p_start_date    IN  VARCHAR2,  
    p_end_date      IN  VARCHAR2);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–88 Get_Sales_Credits IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_transaction_source	VARCHAR2	Y	The Sales Credit Allocation module supports multiple transaction sources. This parameter is used as a filter to identify the transactions to be processed by the Rules Engine.
p_start_date	VARCHAR2	Y	The PROCESSED_DATE of the transaction should be greater than or equal to this input parameter.
p_end_date	VARCHAR2	Y	The PROCESSED_DATE of the transaction should be less than or equal to this parameter.

The following table describes the OUT parameters associated with this API.

Table 1–89 Get_Sales_Credits OUT Parameters

Parameter	Data Type	Descriptions
errbug	VARCHAR2	Standard OUT parameters
retcode	NUMBER	Standard OUT parameters

1.10 Package CN_SCA_CREDITS_ONLINE_PUB

This package contains one API:

- get sales credits

This package is accessed by the users of the Sales Credit Allocation module as an online API interface to the SCA Credit Rules engine. The package implements procedures which accept sales transaction information via global temporary tables and return back sales credit allocation percentages for the resources who participate in the transaction via the Global temporary table cn_sca_lines_output_gtt.

1.10.1 Data Structure Specifications

There are no data structures used.

1.10.2 Get Sales Credits

The `get_sales_credits` procedure in `cn_sca_credits_online_pub` is used for determining the distribution of sales credit allocation percentages among the different resources and role combinations who took part in the sales transaction. The information about transaction is input to the API via a Global Temporary table `cn_sca_headers_interface_gtt`. The information about the roles and resources who took part in the transaction is input via the global temporary table `cn_sca_lines_interface_gtt`. The API determines the revenue/nonrevenue allocation percentages based on the Credit Rules setup and outputs the allocation percentage information.

Procedure Specification

```
PROCEDURE get_sales_credits(  
    p_api_version          IN    number,  
    p_init_msg_list        IN    varchar2 := fnd_api.g_false,  
    x_batch_id             IN    number,  
    x_return_status        OUT   NOCOPY varchar2,  
    x_msg_count            OUT   NOCOPY number,  
    x_msg_data             OUT   NOCOPY varchar2)
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–90 *get_sales_credits IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameter
p_init_msg_list	VARCHAR2	Y	Standard IN parameter
x_batch_id	NUMBER	Y	Unique number which identifies the different batches of transactions sent to the online API for processing. Should be taken from the sequence CN_SCA_BATCH_S.

The following table describes the OUT parameters associated with this API.

Table 1–91 *get_sales_credits OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters
x_msg_count	NUMBER	Standard OUT parameters
x_msg_data	VARCHAR2	Standard OUT parameters

1.11 Package CN_SCA_WF_PKG

This package includes public APIs for Sales Credit Allocation standard Revenue Distribution and Transaction Transfer.

- START_PROCESS (1)
- START_PROCESS (2)
- START_PROCESS (3)

1.11.1 Data Structure Specifications

There are no data structures used.

1.11.2 START_PROCESS (1)

This procedure creates a workflow process with the given specifications that distributes revenue between transactions processed in Batch mode.

Procedure Specification

```
PROCEDURE START_PROCESS (  
    p_sca_batch_id      IN number,  
    p_wf_process        IN varchar2,  
    p_wf_item_type      IN varchar2  
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–92 START_PROCESS (1) IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_sca_batch_id	NUMBER	Y	Sales Credit Allocation Batch identifier. Foreign key to CN_SCA_HEADERS_INTERFACE.
p_wf_process	VARCHAR2	Y	Sales Credit Allocation Workflow Revenue Distribution process identifier ('CN_SCA_REV_DIST_PR').
p_wf_item_type	VARCHAR2	Y	Sales Credit Allocation Workflow item identifier ('CN_SCARPR').

There are no OUT parameters associated with this API.

1.11.3 START_PROCESS (2)

This procedure creates a workflow process with the given specifications that distributes revenue between transactions processed in Online mode.

Procedure Specification

```
PROCEDURE START_PROCESS (  
    p_start_header_id    IN number,  
    p_end_header_id      IN number,  
    p_trx_source         IN varchar2,  
    p_wf_process         IN varchar2,  
    p_wf_item_type       IN varchar2  
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–93 START_PROCESS (2) IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_start_header_id	NUMBER	Y	Minimum identifier of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE.

Table 1–93 START_PROCESS (2) IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_end_header_id	NUMBER	Y	Maximum identifier of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE.
p_trx_source	VARCHAR2	Y	Transaction source of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE.
p_wf_process	VARCHAR2	Y	Sales Credit Allocation Workflow Revenue Distribution process identifier ('CN_SCA_REV_DIST_PR').
p_wf_item_type	VARCHAR2	Y	Sales Credit Allocation Workflow item identifier ('CN_SCARPR').

There are no OUT parameters associated with this API.

1.11.4 START_PROCESS (3)

This procedure creates a deferred workflow process with the given specifications that transfers transactions from the Sales Credit Allocation batch interface tables to the Oracle Incentive Compensation Transaction Interface table, or to custom table(s).

Procedure Specification

```
PROCEDURE START_PROCESS (
    p_start_date      IN date,
    p_end_date        IN date,
    p_trx_source       IN varchar2,
    p_wf_process       IN varchar2,
    p_wf_item_type     IN varchar2,
    x_wf_item_key      OUT NOCOPY varchar2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–94 START_PROCESS (3) IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_start_date	DATE	Y	Minimum date of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE_GTT.
p_end_date	DATE	Y	Maximum date of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE_GTT.
p_trx_source	VARCHAR2	Y	Transaction source of transaction headers that are processed. Foreign key to CN_SCA_HEADERS_INTERFACE_GTT.
p_wf_process	VARCHAR2	Y	Sales Credit Allocation Workflow Revenue Distribution process identifier ('CN_SCA_REV_DIST_PR').
p_wf_item_type	VARCHAR2	Y	Sales Credit Allocation Workflow item identifier ('CN_SCARPR').

The following table describes the OUT parameters associated with this API.

Table 1–95 START_PROCESS (3) OUT Parameters

Parameter	Data Type	Descriptions
x_wf_item_key	VARCHAR2	Workflow item key of the Transaction Transfer process.

1.12 Package CN_SCA_WF_CUST_PKG

This package includes public APIs for Sales Credit Allocation custom Revenue Distribution and Transaction Transfer.

- CUST_REV_DIST
- CUST_TRX_LOAD

1.12.1 Data Structure Specifications

There are no data structures used.

1.12.2 CUST_REV_DIST

This procedure is an empty stub that allows you to implement a custom revenue distribution process.

Procedure Specification

```
PROCEDURE CUST_REV_DIST (
    itemType    IN  varchar2,
    itemKey     IN  varchar2,
    actId       IN  number,
    funcMode    IN  varchar2,
    resultOut    OUT NOCOPY varchar2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–96 CUST_REV_DIST IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
itemType	VARCHAR2	Y	Sales Credit Allocation Workflow item type ('CNSCARPR').
itemKey	VARCHAR2	Y	Workflow item key generated by START_PROCESS procedure.
actId	NUMBER	Y	Workflow function activity (instance ID) generated by Workflow.
funcMode	VARCHAR2	Y	Workflow execution mode, generated by Workflow.

The following table describes the OUT parameters associated with this API.

Table 1–97 CUST_REV_DIST OUT Parameters

Parameter	Data Type	Descriptions
resultOut	VARCHAR2	Workflow function result code.

1.12.3 CUST_REV_DIST

This procedure is an empty stub that allows you to implement a custom transfer of transactions from the Sales Credit Allocation batch interface tables to custom table(s).

Procedure Specification

```
PROCEDURE CUST_TRX_LOAD (  
    itemType    IN  varchar2,  
    itemKey     IN  varchar2,  
    actId       IN  number,  
    funcMode    IN  varchar2,  
    resultOut    OUT NOCOPY varchar2  
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–98 CUST_TRX_LOAD IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
itemType	VARCHAR2	Y	Sales Credit Allocation Workflow item type ('CNSCARPR').
itemKey	VARCHAR2	Y	Workflow item key generated by START_PROCESS procedure.
actId	NUMBER	Y	Workflow function activity (instance ID) generated by Workflow.
funcMode	VARCHAR2	Y	Workflow execution mode, generated by Workflow.

The following table describes the OUT parameters associated with this API.

Table 1–99 CUST_TRX_LOAD OUT Parameters

Parameter	Data Type	Descriptions
resultOut	VARCHAR2	Workflow function result code.

1.13 Package CN_POST_COLLECTION_TAE_PUB

This public package integrates the Oracle Incentive Compensation collection process with the Territory Assignment Engine.

- Get_Assignments

1.13.1 Data Structure Specifications

There are no data structures used.

1.13.2 Get_Assignments

This procedure allows you to add your code to populate the attribute data into the TAE input interface table, make the TAE calls to process the territory assignment and update the original OIC transactions with the new territory resource information.

Procedure Specification

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

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–100 *get_assignments IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	Y	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.

Table 1–100 *get_assignments IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_validation_level	NUMBER	Y	Standard IN parameters.

The following table describes the OUT parameters associated with this API.

Table 1–101 *get_assignments OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2	Standard OUT parameters.

1.14 Package CN_PROCESS_TAE_TRX_PUB

This public package populates results from the Territory Assignment Engine into Oracle Incentive Compensation transactions tables.

- Process_Trx_Records

1.14.1 Data Structure Specifications

There are no data structures used.

1.14.2 Process_Trx_Records

This procedure reads the territory resource from the TAE output table and populates the allocated resource information back to the OIC transaction interface table.

Procedure Specification

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

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–102 Process_Trx_Records IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	Y	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	Y	Standard IN parameters.

The following table describes the OUT parameters associated with this API.

Table 1–103 Process_Trx_Records OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2	Standard OUT parameters.

1.15 Package CN_PRD_QUOTA_PUB

The procedure in this package can be used to distribute the target of the specified plan element across the various periods. When the plan element is created, the target is not distributed by default. While the UI can be used to distribute the target by navigating to the individual plan elements, this package provides a convenient method to do the same by using SQL*PLUS.

- Distribute_Prd_Quota

1.15.1 Data Structure Specifications

The following data structures are used in Package CN_PRD_Quota_PUB:

- PRD_QUOTA_REC_TYPE

■ PRD_QUOTA_TBL_TYPE

1.15.1.1 PRD_QUOTA_REC_TYPE

```
TYPE prd_quota_rec_type IS RECORD
(
    PERIOD_NAME          CN_PERIOD_STATUSES.PERIOD_NAME%TYPE := FND_API.G_MISS_CHAR,
    PERIOD_TARGET        NUMBER := FND_API.G_MISS_NUM,
    PERIOD_PAYMENT       NUMBER := FND_API.G_MISS_NUM,
    PERFORMANCE_GOAL     NUMBER := FND_API.G_MISS_NUM
);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–104 Parameters

Parameter	Data Type	Description
PERIOD_NAME	CN_PERIOD_STATUSES.PERIOD_NAME%TYPE	The period for which the target is applicable.
PERIOD_TARGET	NUMBER	The target to be assigned to this particular period.
PERIOD_PAYMENT	NUMBER	The fixed amount (if any) to be assigned to this particular period.
PERFORMANCE_GOAL	NUMBER	The performance goal (if any) to be set for this period.

1.15.1.2 PRD_QUOTA_TBL_TYPE

```
TYPE prd_quota_tbl_type IS
TABLE OF prd_quota_rec_type INDEX BY BINARY_INTEGER ;
```

This data structure is derived from PRD_QUOTA_REC_TYPE.

1.15.2 Distribute_Prd_Quota

This API is used to distribute the target across the periods of a plan element. A mandatory parameter allows the user to choose between an even distribution of the target across the periods and a user configurable distribution as passed in by the user via the table type data structure.

Procedure Specification

PROCEDURE Distribute_Prd_Quota

```
(
    p_api_version          IN    NUMBER      := CN_API.G_MISS_NUM,
    p_init_msg_list        IN    VARCHAR2    := CN_API.G_FALSE,
    p_commit               IN    VARCHAR2    := CN_API.G_FALSE,
    p_validation_level     IN    NUMBER      := CN_API.G_VALID_LEVEL_FULL,
    p_pe_name              IN    CN_QUOTAS.NAME%TYPE,
    p_target_amount        IN    CN_QUOTAS.target%TYPE,
    p_fixed_amount         IN    CN_QUOTAS.payment_amount%TYPE,
    p_performance_goal     IN    CN_QUOTAS.performance_goal%TYPE,
    p_even_distribute      IN    VARCHAR2,
    p_prd_quota_tbl        IN    prd_quota_tbl_type,
    x_return_status        OUT NOCOPY VARCHAR2,
    x_msg_count            OUT NOCOPY NUMBER,
    x_msg_data             OUT NOCOPY VARCHAR2

);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–105 Distribute_Prd_Quota IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	Y	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_pe_name	VARCHAR2	Y	The plan element for which the target has to be distributed.
p_target_amount	NUMBER	Y	The target amount that has to be distributed across the periods.
p_fixed_amount	NUMBER	Y	The fixed amount that has to be distributed for the periods.

Table 1–105 Distribute_Prd_Quota IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_performance_goal	NUMBER	Y	The performance goal that has to be distributed for the periods.
p_even_distribute	VARCHAR2	Y	The configuration parameter used to choose between even distribution and user configurable distribution. Y = even, N = user configurable.
p_prd_quota_tbl	prd_quota_tbl_type	Y	The user configured distribution if p_even_distribute is set to N.

The following table describes the OUT parameters associated with this API.

Table 1–106 Distribute_Prd_Quota OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2	Standard OUT parameters.

1.16 Package CN_SRP_PRD_QUOTA_PUB

The procedure(s) in this package can be used to distribute the target of the specified plan element across the various periods. When the plan element is created, the target is not distributed by default. While the UI can be used to distribute the target by navigating to the individual plan elements, this package provides a convenient method to do the same via SQL*PLUS.

- Distribute_Srp_Prd_Quota

1.16.1 Data Structure Specifications

The following data structures are used in Package CN_SRP_PRD_QUOTA_PUB:

- SRP_PRD_QUOTA_REC_TYPE
- SRP_PRD_QUOTA_TBL_TYPE

1.16.1.1 SRP_PRD_QUOTA_REC_TYPE

TYPE srp_prd_quota_rec_type IS RECORD

```
(
  PERIOD_NAME      CN_PERIOD_STATUSES.PERIOD_NAME%TYPE := FND_API.G_MISS_CHAR,
  PERIOD_TARGET    NUMBER := FND_API.G_MISS_NUM,
  PERIOD_PAYMENT   NUMBER := FND_API.G_MISS_NUM,
  PERFORMANCE_GOAL NUMBER := FND_API.G_MISS_NUM
);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–107 Parameters

Parameter	Data Type	Description
PERIOD_NAME	CN_PERIOD_STATUSES.PERIOD_NAME%TYPE	The period for which the target is applicable.
PERIOD_TARGET	NUMBER	The target to be assigned to this particular period.
PERIOD_PAYMENT	NUMBER	The fixed amount (if any) to be assigned to this particular period.
PERFORMANCE_GOAL	NUMBER	The performance goal (if any) to be set for this period.

1.16.1.2 PRD_QUOTA_TBL_TYPE

```
TYPE srp_prd_quota_tbl_type IS
  TABLE OF srp_prd_quota_rec_type INDEX BY BINARY_INTEGER ;
```

This data structure is derived from SRP_PRD_QUOTA_REC_TYPE.

1.16.2 Distribute_Srp_Prd_Quota

This API is used to distribute the target across the periods of a plan element. A mandatory parameter allows the user to choose between an even distribution of the target across the periods and a user configurable distribution as passed in by the user via the table type data structure.

Procedure Specification

```
PROCEDURE Distribute_Srp_Prd_Quota
(
  p_api_version    IN    NUMBER      := CN_API.G_MISS_NUM,
  p_init_msg_list  IN    VARCHAR2    := CN_API.G_FALSE,
  p_commit         IN    VARCHAR2    := CN_API.G_FALSE,
```

```

p_validation_level      IN      NUMBER      := CN_API.G_VALID_LEVEL_FULL,
p_salesrep_name         IN      CN_SALESREPS.NAME%TYPE,
p_employee_number       IN      CN_SALESREPS.EMPLOYEE_NUMBER%TYPE,
p_role_name             IN      CN_ROLES.NAME%TYPE,
p_cp_name               IN      CN_COMP_PLANS.NAME%TYPE,
p_srp_plan_start_date   IN      CN_SRP_PLAN_ASSIGNS.START_DATE%TYPE,
p_srp_plan_end_date     IN      CN_SRP_PLAN_ASSIGNS.END_DATE%TYPE,
p_pe_name               IN      CN_QUOTAS.NAME%TYPE,
p_target_amount         IN      CN_SRP_QUOTA_ASSIGNS.target%TYPE,
p_fixed_amount          IN      CN_SRP_QUOTA_ASSIGNS.payment_amount%TYPE,
p_performance_goal      IN      CN_SRP_QUOTA_ASSIGNS.performance_goal%TYPE,
p_even_distribute       IN      VARCHAR2,
p_srp_prd_quota_tbl     IN      srp_prd_quota_tbl_type,
x_return_status         OUT     NOCOPY VARCHAR2,
x_msg_count             OUT     NOCOPY NUMBER,
x_msg_data              OUT     NOCOPY VARCHAR2

);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–108 Distribute_Srp_Prd_Quota IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	Y	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_name	VARCHAR2	Y	The salesperson for whom the target is to be distributed.
p_employee_number	VARCHAR2	Y	The employee number of the salesperson.
p_role_name	VARCHAR2	Y	The role to which the salesperson is assigned during the periods for which the user is distributing the targets.

Table 1–108 Distribute_Srp_Prd_Quota IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_cp_name	VARCHAR2	Y	The compensation plan that is assigned to the salesperson and which contains the plan element for which the user would like to distribute target.
p_srp_plan_start_date	DATE	Y	The start date of the compensation plan assignment to the salesrep.
p_srp_plan_end_date	DATE	Y	The end date of the compensation plan assignment to the salesrep.
p_pe_name	VARCHAR2	Y	The plan element for which the target has to be distributed.
p_target_amount	NUMBER	Y	The target amount that has to be distributed across the periods.
p_fixed_amount	NUMBER	Y	The fixed amount that has to be distributed for the periods.
p_performance_goal	NUMBER	Y	The performance goal that has to be distributed for the periods.
p_even_distribute	VARCHAR2	Y	The configuration parameter used to choose between even distribution and user configurable distribution. Y = even, N = user configurable.
p_srp_prd_quota_tbl	prd_quota_tbl_type	Y	The user configured distribution if p_even_distribute is set to N.

The following table describes the OUT parameters associated with this API.

Table 1–109 Distribute_Srp_Prd_Quota OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2	Standard OUT parameters.

1.17 Package CN_PAYGROUP_PUB

The procedure(s) in this package can be used to get pay group information, validate the input, create pay groups, update pay groups, and delete pay groups. It is also

used to create entry into cn_pay_groups and to update salesrep pay group assignment.

- Get_Pay_Group_Sum
- Create_PayGroup
- Update_PayGroup
- Delete_PayGroup
- Assign_salesreps
- Update_salesrep_assignment

1.17.1 Data Structure Specifications

The following data structures are used in Package CN_PAYGROUP_PUB:

- PAYGROUP_REC_TYPE
- PAY_GROUP_TBL_TYPE

1.17.1.1 PAYGROUP_REC_TYPE

TYPE PayGroup_rec_type IS RECORD

name	cn_pay_groups.name%TYPE,	
period_set_name	cn_pay_groups.period_set_name%TYPE,	
period_type	cn_pay_groups.period_type%TYPE,	
start_date	cn_pay_groups.start_date%TYPE,	
end_date	cn_pay_groups.end_date%TYPE,	
pay_group_description	cn_pay_groups.pay_group_description%TYPE	:= NULL,
attribute_category	cn_pay_groups.attribute_category%TYPE	:= NULL,
attribute1	cn_pay_groups.attribute1%TYPE	:= NULL,
attribute2	cn_pay_groups.attribute2%TYPE	:= NULL,
attribute3	cn_pay_groups.attribute3%TYPE	:= NULL,
attribute4	cn_pay_groups.attribute4%TYPE	:= NULL,
attribute5	cn_pay_groups.attribute5%TYPE	:= NULL,
attribute6	cn_pay_groups.attribute6%TYPE	:= NULL,
attribute7	cn_pay_groups.attribute7%TYPE	:= NULL,
attribute8	cn_pay_groups.attribute8%TYPE	:= NULL,
attribute9	cn_pay_groups.attribute9%TYPE	:= NULL,
attribute10	cn_pay_groups.attribute10%TYPE	:= NULL,
attribute11	cn_pay_groups.attribute11%TYPE	:= NULL,
attribute12	cn_pay_groups.attribute12%TYPE	:= NULL,
attribute13	cn_pay_groups.attribute13%TYPE	:= NULL,
attribute14	cn_pay_groups.attribute14%TYPE	:= NULL,
attribute15	cn_pay_groups.attribute15%TYPE	:= NULL,

```

        object_version_number    NUMBER
    );

```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–110 Parameters

Parameter	Data Type	Description
name	VARCHAR2 (80)	Pay group name
period_set_name	VARCHAR2 (30)	Period set name.
period_type	VARCHAR2 (15)	Period type
start_date	Date	Start date
end_date	Date	End date
pay_group_description	VARCHAR2 (80)	Pay group description
attribute_category	VARCHAR2 (30)	Attribute category
Attribute 1	VARCHAR2 (150)	Standard attribute columns
Attribute 2	VARCHAR2 (150)	Standard attribute columns
Attribute 3	VARCHAR2 (150)	Standard attribute columns
Attribute 4	VARCHAR2 (150)	Standard attribute columns
Attribute 5	VARCHAR2 (150)	Standard attribute columns
Attribute 6	VARCHAR2 (150)	Standard attribute columns
Attribute 7	VARCHAR2 (150)	Standard attribute columns
Attribute 8	VARCHAR2 (150)	Standard attribute columns
Attribute 9	VARCHAR2 (150)	Standard attribute columns
Attribute 10	VARCHAR2 (150)	Standard attribute columns
Attribute 11	VARCHAR2 (150)	Standard attribute columns
Attribute 12	VARCHAR2 (150)	Standard attribute columns
Attribute 13	VARCHAR2 (150)	Standard attribute columns
Attribute 14	VARCHAR2 (150)	Standard attribute columns
Attribute 15	VARCHAR2 (150)	Standard attribute columns
object_version_number	NUMBER	Object version number

1.17.1.2 PAYGROUP_TBL_TYPE

```
TYPE PayGroup_tbl_type IS
    TABLE OF PayGroup_rec_type INDEX BY BINARY_INTEGER ;

TYPE PayGroup_assign_rec IS RECORD
(
    employee_type          cn_salesreps.type%TYPE,
    employee_number        cn_salesreps.employee_number%TYPE,
    assignment_start_date  cn_srp_pay_groups.start_date%TYPE,
    assignment_end_date    cn_srp_pay_groups.end_date%TYPE,
    attribute_category     cn_srp_pay_groups.attribute_category%TYPE := NULL,
    attribute1             cn_srp_pay_groups.attribute1%TYPE         := NULL,
    attribute2             cn_srp_pay_groups.attribute2%TYPE         := NULL,
    attribute3             cn_srp_pay_groups.attribute3%TYPE         := NULL,
    attribute4             cn_srp_pay_groups.attribute4%TYPE         := NULL,
    attribute5             cn_srp_pay_groups.attribute5%TYPE         := NULL,
    attribute6             cn_srp_pay_groups.attribute6%TYPE         := NULL,
    attribute7             cn_srp_pay_groups.attribute7%TYPE         := NULL,
    attribute8             cn_srp_pay_groups.attribute8%TYPE         := NULL,
    attribute9             cn_srp_pay_groups.attribute9%TYPE         := NULL,
    attribute10            cn_srp_pay_groups.attribute10%TYPE        := NULL,
    attribute11            cn_srp_pay_groups.attribute11%TYPE        := NULL,
    attribute12            cn_srp_pay_groups.attribute12%TYPE        := NULL,
    attribute13            cn_srp_pay_groups.attribute13%TYPE        := NULL,
    attribute14            cn_srp_pay_groups.attribute14%TYPE        := NULL,
    attribute15            cn_srp_pay_groups.attribute15%TYPE        := NULL );
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–111 Parameters

Parameter	Data Type	Description
employee_type	VARCHAR2 (30)	Employee type
employee_number	VARCHAR2 (30)	Employee number
assignment_start_date	Date	Assignment start date
Assignment_end_date	Date	Assignment end date
attribute_category	VARCHAR2 (30)	Attribute category
Attribute 1	VARCHAR2 (150)	Standard attribute columns
Attribute 2	VARCHAR2 (150)	Standard attribute columns
Attribute 3	VARCHAR2 (150)	Standard attribute columns

Table 1–111 Parameters

Parameter	Data Type	Description
Attribute 4	VARCHAR2 (150)	Standard attribute columns
Attribute 5	VARCHAR2 (150)	Standard attribute columns
Attribute 6	VARCHAR2 (150)	Standard attribute columns
Attribute 7	VARCHAR2 (150)	Standard attribute columns
Attribute 8	VARCHAR2 (150)	Standard attribute columns
Attribute 9	VARCHAR2 (150)	Standard attribute columns
Attribute 10	VARCHAR2 (150)	Standard attribute columns
Attribute 11	VARCHAR2 (150)	Standard attribute columns
Attribute 12	VARCHAR2 (150)	Standard attribute columns
Attribute 13	VARCHAR2 (150)	Standard attribute columns
Attribute 14	VARCHAR2 (150)	Standard attribute columns
Attribute 15	VARCHAR2 (150)	Standard attribute columns

G_MISS_PAYGROUP_REC PayGroup_rec_type;

G_MISS_PAYGROUP_REC_TB PayGroup_tbl_type;

1.17.2 Get_Pay_Group_Sum

This procedure gets the pay group information.

Procedure Specification

PROCEDURE Get_Pay_Group_Sum

(p_api_version	IN	NUMBER,
p_init_msg_list	IN	VARCHAR2 := FND_API.G_FALSE,
p_commit	IN	VARCHAR2 := FND_API.G_FALSE,
p_validation_level	IN	NUMBER := FND_API.G_VALID_LEVEL_FULL,
p_start_record	IN	NUMBER := -1,
p_fetch_size	IN	NUMBER := -1,
p_search_name	IN	VARCHAR2 := '%',
p_search_start_date	IN	DATE := FND_API.G_MISS_DATE,
p_search_end_date	IN	DATE := FND_API.G_MISS_DATE,
p_search_period_set_name	IN	VARCHAR2 := '%',
x_pay_group	OUT NOCOPY	PayGroup_tbl_type,

```
x_total_record          OUT NOCOPY    NUMBER,
x_return_status         OUT NOCOPY    VARCHAR2,
x_msg_count             OUT NOCOPY    NUMBER,
x_msg_data              OUT NOCOPY    VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–112 Get_Pay_Group_Sum IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_start_record	NUMBER	Y	
p_fetch_size	NUMBER	Y	
p_search_name	VARCHAR2	Y	
p_search_start_date	DATE	Y	
p_search_end_date	DATE	Y	
p_search_period_set_name	VARCHAR2	Y	

The following table describes the OUT parameters associated with this API.

Table 1–113 Get_Pay_Group_Sum OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.

Table 1–113 Get_Pay_Group_Sum OUT Parameters

Parameter	Data Type	Descriptions
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_pay_group	PayGroup_tbl_ type	
x_total_record	NUMBER	

1.17.3 CreatePayGroup

This procedure validates the input for a pay group and creates one if all validations are passed.

The following validations are performed by this API:

- Name should not be null
- Period set name should not be null
- Period Type should not be null
- Start date should not be null
- End date should not be null
- Start date should be less than end date
- Name, start date and end date should be unique
- Period set should be valid
- Period type should be valid

Procedure Specification

```

PROCEDURE Create_PayGroup
(
  p_api_version          IN   NUMBER,
  p_init_msg_list        IN   VARCHAR2,
  p_commit               IN   VARCHAR2,
  p_validation_level      IN   NUMBER,
  x_return_status        OUT  NOCOPY VARCHAR2,
  x_msg_count            OUT  NOCOPY NUMBER,
  x_msg_data             OUT  NOCOPY VARCHAR2,
  p_PayGroup_rec         IN   PayGroup_rec_type,
  x_loading_status       OUT  NOCOPY VARCHAR2,
  x_status               OUT  NOCOPY VARCHAR2
)

```

);

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–114 Distribute_Srp_Prd_Quota IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_PayGroup_rec	PayGroup_rec	Y	
p_fetch_size	NUMBER	Y	
p_search_name	VARCHAR2	Y	
p_search_start_date	DATE	Y	
p_search_end_date	DATE	Y	
p_search_period_set_name	VARCHAR2	Y	

The following table describes the OUT parameters associated with this API.

Table 1–115 Distribute_Srp_Prd_Quota OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure.

Table 1–115 Distribute_Srp_Prd_Quota OUT Parameters

Parameter	Data Type	Descriptions
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.17.4 Update_PayGroup

This API is used to update pay groups.

The following validations are performed by this API:

- Start date cannot be updated to null
- End date cannot be updated to null
- Start date should be less than end date
- The pay group to be updated should exist
- Pay group effectivity cannot be shrunk to be less than assignments
- Pay group name, start date and end date cannot be null
- If period_set is specified, then it should not be null
- If period type is specified, then it should not be null

Procedure Specification

```
PROCEDURE Update_PayGroup
(
  p_api_version      IN  NUMBER,
  p_init_msg_list    IN  VARCHAR2,
  p_commit           IN  VARCHAR2,
  p_validation_level IN  NUMBER,
  x_return_status    OUT NOCOPY VARCHAR2,
  x_msg_count        OUT NOCOPY NUMBER,
  x_msg_data         OUT NOCOPY VARCHAR2,
  p_old_PayGroup_rec IN  PayGroup_rec_type,
  p_PayGroup_rec     IN  PayGroup_rec_type,
  x_status           OUT NOCOPY VARCHAR2,
  x_loading_status   OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–116 *Update_PayGroup IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_old_PayGroup_rec	PayGroup_rec	Y	
p_PayGroup_rec	PayGroup_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–117 *Update_PayGroup OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure.
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.17.5 Delete_PayGroup

This API is used to delete pay groups.

The following validations are performed by this API:

- Pay group to be deleted must exist

Procedure Specification

```
PROCEDURE Delete_PayGroup
( p_api_version          IN  NUMBER,
```

```

p_init_msg_list      IN  VARCHAR2,
p_commit            IN  VARCHAR2,
p_validation_level   IN  NUMBER,
x_return_status      OUT NOCOPY VARCHAR2,
x_msg_count          OUT NOCOPY NUMBER,
x_msg_data           OUT NOCOPY VARCHAR2,
p_PayGroup_rec       IN  PayGroup_rec_type ,
x_status            OUT NOCOPY VARCHAR2,
x_loading_status     OUT NOCOPY VARCHAR2
) ;

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–118 Delete_PayGroup IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_PayGroup_rec	PayGroup_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–119 Delete_PayGroup OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure.

1.17.6 Assign_Salesreps

This API is used to create entry into cn_pay_groups. The procedure validates the input for a pay group and creates one if all validations are passed.

The following validations are performed by this API:

- Checks if the pay group exists
- Checks if the salesrep exists
- Checks that the assignments do not overlap

Procedure Specification

```
PROCEDURE Assign_salesreps
( p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2,
  p_commit               IN  VARCHAR2,
  p_validation_level     IN  NUMBER,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count            OUT NOCOPY NUMBER,
  x_msg_data             OUT NOCOPY VARCHAR2,
  p_pay_group_name       IN   cn_pay_groups.name%TYPE,
  p_pay_group_start_date IN   cn_pay_groups.start_date%TYPE,
  p_pay_group_end_date   IN   cn_pay_groups.end_date%TYPE,
  p_paygroup_assign_rec  IN   PayGroup_assign_rec,
  x_loading_status       OUT NOCOPY VARCHAR2,
  x_status               OUT NOCOPY VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–120 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.

Table 1–120 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_validation_level	NUMBER	N	Standard IN parameters.
p_pay_group_name	cn_pay_groups.name	Y	
p_pay_group_start_date	cn_pay_groups.start_date	Y	
p_pay_group_end_date	cn_pay_groups.end_date	Y	
p_PayGroup__assign_rec	PayGroup__assign_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–121 Assign_Salesreps OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure.
x_status	VARCHAR2 (50)	

1.17.7 Update_Salesrep_Assignment

This API updates the salesrep pay group assignment.

The following validations are performed by this API:

- Checks if the old and new pay group parameters are valid
- Checks if the assignment dates are valid
- Checks that the assignments do not overlap

Procedure Specification

PROCEDURE Update_salesrep_assignment

```
( p_api_version           IN  NUMBER,
  p_init_msg_list         IN  VARCHAR2,
  p_commit                IN  VARCHAR2,
  p_validation_level      IN  NUMBER,
  x_return_status         OUT  NOCOPY VARCHAR2,
  x_msg_count             OUT  NOCOPY NUMBER,
  x_msg_data              OUT  NOCOPY VARCHAR2,
  p_old_pay_group_name    IN   cn_pay_groups.name%TYPE,
  p_old_pay_group_start_date IN cn_pay_groups.start_date%TYPE,
  p_old_pay_group_end_date IN cn_pay_groups.end_date%TYPE,
  p_pay_group_name        IN   cn_pay_groups.name%TYPE,
  p_pay_group_start_date  IN   cn_pay_groups.start_date%TYPE,
  p_pay_group_end_date    IN   cn_pay_groups.end_date%TYPE,
  p_old_paygroup_assign_rec IN PayGroup_assign_rec,
  p_paygroup_assign_rec   IN   PayGroup_assign_rec,
  x_loading_status        OUT  NOCOPY   VARCHAR2,
  x_status                OUT  NOCOPY   VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–122 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_old_pay_group_name	cn_pay_groups.name	Y	
p_old_pay_group_start_date	cn_pay_groups.start_date	Y	
p_old_pay_group_end_date	cn_pay_groups.end_date	Y	

Table 1–122 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_pay_group_name	cn_pay_groups.name	Y	
p_pay_group_start_date	cn_pay_groups.start_date	Y	
p_old_pay_group_end_date	cn_pay_groups.end_date	Y	
p_old_paygroup_assign_rec	PayGroup_assign_rec	Y	
p_paygroup_assign_rec	PayGroup_assign_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–123 Assign_Salesreps OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure.
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.18 Package CN_PMT_PLAN_GET_PUB

This procedure is used to get a salesrep's payment plan for the given role or just to get the payment plans.

- Get_payment_plans

1.18.1 Data Structure Specifications

The following data structures are used in Package CN_PMT_PLAN_GET_PUB:

- PAYMENT_PLAN_REC

■ PAY_GROUP_TBL_TYPE

1.18.1.1 PAYGROUP_REC_TYPE

```
TYPE payment_plan_rec IS RECORD
(  srp_pmt_plan_id          cn_srp_pmt_plans.srp_pmt_plan_id%TYPE,
   payment_plan_id         cn_pmt_plans.pmt_plan_id%TYPE,
   salesrep_id             cn_salesreps.salesrep_id%TYPE,
   salesrep_name           cn_salesreps.name%TYPE,
   minimum_amount          cn_pmt_plans.minimum_amount%TYPE,
   maximum_amount          cn_pmt_plans.maximum_amount%TYPE,
   min_rec_flag            cn_pmt_plans.min_rec_flag%TYPE,
   max_rec_flag            cn_pmt_plans.max_rec_flag%TYPE,
   start_date              cn_pmt_plans.start_Date%TYPE,
   end_date                cn_pmt_plans.end_Date%TYPE,
   payment_plan_name       cn_pmt_plans.name%TYPE,
   period_end_date         cn_pmt_plans.end_date%TYPE,
   payment_end_date        cn_pmt_plans.end_date%TYPE,
   paid_flag               cn_payruns.status%TYPE,
   end_date_modify_flag    varchar2(01),
   recoverable_interval_Type cn_interval_types_all_tl.name%TYPE,
   pay_against_commission  cn_lookups.meaning%TYPE,
   object_version_number   cn_pmt_plans.object_version_number%TYPE,
   payment_group_code      cn_pmt_plans.payment_group_code%TYPE
);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–124 Parameters

Parameter	Data Type	Description
srp_pmt_plan_id	NUMBER (15,0)	Salesrep payment plan ID
payment_plan_id	NUMBER (15,0)	Payment plan ID
salesrep_id	NUMBER (15,0)	Salesrep ID
salesrep_name	VARCHAR2 (360)	Salesrep name
minimum_amount	NUMBER	Minimum amount
maximum_amount	NUMBER	Maximum amount
min_rec_flag	VARCHAR2 (1)	Minimum amount recoverable flag
max_rec_flag	VARCHAR2 (1)	Maximum amount recoverable flag

Table 1–124 Parameters

Parameter	Data Type	Description
start_date	DATE	Start date
end_date	DATE	End date
payment_plan_name	VARCHAR2 (30)	Payment plan name
period_end_date	DATE	Period end date
payment_end_date	DATE	Payment end date
paid_flag	VARCHAR2 (80)	Payrun status
end_date_modify_flag	VARCHAR2 (1)	End date modified flag
recoverable_interval_Type	VARCHAR2 (80)	Recoverable interval type
pay_against_commission	VARCHAR2 (80)	Pay against commission
object_version_number	NUMBER	Object version number
payment_group_code	VARCHAR2 (30)	Payment group code

1.18.1.2 PAYGROUP_REC_TYPE

TYPE payment_plan_tbl IS TABLE OF payment_plan_rec INDEX BY BINARY_INTEGER;

1.18.2 Get_payment_plans

This procedure gets a salesrep's payment plan.

Procedure Specification

```

PROCEDURE Get_payment_plans
( p_api_version IN NUMBER,
  p_init_msg_list IN VARCHAR2,
  p_commit IN VARCHAR2,
  p_validation_level IN NUMBER,
  x_return_status OUT NOCOPY VARCHAR2,
  x_msg_count OUT NOCOPY NUMBER,
  x_msg_data OUT NOCOPY VARCHAR2,
  p_salesrep_id IN cn_salesreps.salesrep_id%TYPE,
  p_start_record IN NUMBER,
  p_increment_count IN NUMBER,
  p_order_by IN VARCHAR2,
  p_payrun_id IN NUMBER,
```

```
x_payment_plan_tbl      OUT NOCOPY cn_pmt_plan_get_pub.payment_plan_tbl,  
x_total_records         OUT NOCOPY NUMBER,  
x_status                OUT NOCOPY VARCHAR2,  
x_loading_status        OUT NOCOPY VARCHAR2  
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–125 Get_payment_plans IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_id	cn_salesreps.id	Y	
p_role_id	cn_roles_id	Y	
payment_plan_id	cn_pmt_plans.id	Y	

The following table describes the OUT parameters associated with this API.

Table 1–126 Get_payment_plans OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status		Detailed Error Message

1.19 Package CN_PMTPLAN_PUB

This procedure is used to create, update, and delete payment plans.

- Create_PmtPlan
- Update_PmtPlan
- Delete_PmtPlan

1.19.1 Data Structure Specifications

The following data structures are used in Package CN_PMTPLAN_PUB:

- PMTPLAN_REC_TYPE

1.19.1.1 PMTPLAN_REC_TYPE

TYPE PmtPlan_rec_type IS RECORD

name	cn_pmt_plans.name%TYPE,	
minimum_amount	cn_pmt_plans.minimum_amount%TYPE,	
maximum_amount	cn_pmt_plans.maximum_amount%TYPE,	
min_rec_flag	cn_pmt_plans.min_rec_flag%TYPE,	
max_rec_flag	cn_pmt_plans.max_rec_flag%TYPE,	
max_recovery_amount	cn_pmt_plans.max_recovery_amount%TYPE,	
credit_type_name	cn_credit_types.name%TYPE,	
pay_interval_type_name	cn_interval_types.name%TYPE,	
start_date	cn_pmt_plans.start_date%TYPE,	
end_date	cn_pmt_plans.end_date%TYPE,	
attribute_category	cn_pmt_plans.attribute_category%TYPE	:= NULL,
attribute1	cn_pmt_plans.attribute1%TYPE	:= NULL,
attribute2	cn_pmt_plans.attribute2%TYPE	:= NULL,
attribute3	cn_pmt_plans.attribute3%TYPE	:= NULL,
attribute4	cn_pmt_plans.attribute4%TYPE	:= NULL,
attribute5	cn_pmt_plans.attribute5%TYPE	:= NULL,
attribute6	cn_pmt_plans.attribute6%TYPE	:= NULL,
attribute7	cn_pmt_plans.attribute7%TYPE	:= NULL,
attribute8	cn_pmt_plans.attribute8%TYPE	:= NULL,
attribute9	cn_pmt_plans.attribute9%TYPE	:= NULL,
attribute10	cn_pmt_plans.attribute10%TYPE	:= NULL,
attribute11	cn_pmt_plans.attribute11%TYPE	:= NULL,
attribute12	cn_pmt_plans.attribute12%TYPE	:= NULL,
attribute13	cn_pmt_plans.attribute13%TYPE	:= NULL,
attribute14	cn_pmt_plans.attribute14%TYPE	:= NULL,
attribute15	cn_pmt_plans.attribute15%TYPE	:= NULL,
payment_group_code	cn_pmt_plans.payment_group_code%TYPE	:= NULL

);

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–127 Parameters

Parameter	Data Type	Description
minimum_amount	NUMBER	Minimum amount
maximum_amount	NUMBER	Maximum amount
min_rec_flag	VARCHAR2 (1)	Minimum amount recoverable flag
max_rec_flag	VARCHAR2 (1)	Maximum amount recoverable flag
max_recovery_amount	NUMBER	Maximum recovery amount
credit_type_name	VARCHAR2 (80)	Credit type name
pay_interval_type_name	VARCHAR2 (80)	Pay interval type name
start_date	DATE	Start date
end_date	DATE	End date
attribute_category	VARCHAR2 (30)	Attribute category
Attribute1	VARCHAR2 (150)	Standard attribute columns
Attribute2	VARCHAR2 (150)	Standard attribute columns
Attribute3	VARCHAR2 (150)	Standard attribute columns
Attribute4	VARCHAR2 (150)	Standard attribute columns
Attribute5	VARCHAR2 (150)	Standard attribute columns
Attribute6	VARCHAR2 (150)	Standard attribute columns
Attribute7	VARCHAR2 (150)	Standard attribute columns
Attribute8	VARCHAR2 (150)	Standard attribute columns
Attribute9	VARCHAR2 (150)	Standard attribute columns
Attribute10	VARCHAR2 (150)	Standard attribute columns
Attribute11	VARCHAR2 (150)	Standard attribute columns
Attribute12	VARCHAR2 (150)	Standard attribute columns
Attribute13	VARCHAR2 (150)	Standard attribute columns

Table 1–127 Parameters

Parameter	Data Type	Description
Attribute14	VARCHAR2 (150)	Standard attribute columns
Attribute15	VARCHAR2 (150)	Standard attribute columns
payment_group_code	VARCHAR2 (30)	Payment group code

Note: The parameter *g_mode* (VARCHAR2 (30)) is a global parameter used to indicate if an operation is insert/update.

1.19.2 Create_PmtPlan

This procedure is used to create a new payment plan. It validates the input for a payment plan and creates one if all validations are passed.

The following validations are performed by this API:

- Start date is less than end date
- Name is not missing or null
- Credit type is not missing or null
- Start date is not null
- Min_rec_flag is 'Y', 'N' or null
- Max_rec_flag is 'Y', 'N' or null
- Payment plan is unique
- Credit Type is valid
- Pay interval is valid 'PERIOD', 'QUARTER', 'YEAR' or null

Procedure Specification

```
PROCEDURE Create_PmtPlan
(
  p_api_version      IN   NUMBER,
  p_init_msg_list    IN   VARCHAR2,
  p_commit           IN   VARCHAR2,
  p_validation_level  IN   NUMBER,
  x_return_status     OUT  NOCOPY VARCHAR2,
  x_msg_count        OUT  NOCOPY NUMBER,
  x_msg_data         OUT  NOCOPY VARCHAR2,
  p_PmtPlan_rec      IN   PmtPlan_rec_type,
```

```

        x_loading_status    OUT NOCOPY VARCHAR2,
        x_status            OUT NOCOPY VARCHAR2
    );
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–128 Create_PmtPlan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_pmt_plan_rec	PmtPlan_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–129 Create_PmtPlan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.19.3 Update_PmtPlan

This procedure is used to update a payment plan. The payment plan must already exist.

The following validations are performed by this API:

- Start date cannot be updated to null
- If an end date is specified, it cannot be less than the start date
- The payment plan to be updated already exists
- If the payment plan has been assigned, then recoverable flags can't be updated
- Start and end dates can't be updated to affect existing assignments
- Name and start date are mandatory parameters
- If credit type is provided, it should be valid
- If pay interval is provided, it should be valid

Procedure Specification

```
PROCEDURE Update_PmtPlan
( p_api_version IN NUMBER,
  p_init_msg_list IN VARCHAR2,
  p_commit IN VARCHAR2,
  p_validation_level IN NUMBER,
  x_return_status OUT NOCOPY VARCHAR2,
  x_msg_count OUT NOCOPY NUMBER,
  x_msg_data OUT NOCOPY VARCHAR2,
  p_old_PmtPlan_rec IN PmtPlan_rec_type,
  p_PmtPlan_rec IN PmtPlan_rec_type,
  x_status OUT NOCOPY VARCHAR2,
  x_loading_status OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–130 Update_PmtPlan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.

Table 1–130 Update_PmtPlan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_old_pmt_plan_rec	PmtPlan_rec_type	Y	
p_pmt_plan_rec	PmtPlan_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–131 Update_PmtPlan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.19.4 Delete_PmtPlan

This procedure is used to delete a payment plan. The payment plan must already exist.

The following validations are performed by this API:

- The payment plan to be deleted already exists
- The payment plan cannot be deleted if it is assigned to salesreps

Procedure Specification

```
PROCEDURE Delete_PmtPlan
( p_api_versionIN NUMBER,
  p_init_msg_list      IN  VARCHAR2,
  p_commit              IN  VARCHAR2,
  p_validation_level    IN  NUMBER,
```

```

x_return_status      OUT NOCOPY VARCHAR2,
x_msg_count          OUT NOCOPY NUMBER,
x_msg_data           OUT NOCOPY VARCHAR2,
p_PmtPlan_rec        IN  PmtPlan_rec_type,
x_status             OUT NOCOPY VARCHAR2,
x_loading_status      OUT NOCOPY VARCHAR2
) ;

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–132 Delete_PmtPlan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
x_PmtPlan_rec	PmtPlan_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–133 Delete_PmtPlan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.20 Package CN_PMTSUB_PUB

This procedure is used to pay a payrun and update the subledger, run the concurrent program, and pay the payrun using a concurrent program.

- Pay
- Pay_Payrun_conc
- submit_request

1.20.1 Data Structure Specifications

There are no data structures used.

1.20.2 Pay

This procedure is used to pay a payrun and update the subledger.

Procedure Specification

```
PROCEDURE Pay
( p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := cn_api.g_false,
  p_commit               IN  VARCHAR2 := cn_api.g_false,
  p_validation_level     IN  NUMBER   := cn_api.g_valid_level_full,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count           OUT NOCOPY NUMBER,
  x_msg_data            OUT NOCOPY VARCHAR2,
  p_payrun_name         IN   cn_payruns.name%TYPE,
  x_status              OUT NOCOPY VARCHAR2,
  x_loading_status      OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–134 Pay IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.

Table 1–134 Pay IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_Payrun_name	cn_payruns.name	Y	

The following table describes the OUT parameters associated with this API.

Table 1–135 Pay OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.20.3 Pay_Payrun_conc

This procedure is used as the executable for the concurrent program CN_PAY_PAYRUN to pay a payrun.

Procedure Specification

```
PROCEDURE Pay_Payrun_conc
( errbuf OUT NOCOPY VARCHAR2,
  retcode OUT NOCOPY NUMBER ,
  p_name          cn_payruns.name%TYPE );
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–136 Pay IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_Payrun_name	cn_payruns.name	Y	

The following table describes the OUT parameters associated with this API.

Table 1–137 Pay OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.20.4 submit_request

This procedure is used to pay a payrun using a concurrent program.

Procedure Specification

```
PROCEDURE submit_request (p_payrun_id    IN    NUMBER,
                          x_request_id    OUT NOCOPY NUMBER) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–138 Pay IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_Payrun_name	cn_payruns.name	Y	

The following table describes the OUT parameters associated with this API.

Table 1–139 Pay OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.21 Package CN_SRP_PMT_PLANS_PUB

This procedure is used to create, update, and delete payment plan assignments for salesreps individually or in mass.

- Create_Srp_Pmt_Plan
- Create_Mass_Asgn_Srp_Pmt_Plan
- Update_Srp_Pmt_Plan
- Update_Mass_Asgn_Srp_Pmt_Plan
- Delete_Srp_Pmt_Plan
- Delete_Mass_Asgn_Srp_Pmt_Plan

1.21.1 Data Structure Specifications

The following data structures are used in CN_SRP_PMT_PLANS_PUB:

- SRP_PMT_PLANS_REC_TYPE
- SRP_PMT_PLANS_TBL_TYPE

1.21.1.1 SRP_PMT_PLANS_REC_TYPE

```
TYPE srp_pmt_plans_rec_type IS RECORD
(PMT_PLAN_NAME      cn_pmt_plans.name%TYPE := CN_API.G_MISS_CHAR,
SALESREP_TYPE       VARCHAR2(100)        := CN_API.G_MISS_CHAR,
EMP_NUM             VARCHAR2(30)         := CN_API.G_MISS_CHAR,
START_DATE          cn_srp_pmt_plans.start_date%TYPE
                    := CN_API.G_MISS_DATE,
END_DATE            cn_srp_pmt_plans.end_date%TYPE
                    := CN_API.G_MISS_DATE,
MINIMUM_AMOUNT      cn_srp_pmt_plans.minimum_amount%TYPE
                    := CN_API.G_MISS_NUM,
MAXIMUM_AMOUNT      cn_srp_pmt_plans.maximum_amount%TYPE
                    := CN_API.G_MISS_NUM,
MAX_RECOVERY_AMOUNT cn_srp_pmt_plans.max_recovery_amount%TYPE
                    := CN_API.G_MISS_NUM,
ATTRIBUTE_CATEGORY   cn_srp_pmt_plans.attribute_category%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE1          cn_srp_pmt_plans.attribute1%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE2          cn_srp_pmt_plans.attribute2%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE3          cn_srp_pmt_plans.attribute3%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE4          cn_srp_pmt_plans.attribute4%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE5          cn_srp_pmt_plans.attribute5%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE6          cn_srp_pmt_plans.attribute6%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE7          cn_srp_pmt_plans.attribute7%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE8          cn_srp_pmt_plans.attribute8%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE9          cn_srp_pmt_plans.attribute9%TYPE
                    := CN_API.G_MISS_CHAR,
ATTRIBUTE10         cn_srp_pmt_plans.attribute10%TYPE
                    := CN_API.G_MISS_CHAR,
```

```

ATTRIBUTE11          cn_srp_pmt_plans.attribute11%TYPE
                     := CN_API.G_MISS_CHAR,
ATTRIBUTE12          cn_srp_pmt_plans.attribute12%TYPE
                     := CN_API.G_MISS_CHAR,
ATTRIBUTE13          cn_srp_pmt_plans.attribute13%TYPE
                     := CN_API.G_MISS_CHAR,
ATTRIBUTE14          cn_srp_pmt_plans.attribute14%TYPE
                     := CN_API.G_MISS_CHAR,
ATTRIBUTE15          cn_srp_pmt_plans.attribute15%TYPE
                     := CN_API.G_MISS_CHAR,
OBJECT_VERSION_NUMBER cn_srp_pmt_plans.object_version_number%TYPE,
SRP_ROLE_ID          cn_srp_pmt_plans.srp_role_id%TYPE
                     := NULL,
ROLE_PMT_PLAN_ID     cn_srp_pmt_plans.role_pmt_plan_id%TYPE
                     := NULL,
LOCK_FLAG            cn_srp_pmt_plans.lock_flag%TYPE
                     := NULL
);

```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–140 Parameters

Parameter	Data Type	Description
Pmt_plan_name	VARCHAR2 (30)	Payment plan name
Salesrep_type	VARCHAR2 (100)	Salesrep type
Emp_num	VARCHAR2 (30)	Employee number
start_date	DATE	Start date
end_date	DATE	End date
minimum_amount	NUMBER	Minimum number
maximum_amount	NUMBER	Maximum number
max_recovery_amount	NUMBER	Maximum recovery amount
attribute_category	VARCHAR2 (30)	Attribute category
Attribute1	VARCHAR2 (150)	Standard attribute columns
Attribute2	VARCHAR2 (150)	Standard attribute columns
Attribute3	VARCHAR2 (150)	Standard attribute columns

Table 1–140 Parameters

Parameter	Data Type	Description
Attribute4	VARCHAR2 (150)	Standard attribute columns
Attribute5	VARCHAR2 (150)	Standard attribute columns
Attribute6	VARCHAR2 (150)	Standard attribute columns
Attribute7	VARCHAR2 (150)	Standard attribute columns
Attribute8	VARCHAR2 (150)	Standard attribute columns
Attribute9	VARCHAR2 (150)	Standard attribute columns
Attribute10	VARCHAR2 (150)	Standard attribute columns
Attribute11	VARCHAR2 (150)	Standard attribute columns
Attribute12	VARCHAR2 (150)	Standard attribute columns
Attribute13	VARCHAR2 (150)	Standard attribute columns
Attribute14	VARCHAR2 (150)	Standard attribute columns
Attribute15	VARCHAR2 (150)	Standard attribute columns
object_version_number	NUMBER	Object version number
Srp_role_id	NUMBER (15, 0)	Salesrep role id
Role_pmt_plan_id	NUMBER (15, 0)	Role payment plan id
Lock_flag	VARCHAR2 (1)	Lock flag

Note: This is a global variable that represents missing values:
G_MISS_SRP_PMT_PLANS_REC srp_pmt_plans_rec_type;

1.21.1.2 SRP_PMT_PLANS_TBL_TYPE

This is the Table Record Type for srp_pmt_plans_rec_type:

```
TYPE srp_pmt_plans_tbl_type IS TABLE OF srp_pmt_plans_rec_type
INDEX BY BINARY_INTEGER;
```

Note: This is a global variable that represents missing values:
G_MISS_SPP_TBL_LIST srp_pmt_plans_tbl_type;

1.21.2 Create_Srp_Pmt_Plan

This procedure is used to create a new payment plan assignment to a salesrep.

Default actions for this procedure:

If a user does not pass in information for the following fields, this program inherits the data from the payment plan:

- MINIMUM_AMOUNT
- MAXIMUM_AMOUNT
- MAX_RECOVERY_AMOUNT

If a user does not pass in information for START_DATE and END_DATE, this program gets the overlapped date range between the Payment Plan active range and Salesrep's active range and uses that date range.

Procedure Specification

```
PROCEDURE Create_Srp_Pmt_Plan
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := CN_API.G_FALSE,
  p_commit               IN  VARCHAR2 := CN_API.G_FALSE,
  p_validation_level     IN  NUMBER   := CN_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_srp_pmt_plans_rec    IN  srp_pmt_plans_rec_type
                        := G_MISS_SRP_PMT_PLANS_REC,
  x_srp_pmt_plan_id      OUT NOCOPY  NUMBER,
  x_loading_status       OUT NOCOPY  VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–141 Create_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.

Table 1–141 Create_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–142 Create_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.21.3 Create_Mass_Asgn_Srp_Pmt_Plan

This procedure is used to create a new mass payment plan assignment to salesreps.

Procedure Specification

```

PROCEDURE Create_Mass_Asgn_Srp_Pmt_Plan
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit               IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level      IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status         OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_srp_role_id          IN  NUMBER,
  p_role_pmt_plan_id     IN  NUMBER,
  x_srp_pmt_plan_id      OUT NOCOPY  NUMBER,
  x_loading_status       OUT NOCOPY  VARCHAR2
);

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–143 Create_Mass_Asgn_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_role_pmt_plan_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–144 Create_Mass_Asgn_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.21.4 Update_Srp_Pmt_Plan

This procedure is used to update the payment plan assignment of a salesrep.

You cannot update a salesrep's plan assignment in the following situations:

- Change the START_DATE assignment: If the payment plan already been used by a payment worksheet and the payment has been paid during this assignment's date range, no update or deleting is allowed for the Srp Payment Plan assignment START_DATE. Otherwise, the user can expand or shrink the START_DATE.

- Shorten the END_DATE assignment: If the payment plan has already been used by a payment worksheet and the payment has been paid during this assignment's date range, the END_DATE of this Srp Payment Plan assignment cannot be shortened, but the user can expand it.

Procedure Specification

```
PROCEDURE Update_Srp_Pmt_Plan
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := CN_API.G_FALSE,
  p_commit               IN  VARCHAR2 := CN_API.G_FALSE,
  p_validation_level      IN  NUMBER   := CN_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_old_srp_pmt_plans_rec IN  srp_pmt_plans_rec_type
                        := G_MISS_SRP_PMT_PLANS_REC,
  p_srp_pmt_plans_rec    IN  srp_pmt_plans_rec_type
                        := G_MISS_SRP_PMT_PLANS_REC,
  x_loading_status       OUT NOCOPY  VARCHAR2,
  p_check_lock           IN  VARCHAR2
:= NULL
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–145 Update_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–146 Update_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.21.5 Update_Mass_Asgn_Srp_Pmt_Plan

This procedure is used to update mass payment plan assignment of salesreps.

Procedure Specification

```
PROCEDURE Update_Mass_Asgn_Srp_Pmt_plan
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit               IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level     IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_srp_role_id          IN  NUMBER,
  p_role_pmt_plan_id     IN  NUMBER,
  x_loading_status       OUT NOCOPY  VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–147 Update_Mass_Asgn_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.

Table 1–147 Update_Mass_Asgn_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_role_id	NUMBER	Y	
p_role_pmt_plan_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–148 Update_Mass_Asgn_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.21.6 Delete_Srp_Pmt_Plan

This procedure is used to delete a payment plan assignment to a salesrep.

Note: If a payment plan has already been used by a payment worksheet and the payment has been paid during this assignment's date range, update or delete is allowed for this Srp Payment Plan assignment.

Procedure Specification

```
PROCEDURE Delete_Srp_Pmt_Plan
(
  p_api_version      IN  NUMBER,
  p_init_msg_list    IN  VARCHAR2 := CN_API.G_FALSE,
  p_commit           IN  VARCHAR2 := CN_API.G_FALSE,
  p_validation_level  IN  NUMBER   := CN_API.G_VALID_LEVEL_FULL,
  x_return_status     OUT NOCOPY  VARCHAR2,
  x_msg_count        OUT NOCOPY  NUMBER,
  x_msg_data         OUT NOCOPY  VARCHAR2,
```

```

p_srp_pmt_plans_rec  IN  srp_pmt_plans_rec_type
                        := G_MISS_SRP_PMT_PLANS_REC,
x_loading_status      OUT NOCOPY  VARCHAR2,
p_check_lock          IN   VARCHAR2
:= NULL

```

)Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–149 Delete_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–150 Delete_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.21.7 Delete_Mass_Asgn_Srp_Pmt_Plan

This procedure is used to delete a mass payment plan assignment to salesreps.

Procedure Specification

```
PROCEDURE Delete_Mass_Asgn_Srp_Pmt_Plan
(
  p_api_version      IN  NUMBER,
  p_init_msg_list    IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit           IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level  IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status     OUT NOCOPY  VARCHAR2,
  x_msg_count        OUT NOCOPY  NUMBER,
  x_msg_data         OUT NOCOPY  VARCHAR2,
  p_srp_role_id      IN  NUMBER,
  p_role_pmt_plan_id IN  NUMBER,
  x_loading_status    OUT NOCOPY  VARCHAR2
);
```

)Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–151 Delete_Mass_Asgn_Srp_Pmt_Plan IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_role_id	NUMBER	Y	
p_role_pmt_plan_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–152 Delete_Mass_Asgn_Srp_Pmt_Plan OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.22 Package CN_WKSHT_CT_UP_PUB

This procedure is used to refresh the worksheet and to update, create, and delete payment plans at the salesrep level.

- Create_delete_Wrkhst
- Apply_payment_plan_upd
- Apply_payment_plan_cre
- Apply_payment_plan_del

1.22.1 Data Structure Specifications

The following data structures are used in CN_SRP_PMT_PLANS_PUB:

- SRP_PMT_PLANS_REC_TYPE

1.22.1.1 SRP_PMT_PLANS_REC_TYPE

```

TYPE srp_pmt_plans_rec_type IS RECORD
(PMT_PLAN_NAME      cn_pmt_plans.name%TYPE := CN_API.G_MISS_CHAR,
SALESREP_TYPE       VARCHAR2(100)      := CN_API.G_MISS_CHAR,
EMP_NUM             VARCHAR2(30)       := CN_API.G_MISS_CHAR,
START_DATE          cn_srp_pmt_plans.start_date%TYPE
                    := CN_API.G_MISS_DATE,
END_DATE            cn_srp_pmt_plans.end_date%TYPE
                    := CN_API.G_MISS_DATE,
MINIMUM_AMOUNT      cn_srp_pmt_plans.minimum_amount%TYPE
                    := CN_API.G_MISS_NUM,
MAXIMUM_AMOUNT      cn_srp_pmt_plans.maximum_amount%TYPE
                    := CN_API.G_MISS_NUM,
MAX_RECOVERY_AMOUNT cn_srp_pmt_plans.max_recovery_amount%TYPE
                    := CN_API.G_MISS_NUM,
```

```

ATTRIBUTE_CATEGORY      cn_srp_pmt_plans.attribute_category%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE1              cn_srp_pmt_plans.attribute1%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE2              cn_srp_pmt_plans.attribute2%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE3              cn_srp_pmt_plans.attribute3%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE4              cn_srp_pmt_plans.attribute4%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE5              cn_srp_pmt_plans.attribute5%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE6              cn_srp_pmt_plans.attribute6%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE7              cn_srp_pmt_plans.attribute7%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE8              cn_srp_pmt_plans.attribute8%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE9              cn_srp_pmt_plans.attribute9%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE10             cn_srp_pmt_plans.attribute10%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE11             cn_srp_pmt_plans.attribute11%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE12             cn_srp_pmt_plans.attribute12%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE13             cn_srp_pmt_plans.attribute13%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE14             cn_srp_pmt_plans.attribute14%TYPE
                        := CN_API.G_MISS_CHAR,
ATTRIBUTE15             cn_srp_pmt_plans.attribute15%TYPE
                        := CN_API.G_MISS_CHAR,
);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–153 Parameters

Parameter	Data Type	Description
Pmt_plan_name	VARCHAR2 (30)	Payment plan name
Salesrep_type	VARCHAR2 (100)	Salesrep type
Emp_num	VARCHAR2 (30)	Employee number

Table 1–153 Parameters

Parameter	Data Type	Description
start_date	DATE	Start date
end_date	DATE	End date
Minimum_amount	NUMBER	Minimum number
Maximum_amount	NUMBER	Maximum number
Max_recovery_amount	NUMBER	Maximum recovery amount
Attribute_category	VARCHAR2 (30)	Attribute category
Attribute1	VARCHAR2 (150)	Standard attribute columns
Attribute2	VARCHAR2 (150)	Standard attribute columns
Attribute3	VARCHAR2 (150)	Standard attribute columns
Attribute4	VARCHAR2 (150)	Standard attribute columns
Attribute5	VARCHAR2 (150)	Standard attribute columns
Attribute6	VARCHAR2 (150)	Standard attribute columns
Attribute7	VARCHAR2 (150)	Standard attribute columns
Attribute8	VARCHAR2 (150)	Standard attribute columns
Attribute9	VARCHAR2 (150)	Standard attribute columns
Attribute10	VARCHAR2 (150)	Standard attribute columns
Attribute11	VARCHAR2 (150)	Standard attribute columns
Attribute12	VARCHAR2 (150)	Standard attribute columns
Attribute13	VARCHAR2 (150)	Standard attribute columns
Attribute14	VARCHAR2 (150)	Standard attribute columns
Attribute15	VARCHAR2 (150)	Standard attribute columns
object_version_number	NUMBER	Object version number
Srp_role_id	NUMBER (15, 0)	Salesrep role id
Role_pmt_plan_id	NUMBER (15, 0)	Role payment plan id
Lock_flag	VARCHAR2 (1)	Lock flag

Note: This is a global variable that represents missing values:

G_MISS_SRP_PMT_PLANS_REC srp_pmt_plans_rec_type;

1.22.2 Create_delete_Wrkhst

This procedure is used to refresh the worksheet with the srp_pmt_asgn_id.

Procedure Specification

```
Procedure Create_delete_Wrkhst
( p_api_version      IN  NUMBER,
  p_init_msg_list    IN  VARCHAR2,
  p_commit           IN  VARCHAR2,
  p_validation_level  IN  NUMBER,
  x_return_status     OUT NOCOPY VARCHAR2,
  x_msg_count        OUT NOCOPY NUMBER,
  x_msg_data         OUT NOCOPY VARCHAR2,
  p_salesrep_id       IN  NUMBER,
  p_srp_pmt_asgn_id   IN  NUMBER,
  p_payrun_id         IN  NUMBER,
  x_status           OUT NOCOPY VARCHAR2,
  x_loading_status    OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–154 Create_delete_Wrkhst IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_id	NUMBER	Y	
p_srp_pmt_asgn_id	NUMBER	Y	

Table 1–154 Create_delete_Wrkhst IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_payrun_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–155 Create_delete_Wrkhst OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_status	VARCHAR2	
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.22.3 Apply_payment_plan_upd

This procedure is used to update a payment plan at the salesrep level.

Procedure Specification

Procedure Apply_payment_plan_upd

```

( p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2,
  p_commit               IN  VARCHAR2,
  p_validation_level     IN  NUMBER,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count            OUT NOCOPY NUMBER,
  x_msg_data             OUT NOCOPY VARCHAR2,
  p_salesrep_id          IN  NUMBER,
  p_srp_pmt_asgn_id      IN  NUMBER,
  p_payrun_id            IN  NUMBER,
  p_old_srp_pmt_plans_rec IN  srp_pmt_plans_rec_type := G_MISS_SRP_PMT_PLANS_
REC,
  p_srp_pmt_plans_rec    IN  srp_pmt_plans_rec_type :=G_MISS_SRP_PMT_PLANS_
REC,
  x_status               OUT NOCOPY VARCHAR2,
  x_loading_status       OUT NOCOPY VARCHAR2
) ;

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–156 Apply_payment_plan_upd IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_id	NUMBER	Y	
p_srp_pmt_asgn_id	NUMBER	Y	
p_payrun_id	NUMBER	Y	
p_old_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–157 Apply_payment_plan_upd OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_status	VARCHAR2	
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.22.4 Apply_payment_plan_cre

This procedure is used to create a payment plan at the salesrep level.

Procedure Specification

Procedure Apply_payment_plan_cre

```
( p_api_version      IN  NUMBER,
  p_init_msg_list     IN  VARCHAR2,
  p_commit            IN  VARCHAR2,
  p_validation_level   IN  NUMBER,
  x_return_status      OUT NOCOPY VARCHAR2,
  x_msg_count          OUT NOCOPY NUMBER,
  x_msg_data           OUT NOCOPY VARCHAR2,
  p_salesrep_id        IN  NUMBER,
  p_srp_pmt_asgn_id    IN  NUMBER,
  p_payrun_id          IN  NUMBER,
  p_srp_pmt_plans_rec  IN  srp_pmt_plans_rec_type := G_MISS_SRP_PMT_PLANS_REC,
  x_status             OUT NOCOPY VARCHAR2,
  x_loading_status     OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–158 Apply_payment_plan_cre IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_id	NUMBER	Y	
p_srp_pmt_asgn_id	NUMBER	Y	
p_payrun_id	NUMBER	Y	

Table 1–158 Apply_payment_plan_cre IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–159 Apply_payment_plan_cre OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_status	VARCHAR2	
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.22.5 Apply_payment_plan_del

This procedure is used to delete a payment plan at the salesrep level.

Procedure Specification

Procedure Apply_payment_plan_del

```
( p_api_version      IN  NUMBER,
  p_init_msg_list     IN  VARCHAR2,
  p_commit            IN  VARCHAR2,
  p_validation_level  IN  NUMBER,
  x_return_status     OUT NOCOPY VARCHAR2,
  x_msg_count         OUT NOCOPY NUMBER,
  x_msg_data          OUT NOCOPY VARCHAR2,
  p_salesrep_id       IN  NUMBER,
  p_srp_pmt_asgn_id   IN  NUMBER,
  p_payrun_id         IN  NUMBER,
  p_srp_pmt_plans_rec IN  srp_pmt_plans_rec_type := G_MISS_SRP_PMT_PLANS_REC,
  x_status            OUT NOCOPY VARCHAR2,
  x_loading_status    OUT NOCOPY VARCHAR2
) ;
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–160 Apply_payment_plan_del IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_salesrep_id	NUMBER	Y	
p_srp_pmt_asgn_id	NUMBER	Y	
p_payrun_id	NUMBER	Y	
p_srp_pmt_plans_rec	srp_pmt_plans_rec_type	Y	

The following table describes the OUT parameters associated with this API.

Table 1–161 Apply_payment_plan_del OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_status	VARCHAR2	
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.23 Package CN_WKSHT_GET_PUB

This procedure is used to get a salesrep's worksheet information for a given payrun.

- get_srp_wksht

1.23.1 Data Structure Specifications

The following data structures are used in CN_SRP_PMT_PLANS_PUB:

- Wksht_rec
- Wksht_tbl

1.23.1.1 WKSHT_REC

```
TYPE Wksht_rec IS RECORD
(
  payment_worksheet_id cn_payment_worksheets.payment_worksheet_id%TYPE,
  salesrep_id          cn_salesreps.salesrep_id%TYPE,
  salesrep_name        cn_salesreps.name%TYPE,
  resource_id          cn_salesreps.resource_id%TYPE,
  employee_number      cn_salesreps.employee_number%TYPE,
  current_earnings     NUMBER,
  pmt_amount_earnings  NUMBER,
  pmt_amount_diff      NUMBER,
  pmt_amount_adj       NUMBER,
  pmt_amount_adj_rec   NUMBER,
  Pmt_amount_total     NUMBER,
  held_amount          NUMBER,
  worksheet_status     cn_lookups.meaning%TYPE,
  worksheet_status_code cn_lookups.lookup_code%TYPE,
  Analyst_name         cn_salesreps.assigned_to_user_name%TYPE,
  object_version_number NUMBER,
  view_notes           VARCHAR2(1),
  view_ced             VARCHAR2(1),
  status_by            fnd_user.user_name%TYPE,
  cost_center          cn_salesreps.cost_center%TYPE,
  charge_to_cost_center cn_salesreps.charge_to_cost_center%TYPE,
  notes               cn_lookups.meaning%TYPE
);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–162 Parameters

Parameter	Data Type	Description
payment_worksheet_id	NUMBER (15, 0)	Payment worksheet ID
salesrep_id	NUMBER	Salesrep ID
salesrep_name	VARCHAR2 (360)	Salesrep name
resource_id	NUMBER	Resource ID
employee_number	VARCHAR2 (30)	Employee number
current_earnings	NUMBER	Current earnings
pmt_amount_earnings	NUMBER	Payment amount earnings
pmt_amount_diff	NUMBER	Payment amount difference
pmt_amount_adj	NUMBER	Adjusted payment amount
pmt_amount_adj_rec	NUMBER	Adjusted recoverable payment amount
Pmt_amount_total	NUMBER	Total payment amount
held_amount	NUMBER	Held amount
worksheet_status	VARCHAR2 (80)	Worksheet status
worksheet_status_code	VARCHAR2 (30)	Worksheet status code
Analyst_name	VARCHAR2 (100)	Payment analyst name
object_version_number	NUMBER	Object version number
view_notes	VARCHAR2 (1)	View notes
view_ced	VARCHAR2 (1)	View current earnings due
status_by	VARCHAR2 (100)	Status by
cost_center	VARCHAR2 (30)	Cost center
charge_to_cost_center	VARCHAR2 (30)	Charge to cost center
notes	VARCHAR2 (80)	Notes

1.23.1.2 WKSHT_TBL

TYPE wksht_tbl IS TABLE OF wksht_rec INDEX BY BINARY_INTEGER;

1.23.2 get_srp_wksht

This procedure is used to get a salesrep's transactions for a given payrun.

Procedure Specification

```
PROCEDURE  get_srp_wksht
(
  p_api_versionIN NUMBER,
  p_init_msg_list      IN  VARCHAR2,
  p_commit             IN  VARCHAR2,
  p_validation_level   IN  NUMBER,
  x_return_status      OUT NOCOPY VARCHAR2,
  x_msg_count          OUT NOCOPY NUMBER,
  x_msg_data           OUT NOCOPY VARCHAR2,
  p_start_record       IN  NUMBER,
  p_increment_count    IN  NUMBER,
  p_payrun_id          IN  NUMBER,
  p_salesrep_name      IN  VARCHAR2,
  p_employee_number    IN  VARCHAR2,
  p_analyst_name       IN  VARCHAR2,
  p_my_analyst         IN  VARCHAR2,
  p_unassigned         IN  VARCHAR2,
  p_worksheet_status   IN  VARCHAR2,
  p_currency_code      IN  VARCHAR2,
  p_order_by          IN  VARCHAR2,
  x_wksht_tbl          OUT NOCOPY wksht_tbl,
  x_tot_amount_earnings OUT NOCOPY NUMBER,
  x_tot_amount_adj      OUT NOCOPY NUMBER,
  x_tot_amount_adj_rec  OUT NOCOPY NUMBER,
  x_tot_amount_total    OUT NOCOPY NUMBER,
  x_tot_held_amount     OUT NOCOPY NUMBER,
  x_tot_ced             OUT NOCOPY NUMBER,
  x_tot_earn_diff       OUT NOCOPY NUMBER,
  x_total_records       OUT NOCOPY NUMBER
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–163 get_srp_wksht IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.

Table 1–163 *get_srp_wksht IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	N	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_Payrun_id	cn_payruns.id%	Y	

The following table describes the OUT parameters associated with this API.

Table 1–164 *get_srp_wksht OUT Parameters*

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (30)	Detailed error code returned from procedure

1.24 Package CN_SRP_PAYGROUP_PUB

The procedure in this package can be used to assign salesreps to a pay group and update that assignment. They can also be used for mass assignment and mass update of pay groups to salesreps.

- Assign_salesreps
- Create_Mass_Asgn_Srp_Pay
- Update_srp_assignment
- Update_Mass_Asgn_Srp_Pay

1.24.1 Data Structure Specifications

The following data structures are used in Package CN_PAYGROUP_PUB:

- PAYGROUP_ASSIGN_REC

1.24.1.1 PAYGROUP_ASSIGN_REC

```
TYPE PayGroup_assign_rec IS RECORD
(   pay_group_name      cn_pay_groups.name%TYPE           := cn_api.g_miss_char,
    employee_type       VARCHAR2(30)                     := cn_api.g_miss_char,
    employee_number      cn_salesreps.employee_number%TYPE := cn_api.g_miss_char,
    source_id            cn_salesreps.source_id%TYPE       := null,
    assignment_start_date cn_srp_pay_groups.start_date%TYPE := cn_api.g_miss_
date,
    assignment_end_date  cn_srp_pay_groups.end_date%TYPE   := cn_api.g_miss_date,
    lock_flag            cn_srp_pay_groups.lock_flag%TYPE  := cn_api.g_miss_char,
    role_pay_group_id    cn_srp_pay_groups.role_pay_group_id%TYPE := cn_api.g_
miss_id,
    attribute_category   cn_srp_pay_groups.attribute_category%TYPE
                        := cn_api.g_miss_char,
    attribute1           cn_srp_pay_groups.attribute1%TYPE
                        := cn_api.g_miss_char,
    attribute2           cn_srp_pay_groups.attribute2%TYPE
                        := cn_api.g_miss_char,
    attribute3           cn_srp_pay_groups.attribute3%TYPE
                        := cn_api.g_miss_char,
    attribute4           cn_srp_pay_groups.attribute4%TYPE
                        := cn_api.g_miss_char,
    attribute5           cn_srp_pay_groups.attribute5%TYPE
                        := cn_api.g_miss_char,
    attribute6           cn_srp_pay_groups.attribute6%TYPE
                        := cn_api.g_miss_char,
    attribute7           cn_srp_pay_groups.attribute7%TYPE
                        := cn_api.g_miss_char,
    attribute8           cn_srp_pay_groups.attribute8%TYPE
                        := cn_api.g_miss_char,
    attribute9           cn_srp_pay_groups.attribute9%TYPE
                        := cn_api.g_miss_char,
    attribute10          cn_srp_pay_groups.attribute10%TYPE
                        := cn_api.g_miss_char,
    attribute11          cn_srp_pay_groups.attribute11%TYPE
                        := cn_api.g_miss_char,
    attribute12          cn_srp_pay_groups.attribute12%TYPE
                        := cn_api.g_miss_char,
    attribute13          cn_srp_pay_groups.attribute13%TYPE
                        := cn_api.g_miss_char,
    attribute14          cn_srp_pay_groups.attribute14%TYPE
                        := cn_api.g_miss_char,
    attribute15          cn_srp_pay_groups.attribute15%TYPE
                        := cn_api.g_miss_char);
```

Parameter Descriptions

The following table describes the parameters associated with this data structure.

Table 1–165 Parameters

Parameter	Data Type	Description
pay_group_name	VARCHAR2 (80)	Pay group name
employee_type	VARCHAR2 (30)	Employee type
employee_number	VARCHAR2 (30)	Employee number
source_id	NUMBER	Source ID
assignment_start_date	DATE	Assignment start date
assignment_end_date	DATE	Assignment end date
lock_flag	VARCHAR2 (1)	Lock flag
role_pay_group_id	NUMBER (15, 0)	Role pay group ID
attribute_category	VARCHAR2 (30)	Attribute category
Attribute 1	VARCHAR2 (150)	Standard attribute columns
Attribute 2	VARCHAR2 (150)	Standard attribute columns
Attribute 3	VARCHAR2 (150)	Standard attribute columns
Attribute 4	VARCHAR2 (150)	Standard attribute columns
Attribute 5	VARCHAR2 (150)	Standard attribute columns
Attribute 6	VARCHAR2 (150)	Standard attribute columns
Attribute 7	VARCHAR2 (150)	Standard attribute columns
Attribute 8	VARCHAR2 (150)	Standard attribute columns
Attribute 9	VARCHAR2 (150)	Standard attribute columns
Attribute 10	VARCHAR2 (150)	Standard attribute columns
Attribute 11	VARCHAR2 (150)	Standard attribute columns
Attribute 12	VARCHAR2 (150)	Standard attribute columns
Attribute 13	VARCHAR2 (150)	Standard attribute columns
Attribute 14	VARCHAR2 (150)	Standard attribute columns
Attribute 15	VARCHAR2 (150)	Standard attribute columns

1.24.2 Assign_Salesreps

This procedure is used to create entry into cn_pay_groups. It validates the input for a pay group and creates one if all validations are passed.

The following validations are performed by this API:

- Checks if the pay group exists
- Checks if the salesrep exists
- Checks that the assignments do not overlap

Procedure Specification

```
PROCEDURE Assign_salesreps
( p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := cn_api.g_false,
  p_commit               IN  VARCHAR2 := cn_api.g_false,
  p_validation_level     IN  NUMBER   := cn_api.g_valid_level_full,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count            OUT NOCOPY NUMBER,
  x_msg_data             OUT NOCOPY VARCHAR2,
  p_paygroup_assign_rec  IN  PayGroup_assign_rec,
  x_loading_status       OUT NOCOPY  VARCHAR2,
  x_status               OUT NOCOPY  VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–166 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.

Table 1–166 Assign_Salesreps IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_PayGroup_Assign_rec	PayGroup_assign_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–167 Assign_Salesreps OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.24.3 Create_Mass_Asgn_Srp_Pay_Group

This procedure is used to create a new mass payment plan assignment to a salesrep.

Procedure Specification

```

PROCEDURE Create_Mass_Asgn_Srp_Pay
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit               IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level     IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_srp_role_id          IN  NUMBER,
  p_role_pay_group_id    IN  NUMBER,
  x_srp_pay_group_id     OUT NOCOPY  NUMBER,
  x_loading_status       OUT NOCOPY  VARCHAR2
);

```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–168 Create_Mass_Asgn_Srp_Pay_Group IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_role_pay_group_id	NUMBER	Y	
p_srp_role_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–169 Create_Mass_Asgn_Srp_Pay_Group OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.24.4 Update_salesrep_assignment

This procedure is used to create entry into cn_pay_groups. It validates the input for a pay group and creates one if all validations are passed.

The following are the validations performed by this API:

- Checks if the old and the new pay group parameters are valid
- Checks if the assignment dates are valid

- Checks that the assignments do not overlap

Procedure Specification

```
PROCEDURE Update_srp_assignment
(
  p_api_version          IN    NUMBER,
  p_init_msg_list        IN    VARCHAR2 := cn_api.g_false,
  p_commit               IN    VARCHAR2 := cn_api.g_false,
  p_validation_level     IN    NUMBER   := cn_api.g_valid_level_full,
  x_return_status        OUT NOCOPY VARCHAR2,
  x_msg_count            OUT NOCOPY NUMBER,
  x_msg_data             OUT NOCOPY VARCHAR2,
  p_old_paygroup_assign_rec IN    PayGroup_assign_rec,
  p_paygroup_assign_rec  IN    PayGroup_assign_rec,
  p_ovn                  IN    NUMBER,
  x_loading_status       OUT NOCOPY   VARCHAR2,
  x_status               OUT NOCOPY   VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–170 Update_salesrep_assignment IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_old_paygroup_assign_rec	PayGroup_assign_rec	Y	
p_paygroup_assign_rec	PayGroup_assign_rec	Y	

The following table describes the OUT parameters associated with this API.

Table 1–171 Update_salesrep_assignment OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure
x_status	VARCHAR2 (50)	Return Sql Statement Status (VALID/INVALID)

1.24.5 Update_Mass_Asgn_Srp_Pay

This procedure is used to mass update the salesrep assignments to a pay group.

Procedure Specification

```
PROCEDURE Update_Mass_Asgn_Srp_Pay
(
  p_api_version          IN  NUMBER,
  p_init_msg_list        IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit               IN  VARCHAR2 := FND_API.G_FALSE,
  p_validation_level     IN  NUMBER   := FND_API.G_VALID_LEVEL_FULL,
  x_return_status        OUT NOCOPY  VARCHAR2,
  x_msg_count            OUT NOCOPY  NUMBER,
  x_msg_data             OUT NOCOPY  VARCHAR2,
  p_srp_role_id          IN  NUMBER,
  p_role_pay_group_id    IN  NUMBER,
  x_srp_pay_group_id     OUT NOCOPY  NUMBER,
  x_loading_status       OUT NOCOPY  VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

Table 1–172 Create_Mass_Asgn_Srp_Pay_Group IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Y	Standard IN parameters.

Table 1–172 Create_Mass_Asgn_Srp_Pay_Group IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
p_init_msg_list	VARCHAR2	N	Standard IN parameters.
p_commit	VARCHAR2	Y	Standard IN parameters.
p_validation_level	NUMBER	N	Standard IN parameters.
p_srp_role_id	NUMBER	Y	
p_role_pay_group_id	NUMBER	Y	

The following table describes the OUT parameters associated with this API.

Table 1–173 Create_Mass_Asgn_Srp_Pay_Group OUT Parameters

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2 (1)	Standard OUT parameters.
x_msg_count	NUMBER	Standard OUT parameters.
x_msg_data	VARCHAR2 (2000)	Standard OUT parameters.
x_srp_pay_group	NUMBER	
x_loading_status	VARCHAR2 (50)	Detailed error code returned from procedure

1.25 Messages and Notifications

The status messages associated with the Oracle Incentive Compensation public APIs are divided into the following categories:

- [Common Messages](#)
- [CN_COMP_PLAN_PUB Messages](#)
- [CN_ROLE_PLANS_PUB Messages](#)
- [CN_COMMISSION_CALC_PUB Messages](#)
- [CN_PLAN_ELEMENT_PUB Messages](#)
- [CN_SCA_CREDITS_ONLINE_PUB Messages](#)
- [CN_SCA_WF_PXG Messages](#)

- [CN_PRD_QUOTA_PUB Messages](#)
- ["CN_SRP_PRD_QUOTA_PUB Messages"](#)
- ["CN_PAYGROUP_PUB Messages"](#)
- ["CN_PMTPLAN_PUB Messages"](#)
- ["CN_PMTSUB_PUB Messages"](#)
- ["CN_SRP_PMT_PLANS_PUB Messages"](#)
- ["CN_WKSHT_GET_PUB Messages"](#)
- ["CN_SRP_PAYGROUP_PUB Messages"](#)

If an API does not have any error messages or notifications specific to it, then it is not listed as a separate heading following the Common Messages table.

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

1.25.1 Common Messages

The following table describes a lists of common error messages and notifications that the Oracle Incentive Compensation API can generate. Note that not all messages are returned by all APIs.

Table 1–174 Oracle Incentive Compensation API Messages

Number	Type	Name	Text
0	NOTE	ALL_PROCESS_DONE_FAIL_LOG	Process completed with errors. Process log number is &AUDIT_ID
0	NOTE	ALL_PROCESS_DONE_OK_LOG	Process completed successfully. Process log number is &AUDIT_ID
0	ERROR	CNSBCS_NO_ONE_TO_BONUS	No one has complete compensation plan or the specified period is not at the end of the interval.
0	ERROR	CNSBCS_NO_ONE_TO_CALCULATE	No resource with complete compensation plan or the period specified is not at the end of the interval.
0	ERROR	CN_CALC_NO_SALESREP	No salesperson is defined.

Table 1–174 Oracle Incentive Compensation API Messages

Number	Type	Name	Text
0	ERROR	CN_CALC_PE_NO_MATCH	The plan element: "A_NAME is not a bonus type plan element with matching interval type.
0	ERROR	CN_CALC_PLAN_NOT_ASSIGNED	The salesperson who has employee number &EMPLOYEE_NUMBER and employee type &EMPLOYEE_TYPE has not been assigned a compensation plan ever.
0	ERROR	CN_CALC_QUOTA_EXISTS	The plan element: "A_NAME either already exists or duplicates are entered.
0	ERROR	CN_CALC_RESP_NOT_EXIST	The application responsibility: &RESP_NAME does not exist.
0	ERROR	CN_CALC_SALESREP_EXISTS	The sales person who has an employee number &EMPLOYEE_NUMBER and employee type &EMPLOYEE_TYPE either already exists or duplicates are entered.
0	ERROR	CN_CALC_SUB_EXISTS	The calculation submission batch: &BATCH_NAME already exists.
0	ERROR	CN_CALC_SUB_NOT_EXIST	The calculation submission batch: &BATCH_NAME does not exist.
0	ERROR	CN_CALC_SUB_NOT_UPDATEABLE	This calculation submission batch: &BATCH_NAME is not updateable because its status is either completed or in progress.
0	ERROR	CN_CALC_SUB_OPEN_DATE	&DATE is not within an open period.
0	ERROR	CN_CALC_USER_NOT_EXIST	The application user: &USER_NAME does not exist.
0	ERROR	CN_INVALID_DATA	The following field has invalid data : &OBJ_NAME
0	ERROR	PROC_NO_TRX_TO_PROCESS	There are no transactions to process.

1.25.2 CN_COMP_PLAN_PUB Messages

Error messages for CN_COMP_PLAN_PUB are listed below.

Table 1–175 CN_COMP_PLAN_PUB Messages

Number	Type	Name	Text
0	ERROR	PLN_PLAN_DUP_REV_CLASS	Compensation Plan : &PLAN_NAME A revenue class has been assigned to this compensation plan multiple times. Review the revenue classes for duplicates. Also ensure that you have not assigned a revenue class that is a child value of another assigned revenue class.
0	ERROR	PLN_QUOTA_ASSIGNED	This plan element is already assigned to the compensation plan.
0	ERROR	CN_CP_NOT_CONSISTENT	The data for this compensation plan is not consistent with the plan in the database. Compensation Plan Name : &CP_NAME Data Name : &OBJ_NAME
0	ERROR	CN_CP_NOT_EXIST	This compensation plan does not exist in the database. Compensation Plan Name : &CP_NAME
0	ERROR	CN_INVALID_DATA	The following field has invalid data : &OBJ_NAME
0	ERROR	CN_INVALID_DIM_UOM	The dimension type value is invalid.
0	ERROR	CN_EXP_NOT_EXIST	The expression given does not exist.
0	ERROR	CN_RATE_DIMENSION_NOT_EXIST	The rate dimension does not exist.
0	ERROR	CN_RATE_DIM_ASSIGN_NOT_EXIST	The rate dimension is not assigned to the rate schedule.
0	ERROR	CN_RATE_DIM_TIER_NOT_EXIST	The rate dimension does not have the given rate tier.
0	ERROR	CN_RATE_SCHEDULE_NOT_EXIST	The rate schedule does not exist in the database.
0	ERROR	CN_INVALID_TIER	The rate dimension tier is invalid.

1.25.3 CN_ROLE_PLANS_PUB Messages

Error messages for CN_ROLE_PLAN_PUB are listed below.

Table 1–176 CN_ROLE_PLANS_PUB Messages

Number	Type	Name	Text
1	ERROR	CN_RL_DEL_ROLE_PLAN_NOT_EXIST	The assignment record you are about to delete does not exist.
2	ERROR	CN_RL_ASGN_ROLE_NOT_EXIST	The role in the assignment does not exist.Role Name : &ROLE_NAME.
3	ERROR	CN_RL_ASGN_CP_NOT_EXIST	The compensation plan in the assignment does not exist.Compensation Plan : &COMP_PLAN.
4	ERROR	CN_RL_CP_DATE_RANGE_NOT_WITHIN	The assignment period should be within the period of the assigned compensation plan. Please check the assignment start date and end date. Assignment Start Date: &START_DATE Assignment End Date: &END_DATE Compensation Plan: &COMP_PLAN_NAME Compensation Plan Start Date: &CP_START_DATE Compensation Plan End Date: &CP_END_DATE Please note that an empty end date means the ending is open.
5	ERROR	CN_RL_ROLE_PLAN_OVERLAP	The date range of the assignment overlaps with the following record. Compensation Plan: &COMP_PLAN_NAME Start Date: &START_DATE End Date : &END_DATE A sales role cannot have more than one compensation plan assigned at the same time.
6	ERROR	CN_RL_UPD_ROLE_PLAN_NOT_EXIST	The assignment record you are about to update does not exist.
7	ERROR	CN_RL_INVALID_DATE_RANGE	The start date must be before or the same as the end date. Start date: &START_DATE End date: &END_DATE
8	ERROR	CN_RP_CREATED_IN_SFP	This role plan assignment is created by Incentive Planning, so it cannot be updated.

1.25.4 CN_COMMISSION_CALC_PUB Messages

Error messages for CN_COMMISSION_CALC_PUB are listed below.

Table 1–177 CN_COMMISSION_CALC_PUB Messages

Number	Type	Name	Text
1	ERROR	CN_INVALID_RES_ID	Resource Ids mentioned at &LINE_NO are invalid.
2	ERROR	CN_INVALID_PRD_ID	Period(s) corresponding to Calculation date(s) mentioned at &LINE_NO is/are not opened in OIC Calendar.
3	ERROR	CN_INVALID_RS_ID	Resultsets are not created for the dates mentioned at the &LINE_NO.
4	ERROR	CN_NO_QUOTA_FORMULA	The plan element "A_NAME does not have a valid formula.
5	ERROR	CN_INV_CALC_FORMULA	Formula for the plan element "ANAME is incomplete. Please check to make sure that your formula status is complete and forecast expression is defined.
6	ERROR	CN_FORMULA_PKG_NOT_VALID	The plan element "A_NAME does not have a valid formula.
7	ERROR	CN_QUOTA_NOT_MAPPED	Projection Identifier at &LINE_NO is not mapped to the plan element.
8	ERROR	CN_PR_CP_NOT_VALID	Plan Element at &LINE_NO may not have a compensation plan or the status of the compensation plan is Incomplete.
9	ERROR	CN_SRPLAN_NOT_FOUND	A plan with the selected Plan Element is not assigned to the salesrep.
10	WARNING	CN_CONV_CURR_FAIL	Currency conversion failed. Using functional currency instead of user currency.

1.25.5 CN_PLAN_ELEMENT_PUB Messages

Error messages for CN_PLAN_ELEMENT_PUB are listed below.

Table 1–178 CN_PLAN_ELEMENT_PUB Messages

Number	Type	Name	Text
1	ERROR	PLN_QUOTA_EXISTS	Plan element with this name already exists.
2	ERROR	CN_DUP_PLN_NAME_TOO_LONG	The plan element to be duplicated has a long plan element name &FROM_PE. Will create the duplicate plan element with the following name &TO_PE.
3	ERROR	CN_INVALID_DATA	The following field has invalid data : &OBJ_NAME.
4	ERROR	CN_PLN_NOT_CONSISTENT	The data for this plan element is not consistent with the one in the database. Plan Element Name: PLAN_NAME Data Name : &OBJ_NAME
5	ERROR	CN_PLN_NOT_EXIST	This Plan Element does not exist in the database. Plan Element Name : &PE_NAME
6	ERROR	CN_REV_CLASS_NOT_EXIST	The revenue class does not exist in the database.
7	ERROR	CN_INVALID_DATE_RANGE	Please enter an end date which is after the start date.
8	ERROR	CN_INTERVAL_TYPE_NOT_EXIST	This interval name does not exist in the database. Interval Name: &INTERVAL_NAME
9	ERROR	CN_CREDIT_TYPE_NOT_EXIST	This Credit Type does not exist in the database. Credit Type: &CREDIT_TYPE
10	ERROR	CN_QUOTA_CANNOT_HAVE_RATES	Cannot assign a rate when the quota type is None.
11	ERROR	CN_QUOTA_RULE_NOT_EXIST	Quota rule does not exist in the database for Plan Element Name: &PLAN_NAME Revenue Class Name: &REVENUE_CLASS_NAME
12	ERROR	CN_FORMULA_NOT_EXIST	This formula does not exist in the database. Formula Name: &FORMULA_NAME

1.25.6 CN_SCA_CREDITS_ONLINE_PUB Messages

Error messages for CN_SCA_CREDITS_ONLINE_PUB are listed below.

Table 1–179 CN_SCA_CREDITS_ONLINE_PUB Messages

Number	Type	Name	Text
1	ERROR	CN_INVALID_ORG	The organization is invalid.
2	ERROR	CN_SCA_NO_ROWS_TO_PROCESS	There are no rows to process in the input table.

1.25.7 CN_SCA_WF_PXG Messages

Error messages for CN_SCA_WF_PXG are listed below.

Table 1–180 CN_SCA_WF_PXG Messages

Number	Type	Name	Text
0	ERROR	CN_SCA_WF_INVLD_BATCH_ID	Invalid batch identifier.
0	ERROR	CN_SCA_WF_INVLD_END_HEADER_ID	Invalid end header identifier.
0	ERROR	CN_SCA_WF_INVLD_ITEM_TYPE	Invalid item type &NAME.
0	ERROR	CN_SCA_WF_INVLD_PR_NAME	Invalid process name &NAME.
0	ERROR	CN_SCA_WF_INVLD_ST_HEADER_ID	Invalid start header identifier.
0	ERROR	CN_SCA_WF_INVLD_TRX_SRC	Invalid transaction source.
0	ERROR	CN_SCA_WF_NO_PROFILE_VA	Profile value is undefined.
0	ERROR	CN_SCA_WF_NO_TRX_SRC	Incomplete transaction data.

1.25.8 CN_PRD_QUOTA_PUB Messages

Error messages for CN_PRD_QUOTA_PUB are listed below.

Table 1–181 CN_PRD_QUOTA_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_INPUT_CANT_NULL	The error message can be returned in two cases: 1. If the plan element name is not provided. 2. If the period name is not provided as part of the table data structure.
0	ERROR	CN_PLN_NOT_EXIST	This error message is returned if a plan element with the same name (as provided in the call) does not exist.
0	ERROR	CN_PERIOD_NOT_FOUND	This error message is returned if a period with the given name (in the table) does not exist.
0	ERROR	CN_PERIOD_QUOTA_NOT_EXIST	This error is a fatal error returned when some of the system tables are not populated. This is an exception case that indicates that there was an error in the plan element creation.

1.25.9 CN_SRP_PRD_QUOTA_PUB Messages

Error messages for CN_SRP_PRD_QUOTA_PUB are listed below.

Table 1–182 CN_SRP_PRD_QUOTA_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_INPUT_CANT_NULL	The error message can be returned in six cases: 1. If the plan element name is not provided. 2. If the salesperson name was not provided. 3. If the role name is not provided. 4. If the compensation plan name is not provided. 5. If the start date is not provided. 6. If the period name is not provided as part of the table data structure.
0	ERROR	CN_PLN_NOT_EXIST	This error message is returned if a plan element with the same name (as provided in the call) does not exist.
0	ERROR	CN_SRP_NOT_FOUND	This error message is returned if a salesperson with the name provided can't be found in the system.

Table 1–182 CN_SRP_PRD_QUOTA_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_QM_INVALID_SRPROLE	This error message is returned if a role with the name provided can't be found in the system.
0	ERROR	CN_CP_NOT_EXIST	This error message is returned if a compensation plan with the name provided can't be found in the system.
0	ERROR	CN_SRP_PLN_ASSIGN_NOT_EXIST	This error message is returned if the compensation plan is not assigned to the salesperson in the system.
0	ERROR	CN_QUOTA_ASSIGN_NOT_EXIST	This error message is returned if the plan element has not been assigned to the compensation plan in the system.
0	ERROR	CN_PE_NOT_CUSTOMIZABLE	This error message is returned if the plan element cannot be customized at the salesperson level. Please make sure that the plan can be customized by going through the UI.
0	ERROR	CN_PERIOD_NOT_FOUND	This error message is returned if a period with the given name (in the table) does not exist.
0	ERROR	CN_PERIOD_QUOTA_NOT_EXIST	This error is a fatal error returned when some of the system tables are not populated. This is an exception case that indicates that there was an error in the plan element creation for the salesperson.

1.25.10 CN_PAYGROUP_PUB Messages

Error messages for CN_PAYGROUP_PUB are listed below.

Table 1–183 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_INVALID_DATE_RANGE	Please enter an end date that is after the start date.
0	ERROR	CN_PAY_GROUP_EXISTS	The pay group already exists.
0	ERROR	CN_PAYGROUP_OVERLAPS	Please enter a date range for this pay group which does not overlap with other pay groups with the same name.
0	ERROR	CN_INVALID_PRD_SET	Please enter a valid calendar name.

Table 1–183 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_INVALID_PERIOD_TYPE	Please enter a valid period type.
0	ERROR	CN_RECORD_CHANGED	Record is changed by other users.
0	ERROR	CN_PG_DT_CANNOT_NULL	Please enter a valid date for the start date and end date for the pay group.
0	ERROR	CN_INVALID_PAY_GROUP	Please enter a valid pay group.
0	ERROR	CN_PAY_GROUP_CHANGE_NA	Start and end dates for the pay group must include the start and end dates for all salespeople.
0	ERROR	CN_PAYGRP_PRD_TYPE_NOT_UPD	The pay group period type cannot be updateable because calculation and posting already exists for this pay group.
0	ERROR	CN_PAYGRP_PRD_SNAME_NOT_UPD	The pay group period set name cannot be updateable because calculation and posting already exists for this pay group.
0	ERROR	CN_PAYGRP_PRD_TYPE_NOT_UPDP	The pay group period type cannot be updateable because calculation and posting already exists for this pay group.
0	ERROR	CN_PAY_GROUP_ASSIGNED_TO_ROLE	A pay group has been assigned to the role. Delete Role-Pay Group assignment to proceed.
0	ERROR	CN_PAY_GROUP_ASSIGNED_TO_SRP	This pay group cannot be deleted because it has already been assigned to a salesperson.
0	ERROR	CN_INVALID_SRP_PGRP_ASGN_DT	Please enter a valid start date and end date for the assignment of pay groups to salespeople.
0	ERROR	CN_OVERLAP_SRP_PGRP_ASGN	Please adjust the assignment dates so that the current assignment does not overlap with existing assignments.
0	ERROR	CN_INVALID_PGRP_ASGN_END_DT	Please enter a valid end date for the assignment of pay groups to the salesperson. The end date should not be in a period that has already been calculated

Table 1–183 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_SRP_PAY_GROUPS_EXIST	Please assign a pay group that has not already been assigned to the salesperson for the same date range.
0	ERROR	CN_INVALID_UPD_SRP_PGRP	Please do not update the salesperson in the current assignment. Instead, update the effectivity or assign a new pay group to the salesperson.
0	ERROR	CN_INVALID_PGRP_END_DT	Please specify a valid end date.

1.25.11 CN_PMTPLAN_PUB Messages

Error messages for CN_PMTPLAN_PUB are listed below.

Table 1–184 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_INVALID_PMT_PLAN_FLAGS	Please enter a valid value for Recoverable and Pay Later.
0	ERROR	CN_INVALID_DATE_RANGE	Please enter an end date that is after the start date.
0	ERROR	CN_INVALID_CREDIT_TYPE	Please enter a valid credit type.
0	ERROR	CN_INVALID_PAY_INTERVAL	Please enter a valid pay interval for the payment plan. Payment plans can only have pay intervals of type Period, Quarter and Year.
0	ERROR	CN_SPP_MAX_LT_MIN	The maximum amount must be greater than or equal to the minimum amount.
0	ERROR	CN_PMT_PLAN_EXISTS	Please enter a new payment plan. This payment plan already exists.
0	ERROR	CN_PP_SDT_CANNOT_NULL	Start date for a payment plan cannot be null. Please enter a valid date as the start date for the payment plan.
0	ERROR	CN_INVALID_PMT_PLAN	Please enter a valid value for Recoverable and Pay Later.

Table 1–184 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_REC_FLG_UPD_NA	The recovery flag cannot be updateable because this payment plan has been assigned to a salesrep.
0	ERROR	CN_PMT_PLAN_CHANGE_NA	Update of payment element is not allowed because this payment plan has already been assigned to some salespeople.
0	ERROR	CN_DELETE_NA	This record cannot be deleted.

1.25.12 CN_PMTSUB_PUB Messages

Error messages for CN_PMTSUB_PUB are listed below.

Table 1–185 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_PAYRUN_ALREADY_PAID	This payrun is already paid.
0	ERROR	CN_PAYRUN_DOES_NOT_EXIST	The specified payrun does not exist. Please specify a valid payrun name.

1.25.13 CN_SRP_PMT_PLANS_PUB Messages

Error messages for CN_SRP_PMT_PLANS_PUB are listed below.

Table 1–186 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_PP_NOT_EXIST	Payment plan does not exist in the database \n\nPayment Plan Name: &PP_NAME.
0	ERROR	CN_SRP_NOT_EXIST	This salesperson does not exist in the database
0	ERROR	CN_SRP_PMT_PLAN_EXIST	The same salesperson and payment plan assignment already exists.
0	ERROR	SRP_MUST_ACTIVATE_REP	You cannot assign a plan to an inactive salesperson. You must define the salesperson's first active period.
0	ERROR	CN_SPP_PRDS_NI_SRP_PRDS	The effective dates of the payment plan assignment of the salesperson do not exist within the date range of the active dates of the salesperson.

Table 1–186 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_SPP_PRDS_NI_PMT_PRDS	The effective dates for the salesperson payment plan assigns are not within the date range of the payment plans.
0	ERROR	CN_SRP_PMT_PLAN_OVERLAPS	This salesperson has payment plans overlapped for the date range.
0	ERROR	CN_SRPPP_CT_MISMATCH	The payment plan credit type does not match with the plan elements assigned.
0	ERROR	CN_SPP_MAX_LT_MIN	The Maximum amount must be greater than or equal to the minimum amount.
0	ERROR	CN_SRP_PMT_PLAN_USED	Delete of salesperson payment plan is not allowed. This record already used in worksheet.
0	ERROR	CN_SPP_CANNOT_SHORTEN_ED	End date cannot be reduced because this salesperson payment plan is used in the worksheet.
0	ERROR	CN_SPP_UPDATE_NOT_ALLOWED	Update not allowed. Worksheet already exists for this salesperson payment plan.
0	ERROR	CN_SRP_PMT_PLAN_NOT_EXIST	Payment plan has been changed or has not been assigned to the salesperson.
0	ERROR	CN_INVALID_OBJECT_VERSION	The record has been updated. Please requery the data.
0	ERROR	CN_SRP_PMT_PLAN_LOCKED	The payment plan assignment is locked.

1.25.14 CN_WKSHT_GET_PUB Messages

Error messages for CN_WKSHT_GET_PUB are listed below.

Table 1–187 CN_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN1158_CURR_CONV_ERR	Currency conversion failed. Please ensure that the user currency is set up correctly.
0	ERROR	CN_NO_SRP_ACCESS	You do not have permission to access this salesrep.

1.25.15 CN_SRP_PAYGROUP_PUB Messages

Error messages for CN_SRP_PAYGROUP_PUB are listed below.

Table 1–188 CN_SRP_PAYGROUP_PUB Messages

Number	Type	Name	Text
0	ERROR	CN_SALESREP_NOT_FOUND	This salesperson does not exist in the database.
0	ERROR	CN_SALESREP_TOO_MANY_ROWS	This salesperson has more than one row in the database.
0	ERROR	CN_INVALID_SRP_PGRP_ASGN_DT	Please enter a valid start date and end date for the assignment of pay groups to salespeople.
0	ERROR	CN_OVERLAP_SRP_PGRP_ASGN	Please adjust the assignment dates so that the current assignment does not overlap with existing assignments.
0	ERROR	CN_INVALID_PAY_GROUP	Please enter a valid pay group.
0	ERROR	CN_SRP_PAY_GROUPS_EXIST	Please assign a pay group that has not already been assigned to the salesperson for the same date range.
0	ERROR	CN_INVALID_SRP_PGRP_ASGN	Please enter a valid start date and end date for the assignment of pay groups to salespeople.
0	ERROR	CN_RECORD_UPDATED	The record has been updated. Please requery the data.
0	ERROR	CN_INVALID_UPD_SRP_PGRP	Please do not update the salesperson in the current assignment. Instead, update the effectivity or assign a new pay group to the salesperson.
0	ERROR	CN_OVERLAP_SRP_PGRP_ASGN	Please adjust the assignment dates so that the current assignment does not overlap with existing assignments.
0	ERROR	CN_PA_ASGN_DATE	Resource is a payee and is assigned to other resources during period being deleted.
0	ERROR	CN_SRP_PG_WS	Resource has been included in a worksheet for the specified period or pay group has been locked.
0	ERROR	CN_SPG_CANNOT_SHORTEN_ED	End date cannot be reduced because this salesperson's pay group is used in the worksheet.
0	ERROR	CN_SPG_UPDATE_NOT_ALLOWED	Update not allowed. Worksheet already exists for this salesperson's pay group.

API Examples

This appendix contains three examples of how APIs can be used in Oracle Incentive Compensation. These examples are for demonstration purposes only; do not attempt to use them exactly as they are presented. Each example contains a link to the API in the main document for your reference.

A.1 Assigning Payment Plans to Resources

This is an example of how an API is used to assign payment plans to resources. See [Section 1.21, "Package CN_SRP_PMT_PLANS_PUB"](#) for more information.

Example

```
DECLARE

g_org_id                NUMBER := to_number(fnd_profile.value('ORG_
ID'));
g_plan_name              VARCHAR2(60) := 'PaymentPlan';

CURSOR Assign_cur is
  select /*+ ORDERED */
    jrs.salesrep_id
  ,jrs.salesrep_number emp_num
  ,jrs.start_date_active actual_start_date
  ,greatest(jrs.start_date_active, trunc(sysdate, 'YY')) start_date
  ,jrs.end_date_active end_date
  , 'OTHER' SrpType
  from jtf_rs_salesreps jrs
  ,jtf_rs_resource_extns jrre
 where not exists
    (select 0
     from cn_srp_pmt_plans cspp
```

```

        ,cn_pmt_plans cppa
      where cspp.salesrep_id = jrs.salesrep_id
        and cppa.pmt_plan_id = cspp.pmt_plan_id
        and cppa.name = g_plan_name)
    and jrs.org_id = g_org_id
    and jrs.resource_id = jrre.resource_id
    and jrre.attribute7 = 'I';

Assign_rec Assign_cur%rowtype;
l_commit_ctr number := 0;
l_ctr number := 0;
l_ctr_jrre number := 0;
g_api_version          NUMBER := 1;
g_error_Mesg_Concat    VARCHAR2(4000) := NULL; -- This Var is used only
by Update API(s)
l_ctr_jrs NUMBER := 0;
l_ctr_others NUMBER := 0;
l_x_return_status      VARCHAR2(1) := NULL;
l_x_msg_count          NUMBER := NULL;
l_x_msg_data           VARCHAR2(2000) := NULL;
l_msg_index_out        VARCHAR2(2000) := NULL;
v_file                 utl_file.file_type;
v_file_name            varchar2(100) := 'AssignPaymentPlans'||to_char(sysdate,
'YYYYMMDDSS');
v_dir                  varchar2(100) := '/OracleApp/oiccomm/temp';
v_buff                 constant INTEGER := 10000;
l_obj_ver_res          NUMBER;
l_obj_ver_srp          NUMBER;
pmt_srp_rec            CN_SRP_PMT_PLANS_PUB.srp_pmt_plans_rec_type;
l_srp_pmt_plan_id      NUMBER;
l_loading_status       VARCHAR2(40);

BEGIN

    fnd_client_info.set_org_context(g_org_id);

    select value into v_dir from v$parameter where name = 'utl_file_dir';
    v_file := UTL_FILE.FOPEN(v_dir, v_file_name, 'W', v_buff);

    FOR Assign_rec in Assign_cur LOOP

        l_ctr := l_ctr + 1;

        /** Initializing Variables **/

```

```

l_x_return_status := null;
l_x_msg_count := null;
l_x_msg_data := null;
l_msg_index_out := null;
l_srp_pmt_plan_id := null;
g_error_Mesg_Concat := NULL;
l_loading_status := NULL;

/** Building the Payment Plan Record **/

pmt_srp_rec.pmt_plan_name := g_plan_name;
pmt_srp_rec.salesrep_type := Assign_rec.SrpType;
pmt_srp_rec.emp_num := Assign_rec.emp_num;
pmt_srp_rec.start_date := Assign_rec.start_date;
pmt_srp_rec.end_date := Assign_rec.end_date;
pmt_srp_rec.minimum_amount := 0;
pmt_srp_rec.maximum_amount := null;
pmt_srp_rec.max_recovery_amount := null;

/** Calling the Assign Payment Plan API **/

BEGIN

    CN_SRP_PMT_PLANS_PUB.CREATE_SRP_PMT_PLAN
        (P_API_VERSION          => g_api_version,
         P_SRP_PMT_PLANS_REC    => pmt_srp_rec,
         X_RETURN_STATUS        => l_x_return_status,
         X_MSG_COUNT            => l_x_msg_count,
         X_MSG_DATA             => l_x_msg_data,
         X_SRP_PMT_PLAN_ID      => l_srp_pmt_plan_id,
         X_LOADING_STATUS       => l_loading_status);

    IF l_x_return_status <> 'S' THEN

        IF (FND_MSG_PUB.Count_Msg <> 0 ) THEN
            FOR I IN 1..FND_MSG_PUB.Count_Msg LOOP
                fnd_msg_pub.get(p_msg_index => I
                               ,p_encoded => FND_API.g_false
                               ,p_data => l_x_msg_data
                               ,p_msg_index_out => l_msg_index_out);
                g_error_Mesg_Concat := g_error_Mesg_Concat || ' ' || l_msg_
index_out;
            END LOOP;
        END IF;

```

```

        l_ctr_jrre := l_ctr_jrre + 1;

        g_error_Mesg_Concat := substr(l_x_msg_data, 1, 500);

        utl_file.put_line(v_file, 'Error : ' || g_error_Mesg_Concat);
        utl_file.put_line(v_file,
'-----');

        ELSIF l_x_return_status = 'S' THEN

            update JTF_RS_SALESREPS
                set ATTRIBUTE15 = 'S'
                where SALESREP_ID = assign_rec.salesrep_id;

        END IF;

    EXCEPTION
    WHEN OTHERS THEN
        l_ctr_others := l_ctr_others + 1;
        utl_file.put_line(v_file, 'Error Occured While Processing Salesrep Number:
' || assign_rec.emp_num);
        utl_file.put_line(v_file, 'Error : ' || sqlcode || ':' || sqlerrm);
        utl_file.put_line(v_file, '-----');

    END;

    IF l_commit_ctr = 100 THEN
        COMMIT;
        l_commit_ctr := 0;
    ELSE
        l_commit_ctr := l_commit_ctr + 1;
    END IF;

END LOOP;

COMMIT;

utl_file.put_line(v_file, 'Total Number of Records Processed = ' || l_ctr);
utl_file.put_line(v_file, 'Total Number of API Errors = ' || l_ctr_jrre);
utl_file.put_line(v_file, 'Total Number of When Others Errors = ' || l_ctr_
others);

```

```

        utl_file.fclose(v_file);

EXCEPTION
    when UTL_FILE.INVALID_PATH then
        raise_application_error(-20101, 'Invalid Path');
    when UTL_FILE.INVALID_MODE then
        raise_application_error(-20102, 'Invalid Mode');
    when UTL_FILE.INVALID_MAXLINESIZE then
        raise_application_error(-20103, 'Invalid Max Line Size');
    when UTL_FILE.INVALID_FILEHANDLE then
        raise_application_error(-20104, 'Invalid File Handle');
    when UTL_FILE.INVALID_OPERATION then
        raise_application_error(-20105, 'Invalid Operation');
    when UTL_FILE.WRITE_ERROR then
        raise_application_error(-20106, 'Erite Error');
    when others then
        raise_application_error(-20107, 'When Others:
'||sqlcode||':'||sqlerrm);

END;
/

```

A.2 Updating Resources' Pay Group Assignment

This is an example of how an API is used to update resources' pay group assignments. See [Section 1.24, "Package CN_SRP_PAYGROUP_PUB"](#) for more information.

Example

```

DECLARE
    mySysDate    CONSTANT DATE           := SYSDATE;
    myUserID     CONSTANT NUMBER         := TO_NUMBER(FND_PROFILE.VALUE('USER_ID'
));
    myLoginID    CONSTANT NUMBER         := TO_NUMBER(FND_PROFILE.VALUE('LOGIN_
ID'));
    myOrgID      CONSTANT NUMBER(15)     := TO_NUMBER(FND_PROFILE.VALUE('ORG_ID'
));
    myLang       CONSTANT VARCHAR2(4)    := USERENV('LANG');
    myUserName   VARCHAR2(100);

CURSOR c1 IS
    SELECT a.salesrep_id, a.start_date, a.pay_group_id

```

```

FROM CN_SRP_PAY_GROUPS_ALL a
WHERE a.START_DATE = '01-APR-02'
      and a.end_date is not null
      and not exists
      (select 0
       from cn_srp_pay_groups_all b
       where b.salesrep_id = a.salesrep_id
            and b.pay_group_id != a.pay_group_id);

CURSOR c2(p_salesrep_id NUMBER) IS
  SELECT cspga.pay_group_id, cpga.name, cspga.start_date, cspga.end_date,
 jrs.salesrep_number, jrre.category
  FROM cn_srp_pay_groups_all cspga,
       cn_pay_groups_all cpga,
       jtf.jtf_rs_salesreps jrs,
       jtf.jtf_rs_resource_extns jrre
 WHERE cspga.salesrep_id = p_salesrep_id
       AND jrre.resource_id = jrs.resource_id
       AND cspga.pay_group_id = cpga.pay_group_id
       AND jrs.salesrep_id = cspga.salesrep_id;

g_sales_credit_type      NUMBER;
g_org_id                 NUMBER;
g_api_version            NUMBER := 1.0;
g_error_Mesg_Concat      VARCHAR2(4000); -- This Var is used only by
Update API(s)
  l_x_return_status      VARCHAR2(1);
  l_x_msg_count          NUMBER;
  l_x_msg_data           VARCHAR2(2000);
  l_x_salesrep_id        NUMBER;
  l_tot_recs            NUMBER; -- Used to count tot Number of Reps
to be created
  l_ctr                  NUMBER := 0; -- total number of records
picked up
  l_ctr_others           NUMBER := 0; -- total number of records
errored out
  l_ctr_commit           NUMBER := 0; -- total number of records
updated
  l_pay_grp_err          NUMBER := 0;

  l_pay_group_id         NUMBER;
  l_paygrp_name          VARCHAR2(60);
  l_x_status             VARCHAR2(50); -- Used only by Paygroup API
  l_x_loading_status     VARCHAR2(100); -- Used only by Paygroup API
  l_pay_grp_start_date   DATE;

```

```

l_msg_index_out          VARCHAR2(2000);
l_error_code             VARCHAR2(100);
l_error_flag             VARCHAR2(5);
v_file                   UTL_FILE.file_type;
v_file_name              VARCHAR2 (100) := 'reassign_paygroups' ||TO_CHAR
(SYSDATE, 'YYYYMMDDSS');
v_dir                    VARCHAR2 (100);
v_buff                   CONSTANT INTEGER          := 10000;

new_paygrp_assgn_rec      cn_srp_paygroup_pub.PayGroup_assign_rec; --
Record Type for Pay Group Assignment
old_paygrp_assgn_rec      cn_srp_paygroup_pub.PayGroup_assign_rec; --
Record Type for Pay Group Assignment
workrec2                  c2%ROWTYPE;

BEGIN

    fnd_client_info.set_org_context(myorgid);

    /** Finding the Utl_file_dir parameter value from v$parameter **/

    select value into v_dir from v$parameter where name = 'utl_file_dir';

    v_file := UTL_FILE.fopen (v_dir, v_file_name, 'W', v_buff);
    UTL_FILE.put_line (v_file,
'-----
--');

    /** Initializing Outer Variables Variables **/

    FOR workrec1 IN c1 LOOP
    BEGIN
        l_msg_index_out := NULL;
        g_error_Mesg_Concat := NULL;
        l_x_msg_data := NULL;

        OPEN c2(workrec1.salesrep_id);
        FETCH c2 INTO workrec2;
        CLOSE c2;

        old_paygrp_assgn_rec.pay_group_name := workrec2.NAME;
        old_paygrp_assgn_rec.assignment_start_date := workrec2.start_date;
        old_paygrp_assgn_rec.assignment_end_date := Workrec2.end_date;
        old_paygrp_assgn_rec.employee_type := workrec2.category;

```

```

old_paygrp_assgn_rec.employee_number := workrec2.salesrep_number;

new_paygrp_assgn_rec.pay_group_name := workrec2.NAME;
new_paygrp_assgn_rec.assignment_start_date := TRUNC(SYSDATE, 'YY');
new_paygrp_assgn_rec.assignment_end_date := '31-DEC-2005';
new_paygrp_assgn_rec.employee_type := workrec2.category;
new_paygrp_assgn_rec.employee_number := workrec2.salesrep_number;

CN_SRP_PAYGROUP_PUB.UPDATE_SRP_ASSIGNMENT(P_API_VERSION => g_api_version,
                                           X_RETURN_STATUS => l_x_return_status,
                                           X_MSG_COUNT => l_x_msg_count,
                                           X_MSG_DATA => l_x_msg_data,
                                           P_OLD_PAYGROUP_ASSIGN_REC => old_
paygrp_assgn_rec,
                                           P_PAYGROUP_ASSIGN_REC => new_paygrp_
assgn_rec,
                                           X_LOADING_STATUS => l_x_loading_
status,
                                           X_STATUS => l_x_status );

-- Since this has 2 more status include them in error message too
IF l_x_return_status <> 'S' THEN

    IF (FND_MSG_PUB.Count_Msg <> 0 ) THEN
        FOR I IN 1..FND_MSG_PUB.Count_Msg LOOP
            fnd_msg_pub.get(p_msg_index => I
                           ,p_encoded => FND_API.g_false
                           ,p_data => l_x_msg_data
                           ,p_msg_index_out => l_msg_index_out);
            g_error_Mesg_Concat := g_error_Mesg_Concat||' '||l_msg_
index_out;
        END LOOP;
    END IF;

    l_error_code := 'API_ASSIGN_PAYGRP_ERROR_MSG_STACK';
    l_error_flag := 'Y';
    g_error_Mesg_Concat := substr(l_x_status||' '||l_x_msg_data,
1, 500);

    utl_file.put_line(v_file, 'Error Occured While Processing
Salesrep : '||workrec2.salesrep_number );
    utl_file.put_line(v_file, '|| g_error_Mesg_Concat);

```

```

        l_pay_grp_err := l_pay_grp_err + 1;
    --    END IF;
    ELSE
        l_ctr_commit := l_ctr_commit + 1;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        l_ctr_others := l_ctr_others + 1;
        utl_file.put_line(v_file, 'Error Occured While Processing Salesrep :
' || workrec2.salesrep_number);
        utl_file.put_line(v_file, 'Error : ' || sqlcode || ':' || sqlerrm);

END;

        l_ctr := l_ctr + 1;

        IF MOD(l_ctr_commit, 1000) = 0 THEN
            COMMIT;
        --    NULL;
        END IF;
END LOOP;

UTL_FILE.put_line (v_file, 'Total Number of Records Picked for Processed = ' ||
l_ctr);
    UTL_FILE.put_line (v_file, 'Total Number of Records Picked Successfully
Processed = ' || l_ctr_commit);
    UTL_FILE.put_line (v_file, 'Total Number of Pay Group Errors = ' || l_pay_
grp_err);
    UTL_FILE.put_line (v_file, 'Total Number of When Others Errors = ' || l_ctr_
others);

--ROLLBACK;
COMMIT;

UTL_FILE.fclose (v_file);

EXCEPTION
    WHEN UTL_FILE.invalid_path THEN
        raise_application_error (-20101, 'Invalid Path');
    WHEN UTL_FILE.invalid_mode THEN
        raise_application_error (-20102, 'Invalid Mode');
    WHEN UTL_FILE.invalid_maxlinesize THEN
        raise_application_error (-20103, 'Invalid Max Line Size');
    WHEN UTL_FILE.invalid_filehandle THEN
        raise_application_error (-20104, 'Invalid File Handle');
    WHEN UTL_FILE.invalid_operation THEN

```

```

        raise_application_error (-20105, 'Invalid Operation');
    WHEN UTL_FILE.write_error THEN
        raise_application_error (-20106, 'Write Error');
    WHEN OTHERS THEN
        raise_application_error (-20107, 'When Others: ' || SQLCODE || ':' ||
SQLERRM);
    END;
/

```

A.3 Assigning Compensation Plans to Roles

This example can be used to assign or deassign compensation plans to roles. See [Section 1.5, "Package CN_ROLE_PLANS_PUB"](#) for more information.

Example

set serveroutput on

DECLARE

```

    mySysDate  CONSTANT DATE      := SYSDATE;
    myUserID   CONSTANT NUMBER    := TO_NUMBER(FND_
PROFILE.VALUE('USER_ID' ));
    myLoginID  CONSTANT NUMBER    := TO_NUMBER(FND_
PROFILE.VALUE('LOGIN_ID' ));
    myOrgID    CONSTANT NUMBER(15) := TO_NUMBER(FND_
PROFILE.VALUE('ORG_ID' ));
    myLang     CONSTANT VARCHAR2(4) := USERENV('LANG');
    myUserName  VARCHAR2(100);

```

CURSOR GetRolePlan IS

```

    select /*+ USE_NL(crpa comp csqaa) INDEX(crpa) INDEX(comp) INDEX(csqaa)
INDEX(jrr) INDEX(compname) +*/
    distinct compname.name compname, jrr.role_name, crpa.start_date, crpa.end_date

```

```

from jtf_rs_roles_tl jrr
    ,cn_comp_plans compname
    ,cn_role_plans crpa
where compname.comp_plan_id = crpa.comp_plan_id
    and jrr.role_id = crpa.role_id
    and jrr.language = USERENV('LANG');
    and compname.name = 'Draw - Payroll Dept. Plan';

```

```

l_commit      NUMBER := 0;
l_start_date  DATE;
l_end_date    DATE;
l_rec         CN_ROLE_PLANS_PUB.role_plan_rec_type;
l_x_return_status  varchar2(10) := 'F';
l_x_msg_count      NUMBER;
l_x_msg_data       VARCHAR2(32767);
l_loading_status   VARCHAR2(100);
l_msg_index_out    VARCHAR2(2000);
l_role_name        VARCHAR2(60);
l_comp_name        VARCHAR2(60);
x_error_mesg       VARCHAR2(2000);
x_error_code       VARCHAR2(200);
v_file            utl_file.file_type;
v_file_name        varchar2(100) := 'AssignDeassignRolePlan' || to_char(sysdate,
'YYYYMMDDHHMISSSS');
v_dir              varchar2(100) := '/OracleApp/oiccomn/temp';
v_buff             constant INTEGER := 10000;
l_deleted          number := 0;

```

```

l_err_deleted    number := 0;
l_created        number := 0;
l_err_created    number := 0;

BEGIN

select value into v_dir from v$parameter where name = 'utl_file_dir';
v_file := UTL_FILE.FOPEN(v_dir, v_file_name, 'W', v_buff);

/** Setting Org Context */
fnd_client_info.set_org_context(myOrgID);

FOR GetRolePlanRec IN GetRolePlan LOOP

    utl_file.put_line(v_file, '-----');

    l_role_name := GetRolePlanRec.role_name;
    l_comp_name := GetRolePlanrec.compname;
    l_start_date := GetRolePlanRec.start_date;
    l_end_date  := GetRolePlanRec.end_date;

    /** Break the Relationship */
    SAVEPOINT REVERTBACK;
    l_rec.ROLE_NAME    := l_role_name;
    l_rec.COMP_PLAN_NAME := l_comp_name;
    l_rec.START_DATE   := l_start_date;
    l_rec.END_DATE     := l_end_date;

```

```

utl_file.put_line(v_file, 'Role = ' || l_role_name);
utl_file.put_line(v_file, 'Plan = ' || l_comp_name);
utl_file.put_line(v_file, 'Start Date : End Date = ' || l_start_date || ':' || l_end_
date);

```

```

CN_ROLE_PLANS_PUB.DELETE_ROLE_PLAN(
P_API_VERSION      => 1.0
, X_RETURN_STATUS   => l_x_return_status
, X_LOADING_STATUS  => l_loading_status
, X_MSG_COUNT       => l_x_msg_count
, X_MSG_DATA        => l_x_msg_data
, P_ROLE_PLAN_REC   => l_rec);

```

```

IF l_x_return_status <> 'S' THEN

```

```

  IF (FND_MSG_PUB.Count_Msg <> 0 ) THEN

```

```

    FOR I IN 1..FND_MSG_PUB.Count_Msg LOOP

```

```

      fnd_msg_pub.get(p_msg_index => I
        ,p_encoded => FND_API.g_false
        ,p_data => l_x_msg_data
        ,p_msg_index_out => l_msg_index_out);

```

```

      x_error_mesg := substr(x_error_mesg || ' ' || l_msg_index_out, 1, 3999);

```

```

END LOOP;

END IF;

x_error_code := 'DELETE: DELETE_ROLE_PLAN';
x_error_Mesg := substr(l_x_msg_data, 1, 500);
utl_file.put_line(v_file, 'Error = ' || x_error_code || ':' || x_error_Mesg);
l_err_deleted := l_err_deleted + 1;
ROLLBACK TO REVERTBACK;

ELSE -- return status = S

utl_file.put_line(v_file, 'Relationship Deleted');
l_deleted := l_deleted + 1;

/** initializing params **/
l_x_return_status := 'F';
l_loading_status := null;
l_x_msg_count := 0;
l_x_msg_data := null;
l_msg_index_out := null;
x_error_code := null;
x_error_Mesg := null;

/** Creating the Relationship Again **/

CN_ROLE_PLANS_PUB.CREATE_ROLE_PLAN(
P_API_VERSION      => 1.0

```

```
, X_RETURN_STATUS      => l_x_return_status
, X_LOADING_STATUS      => l_loading_status
, X_MSG_COUNT           => l_x_msg_count
, X_MSG_DATA            => l_x_msg_data
, P_ROLE_PLAN_REC       => l_rec);
```

```
IF l_x_return_status <> 'S' THEN
```

```
IF (FND_MSG_PUB.Count_Msg <> 0 ) THEN
```

```
FOR I IN 1..FND_MSG_PUB.Count_Msg LOOP
```

```
    fnd_msg_pub.get(p_msg_index => I
        ,p_encoded => FND_API.g_false
        ,p_data => l_x_msg_data
        ,p_msg_index_out => l_msg_index_out);
```

```
        x_error_mesg := substr(x_error_mesg || ' ' || l_msg_index_out, 1,
3999);
```

```
END LOOP;
```

```
END IF;
```

```
x_error_code := 'CREATE: CREATE_ROLE_PLAN';
```

```
x_error_Mesg := substr(l_x_msg_data, 1, 500);
```

```
utl_file.put_line(v_file, 'Error = ' || x_error_code || ':' || x_error_Mesg);
```

```
        l_err_created := l_err_created + 1;
        ROLLBACK TO REVERTBACK;

    ELSE -- return status = S

        utl_file.put_line(v_file, 'Relationship Re-Created');
        l_created := l_created + 1;
        COMMIT;

    END IF; -- status from create

END IF; -- status from delete

END LOOP; -- Main Driver

utl_file.put_line(v_file, 'Total Number of Records Deleted = ' || l_deleted);
utl_file.put_line(v_file, 'Total Number of Deletion Errors = ' || l_err_deleted);
dbms_output.put_line('Total Number of Records Deleted = ' || l_deleted);
dbms_output.put_line('Total Number of Deletion Errors = ' || l_err_deleted);
utl_file.put_line(v_file, 'Total Number of Records Created = ' || l_created);
utl_file.put_line(v_file, 'Total Number of Creation Errors = ' || l_err_created);
dbms_output.put_line('Total Number of Records Created = ' || l_created);
dbms_output.put_line('Total Number of Creation Errors = ' || l_err_created);
utl_file.fclose(v_file);

EXCEPTION
when UTL_FILE.INVALID_PATH then
    raise_application_error(-20101, 'Invalid Path');
```

```
when UTL_FILE.INVALID_MODE then
    raise_application_error(-20102, 'Invalid Mode');
when UTL_FILE.INVALID_MAXLINESIZE then
    raise_application_error(-20103, 'Invalid Max Line Size');
when UTL_FILE.INVALID_FILEHANDLE then
    raise_application_error(-20104, 'Invalid File Handle');
when UTL_FILE.INVALID_OPERATION then
    raise_application_error(-20105, 'Invalid Operation');
when UTL_FILE.WRITE_ERROR then
    raise_application_error(-20106, 'Write Error');
when others then
    raise_application_error(-20107, 'When Others: ' || sqlcode || ':' || sqlerrm);
END;
/
```

