

# **Oracle® Trading Community Architecture**

API User Notes

Release 11*i*

**Part No. B10106-02**

March 2003

Oracle Trading Community Architecture API User Notes, Release 11i

Part No. B10106-02

Copyright © 2002, 2003 Oracle Corporation. All rights reserved.

Primary Author: Charlie Ahern

Contributors: Joe del Callar, Jianying Huang, Shannon Kreps, Sai Rangarajan, Leena Sampemane, Indrajit Sen, Kate Shan, Dylan Wan, and Herve Yu

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

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

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

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

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

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, and Oracle Trading Community Architecture, Oracle8i, Oracle9i, Oracle*MetaLink*, PL/SQL, Pro\*C, SQL\*Net, and SQL\*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>vii</b>
<b>Preface.....</b>	<b>ix</b>
<b>1 Overview</b>	
Introduction .....	1-2
Business Needs Met .....	1-3
<b>2 Features, Parameters, Attributes and Messages</b>	
Major Features and Solution Outline .....	2-2
Standard Parameters and Missing Attributes (PL/SQL API).....	2-5
Debug Messages (PL/SQL API) .....	2-7
Standard Parameters and Missing Attributes (Java API) .....	2-9
Debug Messages (Java API).....	2-13
Security of Oracle Human Resources Data.....	2-15
Third Party Data Integration .....	2-16
<b>3 Party API Use</b>	
Party APIs.....	3-2
<b>4 Party Contact API Use</b>	
Party Contact APIs.....	4-2

<b>5</b>	<b>Person Info, Location and Party Site API Use</b>	
	Person Info APIs .....	5-2
	Location APIs .....	5-7
	Party Site APIs .....	5-19
<b>6</b>	<b>Contact Point API Use</b>	
	Contact Point APIs .....	6-2
<b>7</b>	<b>Relationship Type and Relationship API Use</b>	
	Relationship Type APIs .....	7-2
	Relationship APIs .....	7-13
<b>8</b>	<b>Classification API Use</b>	
	Classification APIs .....	8-2
<b>9</b>	<b>Contact Preference API Use</b>	
	Contact Preference APIs .....	9-2
<b>10</b>	<b>Customer Account API Use</b>	
	Customer Account APIs .....	10-2
<b>11</b>	<b>Customer Account Site API Use</b>	
	Customer Account Site APIs .....	11-2
<b>12</b>	<b>Customer Profile and Customer Account Role API Use</b>	
	Customer Profile APIs .....	12-2
	Customer Account Role APIs .....	12-40
<b>13</b>	<b>Miscellaneous API Use</b>	
	Tax Assignment APIs .....	13-2
	Phone Parsing and Formatting APIs .....	13-6
	Name and Address Formatting APIs .....	13-9

General Data Formatting API..... 13-19  
Hierarchy Retrieval APIs..... 13-20

**A Sample Code**

Sample Code - Demonstrating the APIs ..... A-2  
Difference Between Version 1.0 and Version 2.0 Public API for Update Procedures ..... A-40

**B List of Messages**

Messages for the TCA APIs ..... B-2

**C API Mapping**

Mapping Version 1 to Version 2 APIs ..... C-2



---

---

# Send Us Your Comments

## **Oracle Trading Community Architecture API User Notes, Release 11i**

### **Part No. B10106-02**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information 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 about this manual?

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](mailto:appsdoc_us@oracle.com)
- FAX: (801) 659-7164 Attn: Oracle Applications Documentation Manager
- Postal service:  
Oracle Applications Documentation Manger  
Oracle Corporation  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

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

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



---

---

# Preface

Welcome to Release 11*i* of the *Oracle Trading Community Architecture API User Notes*.

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

- The principles and customary practices of your business area.
- If you have never used Oracle Trading Community Architecture Application Programming Interfaces, Oracle suggests you attend one or more of the Oracle Applications training classes available through Oracle University.
- 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 Trading Community Architecture API User Notes contains the information you need to understand and use Oracle Trading Community Architecture Application Programming Interfaces. These user notes includes:

- Chapter 1 provides a brief overview of the features and benefits of the TCA application programming interfaces (APIs).
- Chapter 2 describes the features, parameters, attributes and messages that are part of the TCA APIs.
- Chapters 3 through 13 describe how to create and update records with each of the TCA APIs. Some of the information provided for an API includes a description, the PL/SQL procedure, the Java method, parameter descriptions, and information about any defaults and validations for a parameter.
- Appendix A provides sample code that uses TCA APIs.
- Appendix B provides the number, code, and text of the TCA API messages.
- Appendix C provides information about mapping the first, internal-only version of the APIs to the second, public version of the APIs.

## **Documentation Accessibility**

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

### **Accessibility of Code Examples in Documentation**

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

### **Accessibility of Links to External Web Sites in Documentation**

This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle Trading Community Architecture Application Programming Interfaces.

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).

- **Online Help** - The new features section in the HTML help describes new features in 11*i*. This information is updated for each new release of Oracle Trading Community Architecture Application Programming Interfaces. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-packs an upgrade, this document describes the new features. Online help patches are available on MetaLink.
- **11*i* Features Matrix** - This document lists new features available by patch and identifies any associated new documentation. The new features matrix document is available on MetaLink.
- **Readme File** - Refer to the readme file for patches that you have installed to learn about new documentation or documentation patches that you can download.

### Related Guides

Oracle Trading Community Architecture Application Programming Interfaces 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 Trading Community Architecture Application Programming Interfaces.

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) available with this release of Oracle Trading Community Architecture Application Programming Interfaces (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

## **Guides Related to This Product**

### **Oracle Trading Community Architecture Administration User Guide**

Use this user guide to learn how to set up relationship types that can be used to create and manage relationships among entities in the TCA Registry. You can create new relationship types, add relationship phrase and role pairs to existing types, and assign phrase and role pairs to relationship groups. All types and role pairs that are created in Administration can be used in Relationship Manager to create or edit relationships with. The user guide also provides information about administration for other features.

### **Oracle Trading Community Architecture Relationship Manager User Guide**

Use this user guide to learn how to manage relationships among existing parties in the TCA Registry. You can view, create, and edit relationships, as well as view hierarchical relationships in a structural hierarchy.

### **Oracle Trading Community Architecture Data Quality Management User Guide**

Use this user guide to learn how to identify and merge duplicate parties in the Oracle Trading Community Architecture registry. The Oracle Trading Community Architecture Data Quality Management User Guide describes how to set up and use transformation functions and match rules to identify possible duplicate parties.

### **Oracle Trading Community Architecture Party Merge User Guide**

Use this user guide to learn how to merge parties and their related entities in the Oracle Trading Community Architecture registry. The Oracle Trading Community Architecture Party Merge User Guide describes how to set up and process party merge batches as well as how to identify merge errors.

## **Oracle Trading Community Architecture Third Party Data Integration User Guide**

Use this user guide to learn how to manage and acquire third party information in the TCA Registry. The user guide describes acquiring third party data from D&B.

## **Oracle Receivables User Guide**

Use this user guide to learn how to implement flexible address formats for different countries. You can use flexible address formats in the suppliers, customers, banks, invoices, and payments windows in both Oracle Payables and Oracle Receivables. This user guide also explains how to set up your system, create transactions, and run reports in Oracle Receivables.

## **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, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user 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.

## **Maintaining Oracle Applications Documentation Set**

This documentation set includes *Oracle Applications AD Procedures Guide* and *Oracle Applications AD Utilities Reference Guide*.

Use these guides to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. They contain how-to steps, screenshots, and other information that you need to run the AD utilities. These guides also provide information on maintaining the Oracle applications file system and database.

## **Oracle Applications System Administrator's Guide**

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

## **Oracle Alert User's Guide**

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

## **Oracle Applications Developer's Guide**

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer 6*i* forms so that they 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 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

### **Oracle Workflow Guide**

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

### **Oracle Applications Flexfields Guide**

This guide provides flexfields planning, setup and reference information for the Oracle Trading Community Architecture Application Programming Interfaces 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 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 Trading Community Architecture Application Programming Interfaces 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 Trading Community Architecture Application Programming Interfaces working for you. This team includes your technical representative, account manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8i server, and your hardware and software environment.

## Do Not Use Database Tools to Modify Oracle Applications Data

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

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

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. 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 Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

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

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

## Your Feedback

Thank you for using Oracle Trading Community Architecture Application Programming Interfaces and these user notes.

Oracle values your comments and feedback. In this guide is a reader's comment form that you can use to explain what you like or dislike about Oracle Trading Community Architecture Application Programming Interfaces or this user guide. Mail your comments to the following address or call us directly at (650) 506-7000.

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

Or, send electronic mail to [appsdoc\\_us@oracle.com](mailto:appsdoc_us@oracle.com).



# 1

---

---

## Overview

This chapter introduces the features of and the business needs met by the Oracle Trading Community Architecture application programming interfaces (API).

## Introduction

This document provides the information that you need to access the Trading Community Architecture (TCA) data model, which is the foundation for applications in Oracle's Release 11i E-Business Suite. Customers, consultants, and Oracle internal development teams can use the public TCA application programming interfaces (APIs).

Important features of the TCA API:

- Flexible, easy to understand, and modular.
- Extensive debugging capability.
- Extensive error handling and reporting capability.
- Robust validation in all of the APIs.
- A new locking mechanism based on the OBJECT\_VERSION\_NUMBER field, which has been included in all of the HZ entities for which the public APIs have been provided.
- Standard signature and availability of common parameters.
- The following main categories of entities are covered:
  - Parties-person, organization, group
  - Locations
  - Party sites, party site use
  - Organization contact, organization contact role
  - Contact points
  - Contact preferences
  - Relationship types
  - Relationships
  - Classification
  - Customer accounts
  - Account sites and site uses
  - Customer account role, role responsibility
  - Customer profile, customer profile amount

## Business Needs Met

Programmatic access to the TCA Data model meets the following business needs:

- Applications in the Oracle E-Business Suite can use the TCA public APIs to insert and update entities in the TCA model, as part of server side and middle tier business logic.
- APIs provide a gateway to the TCA data model from applications that use Forms 6.0 user interfaces (UIs) as well as from HTML UIs.
- Data from providers such as Dun & Bradstreet, Experian, and others can be mass loaded into the TCA model by using the PL/SQL API. The Dun & Bradstreet integration currently available with Oracle Receivables also utilizes the TCA APIs.
- Data migration from legacy systems into the TCA model.
- Access to the TCA model from custom applications built by customers and Oracle Consulting.



---

---

## Features, Parameters, Attributes and Messages

This chapter describes the major features and use of the Oracle Trading Community Architecture application programming interfaces (API).

## Major Features and Solution Outline

### Setting Applications Context

You must set the Applications Context in a database session, before calling any Trading Community Architecture application programming interfaces (API). The APIs rely on global variables and profiles that are part of this Applications Context. For details on setting Applications Context, please refer to Note 209185.1 on Oracle Metalink (<http://metalink.oracle.com>).

### Modular Approach

The TCA API has been designed in a highly modular fashion, giving you code that is easy to understand, easy to maintain and easy to extend.

The modular approach defaults and validates user-entered information, defaults information not provided by the user, and calls the appropriate entity handler to perform the business related tasks.

### Exception Handling and Results Messages

The APIs provide an extensive set of error-handling and error-reporting mechanisms so that errors encountered in the different phases of API execution are reported and put on the message stack. The calling program has the option of looking up all the error messages or the first error message on the stack. If there is only one error in the message stack, the error is displayed as one of the output parameters of the API routine. You do not have to fetch that message from the stack.

There are three types of information that the TCA APIs provide to their calling programs:

- Overall status
- Messages describing the operations performed or errors encountered by the API
- Output values that the program making the call to the API might need to use

### Return Status

The return status (`x_return_status`) of the API informs the caller about the result of the operation or operations performed by the API. The possible return status values and their meanings are:

**Success**

`FND_API.G_RET_STS_SUCCESS` (PL/SQL API)

`HzConstant.getGRetStsSuccess()` (Java API)

A success return status indicates that the API performed all of the operations requested by its caller. A success return status can be accompanied by informative messages in the API message list.

**Error**

`FND_API.G_RET_STS_ERROR` (PL/SQL API)

`HzConstant.getGRetStsError()` (Java API)

An error return status indicates that the API failed to perform some or all of the operations requested by its caller. An error return status is usually accompanied by messages describing any errors and how to resolve them.

In most cases, you should be able to correct normal, expected errors such as missing attributes or invalid date ranges.

**Unexpected Error**

`FND_API.G_RET_STS_UNEXP_ERROR` (PL/SQL API)

`HzConstant.getGRetStsUnexpError()` (Java API)

An unexpected error status indicates that the API encountered an unexpected error condition that it could not handle. In this case, the API cannot continue its regular processing. Examples of such errors are irrecoverable data inconsistency errors, memory errors, and programming errors such as attempting to divide by zero.

In most cases, an end user will not be able to correct unexpected errors. These errors usually require resolution by a system administrator or an application developer.

**Messages**

The APIs put result messages into a message list. Programs calling these APIs can then get the messages from the list and process those messages by issuing them, loading them into a database table, or writing them to a log file.

The APIs store messages in an encoded format so that the API callers can use standard functions provided by the message dictionary to find message names. With the message dictionary you can also store these messages in database tables and generate reports from these tables in different languages. For more information, see List of Messages on page B-2

The API message list must be initialized every time a program calls an API. API callers can either call the message list utility function, `FND_MSG_PUB.Initialize`, or request that the API do the initialization by setting the `p_init_msg_list` parameter to `T` (TRUE).

The program calling the API can retrieve messages from the message stack using the existing FND API functions `FND_MSG_PUB.Count_Msg` (or standard output parameter `x_msg_count`) and `FND_MSG_PUB.Get`. You can see examples of these in the sample code in the appendix.

### Robust Validation

The TCA APIs perform robust validations. The APIs collect all the validation errors encountered and put them on the message stack. The relevant entity handler is called only if no errors are reported during the validation phases.

### Locking Mechanism

The TCA public APIs provide a new locking mechanism for update procedures, based on the new `OBJECT_VERSION_NUMBER` column, which has been included in all HZ tables. For this reason, `OBJECT_VERSION_NUMBER` is a mandatory attribute for all update APIs.

The locking mechanism works as follows:

- Whenever a new record is created, the value in the `OBJECT_VERSION_NUMBER` column is set to 1.
- Whenever a record is updated, the value in the `OBJECT_VERSION_NUMBER` column is reset to `OBJECT_VERSION_NUMBER + 1`.
- For records that existed in the HZ tables prior to introduction of this locking mechanism, the API sets the value in the column `OBJECT_VERSION_NUMBER` to null.

## Standard Parameters and Missing Attributes (PL/SQL API)

The TCA APIs have been developed to conform to the Oracle Applications API standards. The parameters below are common to all of the APIs. Brief descriptions are provided for some of the important features of the API.

### Standard IN or IN/OUT Parameters

- `p_<entity>_object_version_number` IN/OUT NUMBER Required

This parameter is either called `p_object_version_number` or `p_<entity>_object_version_number`. For example, for the `hz_party_v2pub.update_organization` API, this parameter is called `p_party_object_version_number` because organization is a type of party. The parameter value must match the version number in the database of the record being updated. An error will be returned if the calling program passes an object version number that is not identical to the one in the database for the existing record.

- `p_init_msg_list` IN VARCHAR2 Optional

The default is `FND_API.G_FALSE`. If set to true, the API calls `fnd_msg_pub.initialize` to initialize the message stack. If it set to false, then the calling program must initialize the message stack. The initialization must only be done once in the case where more than one API is being called.

---



---

**Warning:** Do not use the `rec` parameter for the OUT parameter. For example, do not use "`p_organization_rec.party_rec.party_id`", instead use "`x_party_id`".

---



---

### Standard OUT Parameters

The names of all output parameters begin with "x\_"

- `x_return_status` OUT VARCHAR2(1)

The Out parameter returns the status of the API. The returned value is one of the following :

- `FND_API.G_RET_STS_SUCCESS` - Success
- `FND_API.G_RET_STS_ERROR` - Expected error, validation or missing data
- `FND_API.G_RET_STS_UNEXP_ERROR` - Unexpected error, cannot be corrected by the calling program

- `x_msg_count` OUT NUMBER
- `x_msg_data` OUT VARCHAR2

The `x_msg_count` column contains the number of messages in the message list. If the count is one, then the `x_msg_data` column holds the encoded message.

## Nested Record Types

PL/SQL record types are used in all of the create and update APIs. In some cases, nested record types have been used as well.

For example, in the `Create_Person` API, the `p_person_rec` input parameter is of the `person_rec_type` record type. The `person_rec_type` has `party_rec` as one of its elements, which itself is of `party_rec_type` record type.

The reason for doing this is that the attributes of the `Person` party type include attributes of a party, because `Person` is a subtype of `Party`.

## Primary Key Generation

The `Create` APIs handle both sequence-generated and manually-passed primary keys, which can be generated on a device from an Oracle Mobile application. In order to prevent duplicate primary keys, the `Create` APIs handle exceptions to the unique key violation.

## Missing Attributes

Optional IN parameters do not have default values. An attribute value that is not passed in is assumed to have a default of null for a `Create` API. For an `Update` API, if a particular attribute value is not passed into the API, then the database retains the existing value. In order to set a database value to null, the calling program must explicitly set the attribute value to one of the following constants, based on the data type of the attribute.

- `FND_API.G_MISS_NUM` for NUMBER type.
- `FND_API.G_MISS_CHAR` for VARCHAR2 type.
- `FND_API.G_MISS_DATE` for DATE type.

These are pre-defined values in the `FND_API` Package (`fndapis.pls`)

- `G_MISS_NUM` CONSTANT NUMBER:= 9.99E125
- `G_MISS_CHAR` CONSTANT VARCHAR2(1):= chr(0)
- `G_MISS_DATE` CONSTANT DATE:= TO\_DATE('1','j');

## Debug Messages (PL/SQL API)

Use the extensive debug messages to trouble shoot in case of unexpected problems. These debugging messages are extremely useful because an API would be difficult to debug otherwise. You can turn on debug messages by the use of a certain profile option. These messages can be written to a log file as well.

The profiles for controlling the debug mechanism are:

- Name - HZ\_API\_FILE\_DEBUG\_ON  
User Profile Name - HZ: Turn On File Debug
- Name - HZ\_API\_DEBUG\_FILE\_NAME  
User Profile Name - HZ: API Debug File Name
- Name - HZ\_API\_DEBUG\_FILE\_PATH  
User Profile Name - HZ: API Debug File Directory

If the HZ\_API\_FILE\_DEBUG\_ON profile is set to Y when any APIs are called, then debug messages are written to the file specified in HZ\_API\_DEBUG\_FILE\_PATH and HZ\_API\_DEBUG\_FILE\_NAME. If the HZ\_API\_FILE\_DEBUG\_ON profile is set to N, no debug messages are generated.

The value of the HZ\_API\_DEBUG\_FILE\_PATH profile specifies a directory file path that the database has write access to, as provided in init.ora. You can find path information by querying: `select value from v$parameter where name equals 'utl_file_dir'`. If you turn the file debug mode on, but did not set a proper value for the HZ\_API\_DEBUG\_FILE\_PATH profile or the HZ\_API\_DEBUG\_FILE\_NAME profile is null, the API errors out.

Debug messages accumulate in the debug file. After collecting any debug messages, you must reset the HZ\_API\_FILE\_DEBUG\_ON profile back to N. If you do not, you might cause an *exceeded file size* error.

There are two ways to run APIs in the debug mode:

- From Oracle Applications:  
You can enable or disable the debug mode by setting HZ\_API\_FILE\_DEBUG\_ON to Y or N, respectively. The default value is N. When the profile is set Y, you must set the proper values for the HZ\_API\_DEBUG\_FILE\_NAME and HZ\_API\_DEBUG\_FILE\_PATH profiles.
- With SQLPLUS or server side PL/SQL custom code:

You can enable or disable the debug mode by calling FND\_PROFILE API.

This example assumes that the directory, /sqlcom/out/tca115, has write access that is specified by the utl\_file\_dir parameter in the init.ora for the relevant database.

### From SQLPLUS

```
exec fnd_profile.put('HZ_API_DEBUG_FILE_PATH', '/sqlcom/out/tca115/');  
exec fnd_profile.put('HZ_API_DEBUG_FILE_NAME', 'api_debug');  
exec fnd_profile.put('HZ_API_FILE_DEBUG_ON', 'Y');
```

### From PL/SQL code

```
fnd_profile.put('HZ_API_DEBUG_FILE_PATH', '/sqlcom/out/tca115/');  
fnd_profile.put('HZ_API_DEBUG_FILE_NAME', 'api_debug');  
fnd_profile.put('HZ_API_FILE_DEBUG_ON', 'Y');
```

This debug strategy is provided as a public utility procedure that you can include in your custom code.

Please refer to the HZ\_UTILITY\_V2PUB package for further details.

## Standard Parameters and Missing Attributes (Java API)

The TCA APIs have been developed to conform to the Oracle Applications API standards. The parameters below are common to all of the APIs. Brief descriptions are provided of some of the important features of the API.

Comments that are the same for PL/SQL API are repeated in this section.

Declare all parameters or member variables as `BigDecimal` if it is PL/SQL `NUMBER` type, `String` if it is PL/SQL `VARCHAR2` type, or `Timestamp` if it is PL/SQL `DATE` type. To access these parameters or variables you must import

- `java.math.BigDecimal`
- `java.sql.Timestamp`

### IN OUT/OUT Parameters

In PL/SQL the caller's value can be modified by the called procedure, if a parameter is declared `OUT` or `IN OUT`. For example, PL/SQL routines calling this procedure:

```
procedure p1(n1 in out number, n2 number);
```

might find that the first variable passed to `p1` was modified by the time the program's call to `p1` is completed, because the parameter is declared `IN OUT`. The value the programs pass to `n2` cannot possibly be modified.

A PL/SQL procedure can declare any parameter to be `OUT` or `IN OUT`. The caller must be prepared in case the PL/SQL procedure has modified any data, and that it is therefore part of the value returned from the procedure.

Java has no clear analogue to the concept of declaring parameters as `OUT` or `IN OUT`. Instead, in Java, there are certain types of objects that are immutable (changes by the called procedure which cannot be seen by the caller) and other types which are mutable.

For this reason, if there are any arguments to PL/SQL which are `OUT` or `IN OUT`, the Java API for it must be of a mutable type. This is why arrays are mutable APIs. For example, the Java API for procedure `p1` above might look like:

```
public static void p1 (BigDecimal [ ] n1, BigDecimal n2) {...}
```

The first parameter is an array and the second is not, even though, in PL/SQL, both parameters are simply `NUMBER`. In the case where `p1` modifies `n1`, you can identify the new value. You can get the value by referencing, for instance, `n1[0]`.

## Standard IN or IN/OUT Parameters

- OracleConnection \_connection Required  
This parameter is for passing Oracle JDBC connection to APIs.
- BigDecimal [ ] p\_object\_version\_number Required  
This parameter is either called p\_object\_version\_number or p\_<entity>\_object\_version\_number. For example, for the HzPartyV2Pub.updateOrganization API, this parameter is called p\_party\_object\_version\_number because organization is a type of party. The value of p\_object\_version\_number[0] must match the version number in the database of the record being updated. An error will be returned if the calling program passes an object version number that is not identical to the one in the database for the existing record.

String p\_init\_msg\_list Optional

If HzConstant.getGTrue() is set to true, the API makes a call to fnd\_msg\_pub.initialize to initialize the message stack. If HzConstant.getGFalse() is set to false, then the calling program must initialize the message stack. This initialization is required only once when more than one API is called.

## Standard OUT Parameters

The names of all output parameters begin with "x\_"

- String [ ] x\_return\_status  
This returns the status of the API. The values returned in x\_return\_status[0] are one of the following:  
HzConstant.getGRetStsSuccess() - Success  
HzConstant.getGRetStsError() - Expected Error - validation or missing data.  
HzConstant.getGRetStsUnexpError() - Unexpected Error, not fixable by calling program.
- BigDecimal [ ] x\_msg\_count
- String [ ] x\_msg\_data  
x\_msg\_count[0] holds the number of messages in the message list. If the count is one, then x\_msg\_data[0] holds the decoded message.

## Nested Record Types

PL/SQL record types are used in all Create and Update APIs. In some cases nested record types are used as well. Correspondingly, the Java Inner Class (for example, Record Class) is defined for each PL/SQL record type.

For example, in the `HzPartyV2Pub.createPerson()` method, the `p_person_rec` input parameter is of type `HzPartyV2Pub.PersonRec`. The `HzPartyV2Pub.PersonRec` parameter has `party_rec` as one of its elements which itself is of type `HzPartyV2Pub.PartyRec`.

The reason for doing this is that the attributes of the Person party type include attributes of a party, because Person is a subtype of Party.

## Primary Key Generation

The Create APIs handle both sequence-generated and manually-passed primary keys, which can be generated on a device from an Oracle Mobile application. In order to prevent duplicate primary keys, the Create APIs handle exceptions to the unique key violation.

## Missing Attributes

Each record class has two constructors:

- `Public <entity>Rec();`

This constructor initializes all `BigDecimal`, `String` and `Timestamp` attributes to `HzConstant.getGMissNum()`, `HzConstant.getGMissChar()`, `HzConstant.getGMissDate()`. If the record class is nested, then the included record classes will also be initialized as `GMissXXX`.

- `HzConstant.getGMissNum()` for `BigDecimal` type.

The application returns `9.99E125`.

- `HzConstant.getGMissChar()` for `String` type.

The application returns `"\0"`.

- `HzConstant.getGMissDate()` for `java.sql.Timestamp` type.

The application returns `"4713-01-01 00:00:00.0"`

- `Public <entity>Rec(boolean __RosettaUseGMISSValues);`

This constructor provides two choices. Calling this constructor with a 'true' value corresponds to the first constructor. If we call the constructor by passing

'false', all BigDecimal, String and Timestamp attributes will be initialized to null values. Similarly, this boolean parameter will be passed into included record classes if any is present.

There is a performance difference between passing 'true' or 'false'. If passing 'true', we only transmit to the database the GMiss values for attributes that a user explicitly sets in creating and updating. In client or middle tier, more memory is needed to hold GMiss values, but it causes less network traffic. If passing 'false', we internally convert null value to GMiss in order to call the public APIs and transmit all attributes to the database. This approach can result in less memory consumption in the client, but more network traffic between the middle tier and the sever. You should consider the physical configuration of your application structure as well as how many attributes you display in the UI when you choose a constructor. If you have less than 20% of the attributes exposed in UI, for instance, the first approach, passing 'true' or using default constructor, is more efficient.

Unlike the PL/SQL APIs, Java APIs take whatever values you pass in to create or update records.

## Debug Messages (Java API)

Use the extensive debug messages to trouble shoot in case of unexpected problems. These debugging messages are extremely useful because an API would be difficult to debug otherwise. You can turn on debug messages by the use of a certain profile option. These messages can be written to a log file as well.

The profiles for controlling the debug mechanism are:

- Name - HZ\_API\_FILE\_DEBUG\_ON  
User Profile Name - HZ:Turn On File Debug
- Name - HZ\_API\_DEBUG\_FILE\_NAME  
User Profile Name - HZ: API Debug File Name
- Name - HZ\_API\_DEBUG\_FILE\_PATH  
User Profile Name - HZ: API Debug File Directory

If the HZ\_API\_FILE\_DEBUG\_ON profile is set to Y when any APIs are called, then debug messages are written to the file specified in HZ\_API\_DEBUG\_FILE\_PATH and HZ\_API\_DEBUG\_FILE\_NAME. If the HZ\_API\_FILE\_DEBUG\_ON profile is set to N, no debug messages are generated.

The value of the HZ\_API\_DEBUG\_FILE\_PATH profile specifies a directory file path that the database has write access to, as provided in init.ora. You can find this path information by querying: select value from v\$parameter where the name equals 'utl\_file\_dir'.

If you turn the file debug mode on, but did not set a proper value for the HZ\_API\_DEBUG\_FILE\_PATH profile or the HZ\_API\_DEBUG\_FILE\_NAME profile is null, the API errors out.

Debug messages accumulate in the debug file. After collecting any debug messages, you must reset the HZ\_API\_FILE\_DEBUG\_ON profile back to N. If you do not you might cause an *exceeded file size* error.

You can call FND\_PROFILE.Put to set profile value.

This example assumes that the directory, /sqlcom/out/tca115, has write access, that is specified by the utl\_file\_dir parameter in the init.ora for the relevant database.

```
OracleCallableStatement ocs = (OracleCallableStatement)conn.prepareCall(
"begin fnd_profile.put('HZ_API_DEBUG_FILE_PATH', '/sqlcom/out/tca115/'); end;");
ocs.execute();
```

```
ocs = (OracleCallableStatement)conn.prepareCall(
"begin fnd_profile.put('HZ_API_DEBUG_FILE_NAME', 'api_debug'); end;");
ocs.execute();
ocs = (OracleCallableStatement)conn.prepareCall(
"begin fnd_profile.put(fnd_profile.put('HZ_API_FILE_DEBUG_ON', 'Y'); end;");
ocs.execute();
```

## Security of Oracle Human Resources Data

You can use the TCA APIs to access data about people after the data is entered and stored using Oracle Human Resources (HR) applications. The security, integrity, and validity of HR data must be maintained due to the sensitivity of personal information.

TCA's Security of HR Person Data feature enables Oracle applications to share data entered and stored with Oracle HR applications. With this feature other Oracle applications can access information about a person without compromising the security, integrity, or validity of the HR information. Although other Oracle applications can access information in the HR tables, only properly authorized users of HR applications can modify data about any person entered by using an Oracle HR application.

## Third Party Data Integration

The following PL/SQL and Java APIs support third-party data integration:

- Party API
  - Create Organization API
  - Update Organization API
  - Create Person API
  - Update Person API
- Location API
- Party Site API
- Contact Point API
- Relationship API

### General Overview

The integration of third-party data, from providers such as D&B, enables users to acquire data from external sources and then use that data to populate the HZ tables. The data in these HZ tables can then be viewed and used across the Oracle e-Business suite.

From an API, you can populate the `actual_content_source` column to identify the third party data source. The `content_source_type` attribute is obsolete. You should use the `actual_content_source` attribute for third-party data integration applications. If necessary, you can use the `content_source_type` attribute to maintain backward compatibility.

Default values for the `actual_content_source` and `content_source_type` attributes are taken from the `actual_content_source` column, unless you have overwritten the default value in either column.

### Party API

In the Organization and Person record types; the values in the `actual_content_source` column default to SST and the values in the `content_source_type` column default to `user_entered`.

## Create Organization API and Create Person API

These APIs first determine the data source based on the values in the `actual_content_source` and `content_source_type` columns. If the data source is either SST or `user_entered`, the API errors out if the same party ID exists in the `HZ_PARTIES` table. Otherwise, the API will create new party profiles and parties.

If the data source is a third-party data source and if we create profiles for existing parties by passing a party ID, the API errors out if the parties have profiles from the same data source. Otherwise, the API creates the third-party profiles for the parties. If you set up Advanced Third Party Data Integration and select the third party data source as one of the ranked data sources, the API creates or updates the third-party profiles, SST or single source of truth, and denormalizes the SST profiles to `HZ_PARTIES`. Users can only see SST profiles.

## Update Organization API and Update Person API

These APIs first determine the data source based on the values in the `actual_content_source` and `content_source_type` columns. If the data came from a third-party source, but the user does not have permission to overwrite third party data, then the API triggers an error message; "You do not have permission to overwrite the information in the `&COLUMN` column, which comes from a third party data source. Please contact your system administrator."

## Location API

The values in both the `actual_content_source` and `content_source_type` columns default to `user_entered`. Users cannot update third-party locations.

## Party Site API

The values in both the `actual_content_source` and `content_source_type` columns default to `user_entered`.

You can control whether third-party data for a site is active or inactive. The API sets the identifying address flag to make the first active, visible party site the identifying address for the party. By default you can only view user-entered party sites. However, you can use the Third Party Data Integration Setup user interface to choose which data sources are visible. For more details, please refer to the *Third Party Data Integration User Guide* and to information about the Party Site API in this document.

## Contact Point API

The values in both the `actual_content_source` and `content_source_type` columns default to `user_entered`.

You can control whether third-party data for a contact point site is active or inactive. For example, the value of the `actual_content_source` attribute is other than `user_entered`. For the primary contact point flag, the API marks the first active, visible contact point based on the combination of the `OWNER_TABLE_NAME`, `OWNER_TABLE_ID` and `CONTACT_POINT_TYPE` attributes. By default you can only view user-entered party sites. However, you can use the Third Party Data Integration Setup user interface to choose which data sources are visible. For more details, please refer to the *Third Party Data Integration User Guide* and to information about the Party Site API in this document.

You cannot update phone components of D&B data.

## Relationship API

The values in both the `actual_content_source` and `content_source_type` columns default to `user_entered`.

You can control whether third-party data for a contact point site is active or inactive. For example, the value of the `actual_content_source` attribute is other than `user_entered`.

---

---

## Party API Use

This chapter provides information about the Party application programming interfaces.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Party APIs

**PL/SQL Package Name: HZ\_PARTY\_V2PUB**

**Java Class Name: HzPartyV2Pub**

**PL/SQL Constant :**

**G\_MISS\_CONTENT\_SOURCE\_TYPE**

CONSTANT VARCHAR2(30) := USER\_ENTERED;

**G\_SST\_SOURCE\_TYPE**

CONSTANT VARCHAR2(30) := SST;

### PL/SQL Record Structure for Party

TYPE party_rec_type	IS RECORD(
party_id	NUMBER,
party_number	VARCHAR2(30),
validated_flag	VARCHAR2(1),
orig_system_reference	VARCHAR2(240),
status	VARCHAR2(1),
category_code	VARCHAR2(30),
salutation	VARCHAR2(60),
attribute_category	VARCHAR2(30),
attribute1	VARCHAR2(150),
attribute2	VARCHAR2(150),
attribute3	VARCHAR2(150),
attribute4	VARCHAR2(150),
attribute5	VARCHAR2(150),
attribute6	VARCHAR2(150),
attribute7	VARCHAR2(150),
attribute8	VARCHAR2(150),
attribute9	VARCHAR2(150),
attribute10	VARCHAR2(150),
attribute11	VARCHAR2(150),
attribute12	VARCHAR2(150),
attribute13	VARCHAR2(150),
attribute14	VARCHAR2(150),
attribute15	VARCHAR2(150),
attribute16	VARCHAR2(150),
attribute17	VARCHAR2(150),
attribute18	VARCHAR2(150),
attribute19	VARCHAR2(150),
attribute20	VARCHAR2(150),

```

        attribute21          VARCHAR2(150),
        attribute22          VARCHAR2(150),
        attribute23          VARCHAR2(150),
        attribute24          VARCHAR2(150)
    )

```

## PL/SQL Record Structure for Organization

```

TYPE organization_rec_type IS RECORD(
    organization_name      VARCHAR2(360),
    duns_number_c          VARCHAR2(30),
    enquiry_duns           VARCHAR2(15),
    ceo_name               VARCHAR2(240),
    ceo_title              VARCHAR2(240),
    principal_name         VARCHAR2(240),
    principal_title        VARCHAR2(240),
    legal_status           VARCHAR2(30),
    control_yr             NUMBER,
    employees_total        NUMBER,
    hq_branch_ind          VARCHAR2(30),
    branch_flag            VARCHAR2(1),
    oob_ind                VARCHAR2(30),
    line_of_business       VARCHAR2(240),
    cong_dist_code         VARCHAR2(2),
    sic_code                VARCHAR2(30),
    import_ind             VARCHAR2(30),
    export_ind             VARCHAR2(30),
    labor_surplus_ind      VARCHAR2(30),
    debarment_ind          VARCHAR2(30),
    minority_owned_ind     VARCHAR2(30),
    minority_owned_type    VARCHAR2(30),
    woman_owned_ind        VARCHAR2(30),
    disadv_8a_ind          VARCHAR2(30),
    small_bus_ind          VARCHAR2(30),
    rent_own_ind           VARCHAR2(30),
    debarments_count       NUMBER,
    debarments_date        DATE,
    failure_score           VARCHAR2(30),
    failure_score_natnl_percentile NUMBER,
    failure_score_override_code VARCHAR2(30),
    failure_score_commentary VARCHAR2(30),
    global_failure_score   VARCHAR2(5),
    db_rating               VARCHAR2(5),
    credit_score            VARCHAR2(30),
    credit_score_commentary VARCHAR2(30),

```

paydex_score	VARCHAR2 (3),
paydex_three_months_ago	VARCHAR2 (3),
paydex_norm	VARCHAR2 (3),
best_time_contact_begin	DATE,
best_time_contact_end	DATE,
organization_name_phonetic	VARCHAR2 (320),
tax_reference	VARCHAR2 (50),
gsa_indicator_flag	VARCHAR2 (1),
jgzz_fiscal_code	VARCHAR2 (20),
analysis_fy	VARCHAR2 (5),
fiscal_yearend_month	VARCHAR2 (30),
curr_fy_potential_revenue	NUMBER,
next_fy_potential_revenue	NUMBER,
year_established	NUMBER,
mission_statement	VARCHAR2 (2000),
organization_type	VARCHAR2 (30),
business_scope	VARCHAR2 (20),
corporation_class	VARCHAR2 (60),
known_as	VARCHAR2 (240),
known_as2	VARCHAR2 (240),
known_as3	VARCHAR2 (240),
known_as4	VARCHAR2 (240),
known_as5	VARCHAR2 (240),
local_bus_iden_type	VARCHAR2 (30),
local_bus_identifier	VARCHAR2 (60),
pref_functional_currency	VARCHAR2 (30),
registration_type	VARCHAR2 (30),
total_employees_text	VARCHAR2 (60),
total_employees_ind	VARCHAR2 (30),
total_emp_est_ind	VARCHAR2 (30),
total_emp_min_ind	VARCHAR2 (30),
parent_sub_ind	VARCHAR2 (30),
incorp_year	NUMBER,
sic_code_type	VARCHAR2 (30),
public_private_ownership_flag	VARCHAR2 (1),
internal_flag	VARCHAR2 (30),
local_activity_code_type	VARCHAR2 (30),
local_activity_code	VARCHAR2 (30),
emp_at_primary_adr	VARCHAR2 (10),
emp_at_primary_adr_text	VARCHAR2 (12),
emp_at_primary_adr_est_ind	VARCHAR2 (30),
emp_at_primary_adr_min_ind	VARCHAR2 (30),
high_credit	NUMBER,
avg_high_credit	NUMBER,
total_payments	NUMBER,

---

credit_score_class	NUMBER,
credit_score_nat1_percentile	NUMBER,
credit_score_incd_default	NUMBER,
credit_score_age	NUMBER,
credit_score_date	DATE,
credit_score_commentary2	VARCHAR2 (30),
credit_score_commentary3	VARCHAR2 (30),
credit_score_commentary4	VARCHAR2 (30),
credit_score_commentary5	VARCHAR2 (30),
credit_score_commentary6	VARCHAR2 (30),
credit_score_commentary7	VARCHAR2 (30),
credit_score_commentary8	VARCHAR2 (30),
credit_score_commentary9	VARCHAR2 (30),
credit_score_commentary10	VARCHAR2 (30),
failure_score_class	NUMBER,
failure_score_incd_default	NUMBER,
failure_score_age	NUMBER,
failure_score_date	DATE,
failure_score_commentary2	VARCHAR2 (30),
failure_score_commentary3	VARCHAR2 (30),
failure_score_commentary4	VARCHAR2 (30),
failure_score_commentary5	VARCHAR2 (30),
failure_score_commentary6	VARCHAR2 (30),
failure_score_commentary7	VARCHAR2 (30),
failure_score_commentary8	VARCHAR2 (30),
failure_score_commentary9	VARCHAR2 (30),
failure_score_commentary10	VARCHAR2 (30),
maximum_credit_recommendation	NUMBER,
maximum_credit_currency_code	VARCHAR2 (240),
displayed_duns_party_id	NUMBER,
content_source_type	VARCHAR2 (30), := G_MISS_CONTENT_SOURCE_TYPE,
content_source_number	VARCHAR2 (30),
attribute_category	VARCHAR2 (30),
attribute1	VARCHAR2 (150),
attribute2	VARCHAR2 (150),
attribute3	VARCHAR2 (150),
attribute4	VARCHAR2 (150),
attribute5	VARCHAR2 (150),
attribute6	VARCHAR2 (150),
attribute7	VARCHAR2 (150),
attribute8	VARCHAR2 (150),
attribute9	VARCHAR2 (150),
attribute10	VARCHAR2 (150),
attribute11	VARCHAR2 (150),
attribute12	VARCHAR2 (150),

```
attribute13          VARCHAR2(150),
attribute14          VARCHAR2(150),
attribute15          VARCHAR2(150),
attribute16          VARCHAR2(150),
attribute17          VARCHAR2(150),
attribute18          VARCHAR2(150),
attribute19          VARCHAR2(150),
attribute20          VARCHAR2(150),
created_by_module    VARCHAR2(150),
application_id       NUMBER,
do_not_confuse_with  VARCHAR2(255),
actual_content_source VARCHAR2(30) := G_SST_SOURCE_TYPE,
party_rec            PARTY_REC_TYPE:= G_MISS_PARTY_REC
)

```

## PL/SQL Record Structure for Person

```
TYPE person_rec_type IS RECORD(
  person_pre_name_adjunct VARCHAR2(30),
  person_first_name       VARCHAR2(150),
  person_middle_name      VARCHAR2(60),
  person_last_name        VARCHAR2(150),
  person_name_suffix      VARCHAR2(30),
  person_title            VARCHAR2(60),
  person_academic_title   VARCHAR2(30),
  person_previous_last_name VARCHAR2(150),
  person_initials         VARCHAR2(6),
  known_as                VARCHAR2(240),
  known_as2               VARCHAR2(240),
  known_as3               VARCHAR2(240),
  known_as4               VARCHAR2(240),
  known_as5               VARCHAR2(240),
  person_name_phonetic    VARCHAR2(320),
  person_first_name_phonetic VARCHAR2(60),
  person_last_name_phonetic VARCHAR2(60),
  middle_name_phonetic    VARCHAR2(60),
  tax_reference           VARCHAR2(50),
  jgzz_fiscal_code        VARCHAR2(20),
  person_iden_type        VARCHAR2(30),
  person_identifier       VARCHAR2(60),
  date_of_birth           DATE,
  place_of_birth          VARCHAR2(60),
  date_of_death           DATE,
  gender                  VARCHAR2(30),
  declared_ethnicity      VARCHAR2(60),

```

```

marital_status                VARCHAR2(30),
marital_status_effective_date DATE,
personal_income               NUMBER,
head_of_household_flag       VARCHAR2(1),
household_income              NUMBER,
household_size                 NUMBER,
rent_own_ind                   VARCHAR2(30),
last_known_gps                 VARCHAR2(60),
content_source_type           VARCHAR2(30) := G_MISS_CONTENT_SOURCE_TYPE,
internal_flag                  VARCHAR2(2),
attribute_category            VARCHAR2(30),
attribute1                     VARCHAR2(150),
attribute2                     VARCHAR2(150),
attribute3                     VARCHAR2(150),
attribute4                     VARCHAR2(150),
attribute5                     VARCHAR2(150),
attribute6                     VARCHAR2(150),
attribute7                     VARCHAR2(150),
attribute8                     VARCHAR2(150),
attribute9                     VARCHAR2(150),
attribute10                    VARCHAR2(150),
attribute11                    VARCHAR2(150),
attribute12                    VARCHAR2(150),
attribute13                    VARCHAR2(150),
attribute14                    VARCHAR2(150),
attribute15                    VARCHAR2(150),
attribute16                    VARCHAR2(150),
attribute17                    VARCHAR2(150),
attribute18                    VARCHAR2(150),
attribute19                    VARCHAR2(150),
attribute20                    VARCHAR2(150),
created_by_module              VARCHAR2(150),
application_id                 NUMBER,
actual_content_source          VARCHAR2(30) := G_SST_SOURCE_TYPE,
party_rec                      PARTY_REC_TYPE := G_MISS_PARTY_REC
)

```

## PL/SQL Record Structure for Group

```

TYPE group_rec_type IS RECORD(
    group_name                VARCHAR2(255),
    group_type                 VARCHAR2(30),
    created_by_module          VARCHAR2(150),
    application_id             NUMBER,
    party_rec                  PARTY_REC_TYPE := G_MISS_PARTY_REC
)

```

)

### Java Inner Class for Party

```
        public BigDecimal                party_id;
        public String                    party_number;
        public String                    validated_flag;
        public String                    orig_system_reference;
        public String                    status;
        public String                    category_code;
        public String                    salutation;
        public String                    attribute_category;
        public String                    attribute1;
        public String                    attribute2;
        public String                    attribute3;
        public String                    attribute4;
        public String                    attribute5;
        public String                    attribute6;
        public String                    attribute7;
        public String                    attribute8;
        public String                    attribute9;
        public String                    attribute10;
        public String                    attribute11;
        public String                    attribute12;
        public String                    attribute13;
        public String                    attribute14;
        public String                    attribute15;
        public String                    attribute16;
        public String                    attribute17;
        public String                    attribute18;
        public String                    attribute19;
        public String                    attribute20;
        public String                    attribute21;
        public String                    attribute22;
        public String                    attribute23;
        public String                    attribute24;

        public PartyRec();
        public PartyRec(boolean _RosettaUseGMISValues);
    }
```

### Java Inner Class for Organization

```
    public static class OrganizationRec {
        public String                    organization_name;
```

```
public String          duns_number_c;
public String          enquiry_duns;
public String          ceo_name;
public String          ceo_title;
public String          principal_name;
public String          principal_title;
public String          legal_status;
public BigDecimal      control_yr;
public BigDecimal      employees_total;
public String          hq_branch_ind;
public String          branch_flag;
public String          oob_ind;
public String          line_of_business;
public String          cong_dist_code;
public String          sic_code;
public String          import_ind;
public String          export_ind;
public String          labor_surplus_ind;
public String          debarment_ind;
public String          minority_owned_ind;
public String          minority_owned_type;
public String          woman_owned_ind;
public String          disadv_8a_ind;
public String          small_bus_ind;
public String          rent_own_ind;
public BigDecimal      debarments_count;
public java.sql.Timestamp debarments_date;
public String          failure_score;
public BigDecimal      failure_score_natnl_percentile;
public String          failure_score_override_code;
public String          failure_score_commentary;
public String          global_failure_score;
public String          db_rating;
public String          credit_score;
public String          credit_score_commentary;
public String          paydex_score;
public String          paydex_three_months_ago;
public String          paydex_norm;
public java.sql.Timestamp best_time_contact_begin;
public java.sql.Timestamp best_time_contact_end;
public String          organization_name_phonetic;
public String          tax_reference;
public String          gsa_indicator_flag;
public String          jgzz_fiscal_code;
public String          analysis_fy;
```

```
public String                fiscal_yearend_month;
public BigDecimal            curr_fy_potential_revenue;
public BigDecimal            next_fy_potential_revenue;
public BigDecimal            year_established;
public String                mission_statement;
public String                organization_type;
public String                business_scope;
public String                corporation_class;
public String                known_as;
public String                known_as2;
public String                known_as3;
public String                known_as4;
public String                known_as5;
public String                local_bus_iden_type;
public String                local_bus_identifier;
public String                pref_functional_currency;
public String                registration_type;
public String                total_employees_text;
public String                total_employees_ind;
public String                total_emp_est_ind;
public String                total_emp_min_ind;
public String                parent_sub_ind;
public BigDecimal            incorp_year;
public String                sic_code_type;
public String                public_private_ownership_flag;
public String                internal_flag;
public String                local_activity_code_type;
public String                local_activity_code;
public String                emp_at_primary_adr;
public String                emp_at_primary_adr_text;
public String                emp_at_primary_adr_est_ind;
public String                emp_at_primary_adr_min_ind;
public BigDecimal            high_credit;
public BigDecimal            avg_high_credit;
public BigDecimal            total_payments;
public BigDecimal            credit_score_class;
public BigDecimal            credit_score_natl_percentile;
public BigDecimal            credit_score_incd_default;
public BigDecimal            credit_score_age;
public java.sql.Timestamp    credit_score_date;
public String                credit_score_commentary2;
public String                credit_score_commentary3;
public String                credit_score_commentary4;
public String                credit_score_commentary5;
public String                credit_score_commentary6;
```

```
public String credit_score_commentary7;
public String credit_score_commentary8;
public String credit_score_commentary9;
public String credit_score_commentary10;
public BigDecimal failure_score_class;
public BigDecimal failure_score_incd_default;
public BigDecimal failure_score_age;
public java.sql.Timestamp failure_score_date;
public String failure_score_commentary2;
public String failure_score_commentary3;
public String failure_score_commentary4;
public String failure_score_commentary5;
public String failure_score_commentary6;
public String failure_score_commentary7;
public String failure_score_commentary8;
public String failure_score_commentary9;
public String failure_score_commentary10;
public BigDecimal maximum_credit_recommendation;
public String maximum_credit_currency_code;
public BigDecimal displayed_duns_party_id;
public String content_source_type;
public String content_source_number;
public String attribute_category;
public String attribute1;
public String attribute2;
public String attribute3;
public String attribute4;
public String attribute5;
public String attribute6;
public String attribute7;
public String attribute8;
public String attribute9;
public String attribute10;
public String attribute11;
public String attribute12;
public String attribute13;
public String attribute14;
public String attribute15;
public String attribute16;
public String attribute17;
public String attribute18;
public String attribute19;
public String attribute20;
public String do_not_confuse_with;
public String actual_content_source;
```

```
        public String                created_by_module;
        public BigDecimal             application_id;
        public PartyRec              party_rec;

        public OrganizationRec();
        public OrganizationRec(boolean_RosettaUseGMISSValues);
    }
```

## Java Inner Class for Person

```
public static class PersonRec {
    public String                person_pre_name_adjunct;
    public String                person_first_name;
    public String                person_middle_name;
    public String                person_last_name;
    public String                person_name_suffix;
    public String                person_title;
    public String                person_academic_title;
    public String                person_previous_last_name;
    public String                person_initials;
    public String                known_as;
    public String                known_as2;
    public String                known_as3;
    public String                known_as4;
    public String                known_as5;
    public String                person_name_phonetic;
    public String                person_first_name_phonetic;
    public String                person_last_name_phonetic;
    public String                middle_name_phonetic;
    public String                tax_reference;
    public String                jgzz_fiscal_code;
    public String                person_iden_type;
    public String                person_identifier;
    public java.sql.Timestamp    date_of_birth;
    public String                place_of_birth;
    public java.sql.Timestamp    date_of_death;
    public String                gender;
    public String                declared_ethnicity;
    public String                marital_status;
    public java.sql.Timestamp    marital_status_effective_date;
    public BigDecimal            personal_income;
    public String                head_of_household_flag;
    public BigDecimal            household_income;
    public BigDecimal            household_size;
    public String                rent_own_ind;
```

```
public String          last_known_gps;
public String          content_source_type;
public String          internal_flag;
public String          attribute_category;
public String          attribute1;
public String          attribute2;
public String          attribute3;
public String          attribute4;
public String          attribute5;
public String          attribute6;
public String          attribute7;
public String          attribute8;
public String          attribute9;
public String          attribute10;
public String          attribute11;
public String          attribute12;
public String          attribute13;
public String          attribute14;
public String          attribute15;
public String          attribute16;
public String          attribute17;
public String          attribute18;
public String          attribute19;
public String          attribute20;
public String          created_by_module;
public BigDecimal      application_id;
public String          actual_content_source;
public PartyRec        party_rec;

public PersonRec();
public PersonRec(boolean __RosettaUseGMISSValues);
}
```

### Java Inner Class for Group

```
public static class GroupRec {
    public String          group_name;
    public String          group_type;
    public String          created_by_module;
    public BigDecimal      application_id;
    public PartyRec        party_rec;

    public GroupRec();
    public GroupRec(boolean_RosettaUseGMISSValues);
}
```

## Create Organization API

### Description

This routine is used to create an organization. The API creates a record in the HZ\_PARTIES table with Organization party type. The HZ\_PARTIES table holds the basic information about the party. The API additionally creates a record in the HZ\_ORGANIZATION\_PROFILES table. That record holds more detail and specific information about the organization.

### PL/SQL Procedure

```
PROCEDURE create_organization (  
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,  
    p_organization_rec       IN          ORGANIZATION_REC_TYPE,  
    x_return_status         OUT         VARCHAR2,  
    x_msg_count             OUT         NUMBER,  
    x_msg_data              OUT         VARCHAR2,  
    x_party_id              OUT         NUMBER,  
    x_party_number          OUT         VARCHAR2,  
    x_profile_id            OUT         NUMBER  
)
```

### Java Method

```
public static void createOrganization(  
    OracleConnection_connection,  
    String p_init_msg_list,  
    OrganizationRec p_organization_rec,  
    String [ ] x_return_status,  
    BigDecimal [ ] x_msg_count,  
    String [ ] x_msg_data,  
    BigDecimal [ ] x_party_id,  
    String [ ] x_party_number,  
    BigDecimal [ ] x_profile_id  
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Organization API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of each parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
organization_name	IN	VARCHAR2	Yes	Validation: Mandatory attribute
duns_number_c	IN	VARCHAR2	No	
enquiry_duns	IN	VARCHAR2	No	
ceo_name	IN	VARCHAR2	No	
ceo_title	IN	VARCHAR2	No	
principal_name	IN	VARCHAR2	No	
principal_title	IN	VARCHAR2	No	
legal_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type LEGAL_STATUS
control_yr	IN	NUMBER	No	
employees_total	IN	NUMBER	No	
hq_branch_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type HQ_BRANCH_IND
branch_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
oob_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
line_of_business	IN	VARCHAR2	No	
cong_dist_code	IN	VARCHAR2	No	
sic_code	IN	VARCHAR2	No	
import_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
export_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
labor_surplus_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
debarment_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
minority_owned_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
minority_owned_type	IN	VARCHAR2	No	
woman_owned_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
disadv_8a_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
small_bus_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
rent_own_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
debarments_count	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
debarments_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_natnl_percentile	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_override_code	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY
global_failure_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
db_rating	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY Comment: This parameter should only be populated with data provided by D&B.
paydex_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_three_months_ago	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_norm	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
best_time_contact_begin	IN	DATE	No	
best_time_contact_end	IN	DATE	No	
organization_name_phonetic	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	
gsa_indicator_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
gjzz_fiscal_code	IN	VARCHAR2	No	
analysis_fy	IN	VARCHAR2	No	
fiscal_yearend_month	IN	VARCHAR2	No	Validation: Validated against AR lookup type MONTH
curr_fy_potential_revenue	IN	NUMBER	No	
next_fy_potential_revenue	IN	NUMBER	No	
year_established	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
mission_statement	IN	VARCHAR2	No	
organization_type	IN	VARCHAR2	No	
business_scope	IN	VARCHAR2	No	
corporation_class	IN	VARCHAR2	No	
known_as	IN	VARCHAR2	No	
known_as2	IN	VARCHAR2	No	
known_as3	IN	VARCHAR2	No	
known_as4	IN	VARCHAR2	No	
known_as5	IN	VARCHAR2	No	
local_bus_iden_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type LOCAL_BUS_IDEN_TYPE
local_bus_identifier	IN	VARCHAR2	No	
pref_functional_currency	IN	VARCHAR2	No	
registration_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRATION_TYPE
total_employees_text	IN	VARCHAR2	No	
total_employees_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
total_emp_est_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
total_emp_min_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
parent_sub_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
incorp_year	IN	NUMBER	No	
sic_code_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type SIC_CODE_TYPE
public_private_ownership_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
internal_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Default: N
local_activity_code_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type LOCAL_ACTIVITY_CODE_TYPE
local_activity_code	IN	VARCHAR2	No	
emp_at_primary_adr	IN	VARCHAR2	No	
emp_at_primary_adr_text	IN	VARCHAR2	No	
emp_at_primary_adr_est_ind	IN	VARCHAR2	No	
emp_at_primary_adr_min_ind	IN	VARCHAR2	No	
high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
avg_high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
total_payments	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_class	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_natl_percentile	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_incd_default	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_age	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
credit_score_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary2	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary3	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary4	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary5	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary6	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary7	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
credit_score_commentary8	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary9	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary10	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_class	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_incd_default	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_age	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary2	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
failure_score_commentary3	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary4	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary5	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary6	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary7	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary8	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
failure_score_commentary9	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary10	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
maximum_credit_recommendation	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
maximum_credit_currency_code	IN	VARCHAR2	No	Validation: Foreign Key to fnd_currencies.currency_code  Comment: This parameter should only be populated with data provided by D&B.
displayed_duns_party_id	IN	NUMBER	No	Validation: Foreign Key to HZ_PARTIES.PARTY_ID  Comment: This parameter should only be populated with data provided by D&B.
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE  Default: USER_ENTERED
content_source_number	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
do_not_confuse_with	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	No	Validation : Validated against AR lookup type CONTENT_SOURCE_TYPE. Default - SST
party_recRecord Type				

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated by from sequence
party_number	IN	VARCHAR2	Yes/No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else mandatory.
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id
status	IN	VARCHAR2	No	Comment: This attribute is no longer used. Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_party_id	OUT	NUMBER	No	Comment: party_id of the party created
x_party_number	OUT	VARCHAR2	No	Comment: party_number of the party created
x_profile_id	OUT	NUMBER	No	Comment: organization_profile_id of the organization profile record created

## Update Organization API

### Description

This routine is used to update an organization. The API updates the party record for the organization in the HZ\_PARTIES table. The API additionally creates or updates a record in the HZ\_ORGANIZATION\_PROFILES table. If an organization is updated on the same day as it is created, the active profile record is updated. Otherwise, a new profile record that is created and the old profile record is end dated.

### PL/SQL Procedure

```
PROCEDURE update_organization (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_organization_rec       IN          ORGANIZATION_REC_TYPE,
    p_party_object_version_number IN OUT  NUMBER,
    x_profile_id             OUT         NUMBER,
    x_return_status          OUT         VARCHAR2,
```

```

        x_msg_count          OUT          NUMBER,
        x_msg_data          OUT          VARCHAR2
    )

```

### Java Method

```

public static void updateOrganization(
    OracleConnection_connection,
    String                    p_init_msg_list,
    OrganizationRec           p_organization_rec,
    BigDecimal [ ]           p_party_object_version_number,
    BigDecimal [ ]           x_profile_id,
    String [ ]               x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Organization API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
organization name	IN	VARCHAR2	Yes	Validation: Cannot be updated to null
duns_number_c	IN	VARCHAR2	No	
enquiry_duns	IN	VARCHAR2	No	
ceo_name	IN	VARCHAR2	No	
ceo_title	IN	VARCHAR2	No	
principal_name	IN	VARCHAR2	No	
principal_title	IN	VARCHAR2	No	
legal_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type LEGAL_STATUS
control_yr	IN	NUMBER	No	
employees_total	IN	NUMBER	No	
hq_branch_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type HQ_BRANCH_ID

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
branch_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
oob_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
line_of_business	IN	VARCHAR2	No	
cong_dist_code	IN	VARCHAR2	No	
sic_code	IN	VARCHAR2	No	
import_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
export_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
labor_surplus_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
debarment_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Comment: This parameter should only be populated with data provided by D&B.
minority_owned_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
minority_owned_type	IN	VARCHAR2	No	
woman_owned_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
disadv_8a_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
small_bus_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
rent_own_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
debarments_count	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
debarments_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_natnl_percentile	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_override_code	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY Comment: This parameter should only be populated with data provided by D&B.
global_failure_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
db_rating	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY Comment: This parameter should only be populated with data provided by D&B.
paydex_score	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_three_months_ago	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
paydex_norm	IN	VARCHAR2	No	Comment: This parameter should only be populated with data provided by D&B.
best_time_contact_begin	IN	DATE	No	
best_time_contact_end	IN	DATE	No	
organization_name_phonetic	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	Validation: Validated against AR lookup type MONTH
gsa_indicator_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
jjzz_fiscal_code	IN	VARCHAR2	No	
analysis_fy	IN	VARCHAR2	No	
fiscal YEAREND_month	IN	VARCHAR2	No	
curr_fy_potential_revenue	IN	NUMBER	No	
next_fy_potential_revenue	IN	NUMBER	No	
year_established	IN	NUMBER	No	
mission_statement	IN	VARCHAR2	No	
organization_type	IN	VARCHAR2	No	
business_scope	IN	VARCHAR2	No	
corporation_class	IN	VARCHAR2	No	
known_as	IN	VARCHAR2	No	
known_as2	IN	VARCHAR2	No	
known_as3	IN	VARCHAR2	No	
known_as4	IN	VARCHAR2	No	
known_as5	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
local_bus_iden_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type LOCAL_BUS_IDEN_TYPE
local_bus_identifier	IN	VARCHAR2	No	
pref_functional_currency	IN	VARCHAR2	No	
registration_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRATION_TYPE
total_employees_text	IN	VARCHAR2	No	
total_employees_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
total_emp_est_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
total_emp_min_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
parent_sub_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
incorp_year	IN	NUMBER	No	
sic_code_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type SIC_CODE_TYPE
public_private_ownership_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
internal_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
local_activity_code_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type LOCAL_ACTIVITY_CODE_TYPE
local_activity_code	IN	VARCHAR2	No	
emp_at_primary_adr	IN	VARCHAR2	No	
emp_at_primary_adr_text	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
emp_at_primary_ adr_est_ind	IN	VARCHAR2	No	
emp_at_primary_ adr_min_ind	IN	VARCHAR2	No	
high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
avg_high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
total_payments	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_natl_ percentile	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_incd_ default	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_age	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_ commentary2	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_ commentary3	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
credit_score_commentary4	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary5	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary6	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary7	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary8	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary9	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
credit_score_commentary10	IN	VARCHAR2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_class	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_incd_default	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_age	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary2	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary3	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary4	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
failure_score_commentary5	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary6	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary7	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary8	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary9	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary10	IN	VARCHAR2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.
maximum_credit_recommendation	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
maximum_credit_currency_code	IN	VARCHAR2	No	Validation: Foreign key to FND_CURRENCIES.CURRENCY_CODE  Comment: This parameter should only be populated with data provided by D&B.
displayed_duns_party_id	IN	NUMBER	No	Validation: Foreign key to HZ_PARTIES.PARTY_ID
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Cannot be updated
content_source_number	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_ module	IN	VARCHAR2	No	Validation: Non updateable if value exists
do_not_confuse_ with	IN	VARCHAR2	No	
actual_content_ source	IN	VARCHAR2	No	Validation : Cannot be updated
party_rec Record Type				
party_id	IN	NUMBER	Yes	Validation: Valid party_id from HZ_PARTIES table  Comment: Pass the party_id from HZ_PARTIES record for the organization
party_number	IN	VARCHAR2	No	Validation: Non updateable
validated_flag	IN	VARCHAR2	No	
orig_system_ reference	IN	VARCHAR2	No	Validation: Non updateable
status	IN	VARCHAR2	No	Comment: This column is no longer used.  Validation: Validated against AR lookup type REGISTRY_STATUS. Cannot be updated to null
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
p_party_object_ version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing party record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_ version_number of the record from hz_parties</li> <li>■ Return new value after update</li> </ul>
x_profile_id	OUT	NUMBER	No	Comment: Returns organization_ profile_id of the profile record created or updated

## Create Person API

### Description

This routine is used to create a person. The API creates a record in the HZ\_PARTIES table with party type 'PERSON'. The HZ\_PARTIES table contains basic information about the party. The API also creates a record in the HZ\_PERSON\_PROFILES table. That record holds more detail and specific information about the person.

### PL/SQL Procedure

```
PROCEDURE create_person (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_person_rec             IN          PERSON_REC_TYPE,
    x_party_id              OUT         NUMBER,
    x_party_number          OUT         VARCHAR2,
    x_profile_id            OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

### Java Method

```
public static void createPerson(
    OracleConnection_connection,
    String,
    PersonRec,
    p_init_msg_list,
    p_person_rec,
```

```

        BigDecimal [ ]           x_party_id,
        String [ ]              x_party_number,
        BigDecimal [ ]         x_profile_id,
        String [ ]             x_return_status,
        BigDecimal [ ]         x_msg_count,
        String [ ]             x_msg_data
    ) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Person API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
person_pre_name_adjunct	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTACT_TITLE
person_first_name	IN	VARCHAR2	Yes/No	Validation: Either one of person_first_name or person_last_name should be passed in
person_middle_name	IN	VARCHAR2	No	
person_last_name	IN	VARCHAR2	Yes/No	Validation: Either one of person_first_name or person_last_name should be passed in
person_name_suffix	IN	VARCHAR2	No	
person_title	IN	VARCHAR2	No	
person_academic_title	IN	VARCHAR2	No	
person_previous_last_name	IN	VARCHAR2	No	
person_initials	IN	VARCHAR2	No	
known_as	IN	VARCHAR2	No	
known_as2	IN	VARCHAR2	No	
known_as3	IN	VARCHAR2	No	
known_as4	IN	VARCHAR2	No	
known_as5	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_name_phonetic	IN	VARCHAR2	No	
person_first_name_phonetic	IN	VARCHAR2	No	
person_last_name_phonetic	IN	VARCHAR2	No	
middle_name_phonetic	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	
jjzz_fiscal_code	IN	VARCHAR2	No	
person_iden_type	IN	VARCHAR2	No	
person_identifier	IN	VARCHAR2	No	
date_of_birth	IN	DATE	No	
place_of_birth	IN	VARCHAR2	No	
date_of_death	IN	DATE	No	
gender	IN	VARCHAR2	No	
declared_ethnicity	IN	VARCHAR2	No	
marital_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type MARITAL_STATUS
marital_status_effective_date	IN	DATE	No	
personal_income	IN	NUMBER	No	
head_of_household_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
household_income	IN	NUMBER	No	
household_size	IN	NUMBER	No	
rent_own_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
last_known_gps	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
content_source_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE. Default: USER_ENTERED Comment: This parameter is no longer used. Use actual_content_source.
internal_flag	IN	VARCHAR2	No	Default: N
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
actual_content_source	IN	VARCHAR2	No	Validation : Validated against AR lookup type CONTENT_SOURCE_TYPE Default - SST
Party_rec Record Type Attributes				
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
party_number	IN	VARCHAR2	Yes/No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else unique and mandatory
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_party_id	OUT	NUMBER	No	Comment: party_id of the party created
x_party_number	OUT	VARCHAR2	No	Comment: party_number of the party created
x_profile_id	OUT	NUMBER	No	Comment: person_profile_id of the person profile record created

## Update Person API

### Description

This routine is used to update a person. The API updates the party record for the person in the HZ\_PARTIES table. The API additionally creates or updates a record in the HZ\_PERSON\_PROFILES table. If a person is updated on the same day that it

is created, the active profile record is updated. Otherwise a new profile record is created and the old profile record is end dated.

### PL/SQL Procedure

```
PROCEDURE update_person (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_person_rec            IN          PERSON_REC_TYPE,
    p_party_object_version_number IN OUT  NUMBER,
    x_profile_id            OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
)
```

### Java Method

```
public static void updatePerson(
    OracleConnection connection,
    String p_init_msg_list,
    PersonRec p_person_rec,
    BigDecimal [ ] p_party_object_version_number,
    BigDecimal [ ] x_profile_id,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Person API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
person_pre_name_adjunct	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTACT_TITLE
person_first_name	IN	VARCHAR2	No	Validation: During update both person_first_name and person_last_name cannot be set to null
person_middle_name	IN	VARCHAR2	No	
person_last_name	IN	VARCHAR2	No	Validation: During update both person_first_name and person_last_name cannot be set to null

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_name_suffix	IN	VARCHAR2	No	
person_title	IN	VARCHAR2	No	
person_academic_title	IN	VARCHAR2	No	
person_previous_last_name	IN	VARCHAR2	No	
person_initials	IN	VARCHAR2	No	
known_as	IN	VARCHAR2	No	
known_as2	IN	VARCHAR2	No	
known_as3	IN	VARCHAR2	No	
known_as4	IN	VARCHAR2	No	
known_as5	IN	VARCHAR2	No	
person_name_phonetic	IN	VARCHAR2	No	
person_first_name_phonetic	IN	VARCHAR2	No	
person_last_name_phonetic	IN	VARCHAR2	No	
middle_name_phonetic	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	
jgzz_fiscal_code	IN	VARCHAR2	No	
person_iden_type	IN	VARCHAR2	No	
person_identifier	IN	VARCHAR2	No	
date_of_birth	IN	DATE	No	
place_of_birth	IN	VARCHAR2	No	
date_of_death	IN	DATE	No	
gender	IN	VARCHAR2	No	
declared_ethnicity	IN	VARCHAR2	No	
marital_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type MARITAL_STATUS

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
marital_status_effective_date	IN	DATE	No	
personal_income	IN	NUMBER	No	
head_of_household_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
household_income	IN	NUMBER	No	
household_size	IN	NUMBER	No	
rent_own_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
last_known_gps	IN	VARCHAR2	No	
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Cannot be updated
internal_flag	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Cannot be updated if value exists
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
actual_content_source	IN	VARCHAR2	No	Validation : Cannot be updated
Party_rec Record Type Attributes				
party_id	IN	NUMBER	Yes	Validation: Valid party id in hz_parties Comment: Pass the party_id from hz_parties record for the person
party_number	IN	VARCHAR2	No	Validation: Non updateable
validated_flag	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Non updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS.</li> <li>■ Cannot be updated to null</li> </ul>
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
p_party_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing party record.</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_PARTIES</li> <li>■ Return new value after update</li> </ul>
x_profile_id	OUT	NUMBER	No	Comment: person_profile_id of the person profile record created

## Create Group API

### Description

This routine is used to create a group. The API creates a record in the HZ\_PARTIES table with party type 'GROUP'. The HZ\_PARTIES table holds the basic information about the party. There is no profile information for a Group party unlike an Organization or Person party.

### PL/SQL Procedure

```

PROCEDURE create_group (
    p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,
    p_group_rec              IN          GROUP_REC_TYPE,
    x_party_id               OUT         NUMBER,
    x_party_number           OUT         VARCHAR2,
    x_return_status          OUT         VARCHAR2,
    x_msg_count              OUT         NUMBER,
    x_msg_data               OUT         VARCHAR2
)

```

### Java Method

```

public static void createGroup(
    OracleConnection_connection,
    String,
    GroupRec,
    BigDecimal [ ],
    String [ ]
    p_init_msg_list,
    p_group_rec,
    x_party_id,
    x_party_number,

```

```

    String [ ]                x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Group API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
group_name	IN	VARCHAR2	Yes	Validation: Mandatory attribute
group_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
party_rec Record Type Attributes				
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
party_number	IN	VARCHAR2	Yes/No	Validation: Generated from sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else unique and mandatory
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_party_id	OUT	NUMBER	No	Comment: party_id of the party created
x_party_number	OUT	VARCHAR2	No	Comment: party_number of the party created

## Update Group API

### Description

This routine is used to update a group. The API updates the party record for the group in the HZ\_PARTIES table.

### PL/SQL Procedure

```
PROCEDURE update_group (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_group_rec             IN          GROUP_REC_TYPE,
    p_party_object_version_number IN OUT  NUMBER,
    x_return_status         OUT        VARCHAR2,
    x_msg_count             OUT        NUMBER,
    x_msg_data              OUT        VARCHAR2
)
```

### Java Method

```
public static void updateGroup(
    OracleConnection_connection,
    String                p_init_msg_list,
    GroupRec              p_group_rec,
    BigDecimal [ ]       p_party_object_version_number,
    String [ ]           x_return_status,
    BigDecimal [ ]       x_msg_count,
    String [ ]           x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Group API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
group_name	IN	VARCHAR2	No	Validation: Cannot be updated to null
group_type	IN	VARCHAR2	No	Validation: Cannot be updated to null
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Validation: Non Updateable if value exists
party_rec Record Type Attributes				
party_id	IN	NUMBER	Yes	Validation: Valid party id in hz_parties table Comment: Pass the party_id from hz_parties record for the group
party_number	IN	VARCHAR2	No	Validation: Non updateable
validated_flag	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Non updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS</li> <li>■ Cannot be updated to null</li> </ul>
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
p_party_object_version_number	IN/O UT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing party record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_PARTIES</li> <li>■ Return new value after update.</li> </ul>



---

---

## Party Contact API Use

This chapter provides information about the Party Contact application programming interfaces.

The information provided for the API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Party Contact APIs

**PL/SQL Package Name: HZ\_PARTY\_CONTACT\_V2PUB**

**Java Class Name: HzPartyContactV2Pub**

### PL/SQL Record Structure for Org Contact

```
TYPE org_contact_rec_type IS RECORD(  
    org_contact_id                NUMBER,  
    comments                      VARCHAR2(240),  
    contact_number                VARCHAR2(30),  
    department_code              VARCHAR2(30),  
    department                    VARCHAR2(60),  
    title                         VARCHAR2(30),  
    job_title                     VARCHAR2(100),  
    decision_maker_flag          VARCHAR2(1),  
    job_title_code                VARCHAR2(30),  
    reference_use_flag            VARCHAR2(1),  
    rank                          VARCHAR2(30),  
    party_site_id                NUMBER,  
    orig_system_reference         VARCHAR2(240),  
    attribute_category            VARCHAR2(30),  
    attribute1                    VARCHAR2(150),  
    attribute2                    VARCHAR2(150),  
    attribute3                    VARCHAR2(150),  
    attribute4                    VARCHAR2(150),  
    attribute5                    VARCHAR2(150),  
    attribute6                    VARCHAR2(150),  
    attribute7                    VARCHAR2(150),  
    attribute8                    VARCHAR2(150),  
    attribute9                    VARCHAR2(150),  
    attribute10                   VARCHAR2(150),  
    attribute11                   VARCHAR2(150),  
    attribute12                   VARCHAR2(150),  
    attribute13                   VARCHAR2(150),  
    attribute14                   VARCHAR2(150),  
    attribute15                   VARCHAR2(150),  
    attribute16                   VARCHAR2(150),  
    attribute17                   VARCHAR2(150),  
    attribute18                   VARCHAR2(150),  
    attribute19                   VARCHAR2(150),  
    attribute20                   VARCHAR2(150),  
    attribute21                   VARCHAR2(150),
```

```

attribute22                VARCHAR2(150),
attribute23                VARCHAR2(150),
attribute24                VARCHAR2(150),
created_by_module         VARCHAR2(150),
application_id            NUMBER,
party_rel_rec             HZ_RELATIONSHIP_V2PUB.relationship_rec_
type:=HZ_RELATIONSHIP_V2PUB.G_MISS_REL_REC
)

```

## PL/SQL Record Structure for Org Contact Role

```

TYPE org_contact_role_rec_type IS RECORD(
  org_contact_role_id      NUMBER,
  role_type                VARCHAR2(30),
  primary_flag             VARCHAR2(1),
  org_contact_id           NUMBER,
  orig_system_reference    VARCHAR2(240),
  role_level               VARCHAR2(30),
  primary_contact_per_role_type VARCHAR2(1),
  status                   VARCHAR2(1),
  created_by_module        VARCHAR2(150),
  application_id           NUMBER
)

```

## Java Inner Class for Org Contact

```

public static class OrgContactRec {
  public BigDecimal      org_contact_id;
  public String          comments;
  public String          contact_number;
  public String          department_code;
  public String          department;
  public String          title;
  public String          job_title;
  public String          decision_maker_flag;
  public String          job_title_code;
  public String          reference_use_flag;
  public String          rank;
  public BigDecimal      party_site_id;
  public String          orig_system_reference;
  public String          attribute_category;
  public String          attribute1;
  public String          attribute2;
  public String          attribute3;
  public String          attribute4;
}

```

```

        public String          attribute5;
        public String          attribute6;
        public String          attribute7;
        public String          attribute8;
        public String          attribute9;
        public String          attribute10;
        public String          attribute11;
        public String          attribute12;
        public String          attribute13;
        public String          attribute14;
        public String          attribute15;
        public String          attribute16;
        public String          attribute17;
        public String          attribute18;
        public String          attribute19;
        public String          attribute20;
        public String          attribute21;
        public String          attribute22;
        public String          attribute23;
        public String          attribute24;
        public String          created_by_module;
        public BigDecimal      application_id;

        party_rel_rec;
        public HzRelationshipV2Pub.RelationshipRec
        public OrgContactRec();
        public OrgContactRec(boolean __RosettaUseGMISSValues);
    }

```

## Java Inner Class for Org Contact Role

```

    public static class OrgContactRoleRec {
        public BigDecimal      org_contact_role_id;
        public String          role_type;
        public String          primary_flag;
        public BigDecimal      org_contact_id;
        public String          orig_system_reference;
        public String          role_level;
        public String          primary_contact_per_role_type;
        public String          status;
        public String          created_by_module;
        public BigDecimal      application_id;

        public OrgContactRoleRec();
        public OrgContactRoleRec(boolean __RosettaUseGMISSValues);
    }

```

}

## Create Org Contact API

### Description

This routine is used to create a Contact person for an organization or person. The API creates a record in the HZ\_ORG\_CONTACTS table. It additionally creates a relationship record in the HZ\_RELATIONSHIPS table using the contact person as the subject, the organization or person as object and relationship type and code passed by the caller. A reverse relationship record is also created at the same time. There is a denormalized party record of type PARTY\_RELATIONSHIP created for the relationship depending on relationship type set up for the relationship that is being used for the org contact.

### PL/SQL Procedure

```
PROCEDURE create_org_contact (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_org_contact_rec       IN          ORG_CONTACT_REC_TYPE,
    x_org_contact_id        OUT         NUMBER,
    x_party_rel_id          OUT         NUMBER,
    x_party_id              OUT         NUMBER,
    x_party_number          OUT         VARCHAR2,
    x_return_status        OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

### Java Method

```
public static void createOrgContact(
    OracleConnection_connection,
    String                p_init_msg_list,
    OrgContactRec         p_org_contact_rec,
    BigDecimal [ ]        x_org_contact_id,
    BigDecimal [ ]        x_party_rel_id,
    BigDecimal [ ]        x_party_id,
    String [ ]            x_party_number,
    String [ ]            x_return_status,
    BigDecimal [ ]        x_msg_count,
    String [ ]            x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Org Contact API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
org_contact_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
comments	IN	VARCHAR2	No	
contact_number	IN	VARCHAR2	Yes/No	Comment: If HZ_GENERATE_CONTACT_NUMBER= Y or null and caller does not pass any value, then generated from sequence, otherwise caller is passed value is accepted.
department_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type DEPARTMENT_TYPE
department	IN	VARCHAR2	No	
title	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTACT_TITLE
job_title	IN	VARCHAR2	No	
decision_maker_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
job_title_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type RESPONSIBILITY
reference_use_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
rank	IN	VARCHAR2	No	
party_site_id	IN	NUMBER	No	Validation: Foreign key to HZ_PARTY_SITES.PARTY_SITE_ID. If a value is passed, then the party_id of the party site should be same as the object_id of the relationship to be created for this org contact.
orig_system_reference	IN	VARCHAR2	No	Default: org_contact_id

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
party_rel_rec Record Type Attributes				
relationship_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.
subject_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to HZ_PARTIES.PARTY_ID</li> </ul> Comment: Pass the party_id of the contact person here.
subject_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to FND_OBJECT_INSTANCE_SETS.INSTANCESET_NAME</li> </ul> Comment: Pass the party_type of the subject person, which is PERSON here.
subject_table_name	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to FND_OBJECTS.OBJ.NAME</li> </ul> Comment: Pass HZ_PARTIES for the table name of the source of the subject.
object_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to HZ_PARTIES.PARTY_ID</li> </ul> Comment: Pass the party_id of the organization or person for which you are creating the contact.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
object_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign key to FND_OBJECT_INSTANCE_SETS.INSTANCE_SET_NAME</li> </ul> Comment: Pass ORGANIZATION or PERSON depending on whether you are creating contact for an organization or for a person.
object_table_name	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign key to FND_OBJECTS.OBJ_NAME</li> </ul> Comment: Pass HZ_PARTIES as the table name that is the source of the object.
relationship_code	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type PARTY_RELATIONS_TYPE</li> <li>▪ Required to be a valid forward_rel_code for the particular relationship type requested.</li> </ul>
relationship_type	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Must be a valid relationship_type from the HZ_RELATIONSHIP_TYPE table for the combination of subject_type, object_type, and relationship_code passed.</li> </ul>
comments	IN	VARCHAR2	No	
start_date	IN	DATE	No	Validation: Mandatory attribute

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
end_date	IN	DATE	No	Validation: Must not be less than start_date Default: 31-DEC-4712
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE Default: USER_ENTERED
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory Attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
party_rec Record Type Attributes				
party_id	IN	NUMBER	Yes	Validation: Unique if passed in, otherwise generated from sequence
party_number	IN	VARCHAR2	Yes/No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else mandatory
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_org_contact_id	OUT	NUMBER	No	Comment: org_contact_id of the org contact record created
x_party_rel_id	OUT	NUMBER	No	Comment: relationship_id of the relationship record created
x_party_id	OUT	NUMBER	No	Comment: party_id of the party record created
x_party_number	OUT	VARCHAR2	No	Comment: party_number of the party record created

## Update Org Contact API

### Description

This routine is used to update a Contact person. The contact record in the HZ\_ORG\_CONTACTS table is updated by this API. Optionally you can update the relevant relationship record in the HZ\_RELATIONSHIPS table and underlying party record in the HZ\_PARTIES table by calling this API. For that you should pass the corresponding id and object version number.

### PL/SQL Procedure

```
PROCEDURE update_org_contact (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_org_contact_rec       IN          ORG_CONTACT_REC_TYPE,
    p_cont_object_version_number IN OUT  NUMBER,
    p_rel_object_version_number IN OUT  NUMBER,
    p_party_object_version_number IN OUT  NUMBER,
    x_return_status        OUT          VARCHAR2,
    x_msg_count            OUT          NUMBER,
    x_msg_data             OUT          VARCHAR2
)
)
```

### Java Method

```
public static void updateOrgContact(
    OracleConnection_connection,
    String p_init_msg_list,
    OrgContactRec p_org_contact_rec,
    BigDecimal [ ] p_cont_object_version_number,
    BigDecimal [ ] p_rel_object_version_number,
    BigDecimal [ ] p_party_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Org Contact API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
org_contact_id	IN	NUMBER	Yes	Validation: Valid org_contact_id should be passed in. Comment: Pass the org_contact_id from hz_org_contacts table for this org contact.
comments	IN	VARCHAR2	No	
contact_number	IN	VARCHAR2	No	
department_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type DEPARTMENT_TYPE
department	IN	VARCHAR2	No	
title	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTACT_TITLE.
job_title	IN	VARCHAR2	No	
decision_maker_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
job_title_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type RESPONSIBILITY
reference_use_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
rank	IN	VARCHAR2	No	
party_site_id	IN	NUMBER	No	Validation: Foreign key to HZ_PARTY_SITES.PARTY_SITE_ID. If value passed in, then the party_id of the party site should be same as the object_id of the relationship created for this org contact.
orig_system_reference	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if a value exists
application_id	IN	NUMBER	No	Validation: Not updateable if a value exists
party_rel_rec Record Type Attributes				

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_id	IN	NUMBER	Yes/No	Validation: Valid relationship_id should be passed in Comment: <ul style="list-style-type: none"> <li>■ Pass the relationship_id of the relationship record for this org contact.</li> <li>■ Pass if you want to update the relationship record.</li> </ul>
subject_id	IN	NUMBER	No	Validation: Non updateable
subject_type	IN	VARCHAR2	No	Validation: Non updateable
subject_table_name	IN	VARCHAR2	No	Validation: Non updateable
object_id	IN	NUMBER	No	Validation: Non updateable
object_type	IN	VARCHAR2	No	Validation: Non updateable
object_table_name	IN	VARCHAR2	No	Validation: Non updateable
relationship_code	IN	VARCHAR2	No	Validation: Non updateable
relationship_type	IN	VARCHAR2	No	Validation: Non updateable
comments	IN	VARCHAR2	No	
start_date	IN	DATE	No	Validation: Cannot be updated to null
end_date	IN	DATE	No	Validation: Cannot be less than start_date
status	IN	VARCHAR2	No	Validation: Validate against AR lookup type REGISTRY_STATUS Cannot be updated to null
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Not updateable
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
party_rec Record Type Attributes				
party_id	IN	NUMBER	Yes/No	Validation: Valid party id in HZ_PARTIES Comment: <ul style="list-style-type: none"> <li>▪ Pass party_id of the relationship's party record</li> <li>▪ Pass if you want to update the party record</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_number	IN	VARCHAR2	No	Validation: Not updateable
validated_flag	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validate against AR lookup type REGISTRY_STATUS</li> <li>■ Cannot be updated to null</li> </ul>
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
p_cont_object_ version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against the value in the database for the existing org contact record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_ORG_CONTACTS</li> <li>■ Return a new value after update</li> </ul>
p_rel_object_version_ number	IN OUT	NUMBER	Yes/No	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute if relationship to be updated</li> <li>■ Validated against the value in the database for the existing relationship record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_RELATIONSHIPS</li> <li>■ Return a new value after update</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_party_object_ version_number	IN OUT	NUMBER	Yes/No	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute if party to be updated</li> <li>■ Validated against the value in the database for the existing party record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_PARTIES</li> <li>■ Return a new value after update</li> </ul>

## Create Org Contact Role API

### Description

This routine is used to create a Contact Role for a contact person. The API creates a record in the HZ\_ORG\_CONTACT\_ROLES table. You can create multiple role records for a particular org contact. For a particular org contact, one of the org contact role records can be marked as Primary and there can be one role record per role type. For a particular organization or person, among all its org contacts, you can mark one role record per role type as primary.

### PL/SQL Procedure

```
PROCEDURE create_org_contact_role (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_org_contact_role_rec  IN          ORG_CONTACT_ROLE_REC_TYPE,
    x_org_contact_role_id   OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
;
```

### Java Method

```
public static void createOrgContactRole(
    OracleConnection_connection,
    String                                p_init_msg_list,
    OrgContactRoleRec                    p_org_contact_role_rec,
    BigDecimal [ ]                       x_org_contact_role_id,
    String [ ]                            x_return_status,
```

```

        BigDecimal [ ]                x_msg_count,
        String [ ]                    x_msg_data
    ) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Org Contact Role API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
org_contact_role_id	IN	NUMBER	No	Validation: Unique when passed in, else generated from sequence
role_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against AR lookup type CONTACT_ROLE_TYPE</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
org_contact_id	IN	NUMBER	Yes	Validation: <p>Mandatory attribute</p> <p>Foreign key to HZ_ORG_CONTACTS.ORG_CONTACT_ID</p>
orig_system_reference	IN	VARCHAR2	No	Default: org_contact_role_id
role_level	IN	VARCHAR2	No	
primary_contact_per_role_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
x_org_contact_role_id	OUT	NUMBER	No	Comment: Return org_contact_role_id for record created

#### Other Validations

- Primary Flag can be set to 'Y' only for one org contact role record for each org contact.
- The combination of org\_contact\_id and role\_type must be unique.
- Only one org contact can be set as primary within the same organization party.

## Update Org Contact Role API

#### Description

This routine is used to update a contact role record. The API updates the record in the HZ\_ORG\_CONTACT\_ROLES table.

#### PL/SQL Procedure

```
PROCEDURE update_org_contact_role (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_org_contact_role_rec  IN          ORG_CONTACT_ROLE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

#### Java Method

```
public static void updateOrgContactRole(
    OracleConnection_connection,
    String
    OrgContactRoleRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) throws SQLException;
    p_init_msg_list,
    p_org_contact_role_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data

```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Org Contact Role API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
org_contact_role_id	IN	NUMBER	Yes	Validation: Valid org_contact_role_id should be passed in  Comment: Pass the org_contact_role_id from hz_org_contact_roles table
role_type	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Cannot be set to null during update</li> <li>■ Validated against AR lookup type CONTACT_ROLE_TYPE</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
org_contact_id	IN	NUMBER	No	Validation: Not updateable
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
role_level	IN	VARCHAR2	No	
primary_contact_per_role_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS<sup>2</sup></li> <li>■ Cannot be set to null during update</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>■ Mandatory attribute<sup>2</sup></li><li>■ Validated against value in the database for the existing org contact role record</li></ul> Comment: <ul style="list-style-type: none"><li>■ Pass the current object_version_number of the record from hz_org_contact_roles<sup>2</sup></li><li>■ Return new value after update.</li></ul>

---

**Other Validations**

- Primary Flag can be set to 'Y' only for one org contact role record for each org contact.
- The combination of org\_contact\_id and role\_type must be unique.
- Only one org contact can be set as primary within the same organization party.

---

---

# Person Info, Location and Party Site API Use

This chapter provides information about the following application programming interfaces:

- Person Info APIs
- Location APIs
- Party Site APIs

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Person Info APIs

**PL/SQL Package Name: HZ\_PERSON\_INFO\_V2PUB**

**Java Class Name: HzPersonInfoV2Pub**

### PL/SQL Record Structure for Person Language

```
TYPE person_language_rec_type IS RECORD(  
    language_use_reference_id    NUMBER,  
    language_name                VARCHAR2(4),  
    party_id                    NUMBER,  
    native_language              VARCHAR2(1),  
    primary_language_indicator   VARCHAR2(1),  
    reads_level                  VARCHAR2(20),  
    speaks_level                 VARCHAR2(20),  
    writes_level                 VARCHAR2(20),  
    status                       VARCHAR2(1),  
    created_by_module            VARCHAR2(150),  
    application_id               NUMBER  
);
```

### Java Inner Class for Person Language

```
public static class PersonLanguageRec {  
    public BigDecimal    language_use_reference_id;  
    public String        language_name;  
    public BigDecimal    party_id;  
    public String        native_language;  
    public String        primary_language_indicator;  
    public String        reads_level;  
    public String        speaks_level;  
    public String        writes_level;  
    public String        status;  
    public String        created_by_module;  
    public BigDecimal    application_id;  
  
    public PersonLanguageRec();  
    public PersonLanguageRec(boolean_RosettaUseGMISSValues);  
}
```

## Create Person Language API

### Description

This routine is used to create Language for a party. The API creates a record in the HZ\_PERSON\_LANGUAGE table. The party must be created before you can create its language. You can create multiple language records for a party with different language names. The API allows to mark only one language record as primary language and only one language record as native language for a given party.

### PL/SQL Procedure

```
PROCEDURE create_person_language(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_person_language_rec    IN          PERSON_LANGUAGE_REC_TYPE,
    x_language_use_reference_id OUT      NUMBER,
    x_return_status         OUT      VARCHAR2,
    x_msg_count             OUT      NUMBER,
    x_msg_data              OUT      VARCHAR2
)
```

### Java Method

```
public static void createPersonLanguage(
    OracleConnection_connection,
    String                p_init_msg_list,
    PersonLanguageRec     p_person_language_rec,
    BigDecimal [ ]       x_language_use_reference_id,
    String [ ]           x_return_status,
    BigDecimal [ ]       x_msg_count,
    String [ ]           x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Person Language API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
language_use_reference_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
language_name	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to fnd_languages.language_code</li> </ul>
party_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to hz_parties.party_id</li> </ul>
native_language	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
primary_language_indicator	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
reads_level	IN	VARCHAR2	No	
speaks_level	IN	VARCHAR2	No	
writes_level	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
x_language_use_reference_id	OUT	NUMBER		Comment: Return language_use_reference_id of the record created

**Other Validations**

- There can be only one record for a given party and language.
- A party can have only one native language.
- A party can have only one primary language.

## Update Person Language API

### Description

This routine is used to update Language for a party. The API updates a record in the HZ\_PERSON\_LANGUAGE table. You cannot update the language name, but you can change other attributes of the language record.

### PL/SQL Procedure

```
PROCEDURE update_person_language(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_person_language_rec   IN          PERSON_LANGUAGE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

### Java Method

```
public static void updatePersonLanguage(
    OracleConnection_connection,
    String                p_init_msg_list,
    PersonLanguageRec     p_person_language_rec,
    BigDecimal [ ]       p_object_version_number,
    String [ ]           x_return_status,
    BigDecimal [ ]       x_msg_count,
    String [ ]           x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Person Language API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
language_use_reference_id	IN	NUMBER	Yes	Validation: Valid language_use_reference_id should be passed in  Comment: Pass the language_use_reference_id from hz_person_language table
language_name	IN	VARCHAR2	No	Validation: Not updateable

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	No	Validation: Not updateable
native_language	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
primary_language_indicator	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
reads_level	IN	VARCHAR2	No	
speaks_level	IN	VARCHAR2	No	
writes_level	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS</li> <li>■ Cannot be set to null during update</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Non updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing person language record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_PERSON_LANGUAGE.</li> <li>■ Return new value after update.</li> </ul>

**Other Validations**

- There can be only one record for a given party and language.
- A party can have only one native language.
- A party can have only one primary language.

## Location APIs

**PL/SQL Package Name: HZ\_LOCATION\_V2PUB**

**Java Class Name: HzLocationV2Pub**

### PL/SQL Constant:

```
G_MISS_CONTENT_COURSE_TYPE
CONSTANT VARCHAR2(30) := USER_ENTERED;
HZ_GEOMETRY_DEFAULT
CONSTANT MDSYS.SDO_GEOMETRY :=
MDSYS.SDO_GEOMETRY(FND_API.G_MISS_NUM,
FND_API.G_MISS_NUM, NULL, NULL, NULL)
```

### PL/SQL Record Structure for Location

```
TYPE location_rec_type IS RECORD(
    location_id                NUMBER,
    orig_system_reference      VARCHAR2(240),
    country                    VARCHAR2(60),
    address1                   VARCHAR2(240),
    address2                   VARCHAR2(240),
    address3                   VARCHAR2(240),
    address4                   VARCHAR2(240),
    city                       VARCHAR2(60),
    postal_code                VARCHAR2(60),
    state                      VARCHAR2(60),
    province                   VARCHAR2(60),
    county                     VARCHAR2(60),
    address_key                 VARCHAR2(500),
    address_style              VARCHAR2(30),
    validated_flag             VARCHAR2(1),
    address_lines_phonetic     VARCHAR2(560),
    po_box_number              VARCHAR2(50),
    house_number               VARCHAR2(50),
    street_suffix              VARCHAR2(50),
    street                     VARCHAR2(50),
    street_number              VARCHAR2(50),
    floor                      VARCHAR2(50),
    suite                      VARCHAR2(50),
    postal_plus4_code          VARCHAR2(10),
    position                   VARCHAR2(50),
    location_directions        VARCHAR2(640),
```

```

address_effective_date          DATE,
address_expiration_date        DATE,
cli_code                       VARCHAR2(60),
language                       VARCHAR2(4),
short_description              VARCHAR2(240),
description                    VARCHAR2(2000),
geometry                      MDSYS.SDO_GEOMETRY:= hz_geometry_default,
loc_hierarchy_id               NUMBER,
sales_tax_geocode              VARCHAR2(30),
sales_tax_inside_city_limits   VARCHAR2(30),
fa_location_id                 NUMBER,
content_source_type            VARCHAR2(30):= G_MISS_CONTENT_SOURCE_TYPE,
attribute_category             VARCHAR2(30),
attribute1                     VARCHAR2(150),
attribute2                     VARCHAR2(150),
attribute3                     VARCHAR2(150),
attribute4                     VARCHAR2(150),
attribute5                     VARCHAR2(150),
attribute6                     VARCHAR2(150),
attribute7                     VARCHAR2(150),
attribute8                     VARCHAR2(150),
attribute9                     VARCHAR2(150),
attribute10                    VARCHAR2(150),
attribute11                    VARCHAR2(150),
attribute12                    VARCHAR2(150),
attribute13                    VARCHAR2(150),
attribute14                    VARCHAR2(150),
attribute15                    VARCHAR2(150),
attribute16                    VARCHAR2(150),
attribute17                    VARCHAR2(150),
attribute18                    VARCHAR2(150),
attribute19                    VARCHAR2(150),
attribute20                    VARCHAR2(150),
timezone_id                   NUMBER,
created_by_module              VARCHAR2(150),
application_id                 NUMBER
)

```

### Java Inner Class for Location

```

public static class LocationRec {
    public BigDecimal          location_id;
    public String              orig_system_reference;
    public String              country;
    public String              address1;
    public String              address2;
}

```

```
public String address3;
public String address4;
public String city;
public String postal_code;
public String state;
public String province;
public String county;
public String address_key;
public String address_style;
public String validated_flag;
public String address_lines_phonetic;
public String po_box_number;
public String house_number;
public String street_suffix;
public String street;
public String street_number;
public String floor;
public String suite;
public String postal_plus4_code;
public String position;
public String location_directions;
public java.sql.Timestamp address_effective_date;
public java.sql.Timestamp address_expiration_date;
public String clli_code;
public String language;
public String short_description;
public String description;
public BigDecimal loc_hierarchy_id;
public String sales_tax_geocode;
public String sales_tax_inside_city_limits;
public BigDecimal fa_location_id;
public String content_source_type;
public String attribute_category;
public String attribute1;
public String attribute2;
public String attribute3;
public String attribute4;
public String attribute5;
public String attribute6;
public String attribute7;
public String attribute8;
public String attribute9;
public String attribute10;
public String attribute11;
public String attribute12;
```

```

        public String          attribute13;
        public String          attribute14;
        public String          attribute15;
        public String          attribute16;
        public String          attribute17;
        public String          attribute18;
        public String          attribute19;
        public String          attribute20;
        public BigDecimal      timezone_id;
        public String          actual_content_source;
        public String          created_by_module;
        public BigDecimal      application_id;

        public LocationRec();
        public LocationRec(boolean __RosettaUseGMISSValues);
    }

```

---



---

**Note:** Java Wrapper/API currently does not support the geometry column in HZ\_LOCATIONS table.

---



---

## Create Location API

---



---

**Note:** Java Wrapper/API currently does not support the geometry column in HZ\_LOCATIONS table.

---



---

### Description

This routine is used to create an Address Location. The API creates a record in the HZ\_LOCATIONS table. The location created by this API is just a physical location and can be used to create party site or customer account site.

### PL/SQL Procedure

```

PROCEDURE create_location (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_location_rec          IN          LOCATION_REC_TYPE,
    x_location_id           OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

**Java Method**

```

public static void createLocation(
    OracleConnection_connection,
    String p_init_msg_list,
    LocationRec                p_location_rec,
    BigDecimal [ ]             x_location_id,
    String [ ]                  x_return_status,
    BigDecimal [ ]             x_msg_count,
    String [ ]                  x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Create Location API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
location_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
orig_system_reference	IN	VARCHAR2	No	Default: location_id
country	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory Attribute</li> <li>■ Foreign Key to fnd_territories.territory_code</li> </ul>
address1	IN	VARCHAR2	Yes	Validation: Mandatory attribute
address2	IN	VARCHAR2	No	
address3	IN	VARCHAR2	No	
address4	IN	VARCHAR2	No	
city	IN	VARCHAR2	No	
postal_code	IN	VARCHAR2	No	
state	IN	VARCHAR2	No	
province	IN	VARCHAR2	No	
county	IN	VARCHAR2	No	
address_key	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
address_style	IN	VARCHAR2	No	
validated_flag	IN	VARCHAR2	No	
address_lines_phonetic	IN	VARCHAR2	No	
po_box_number	IN	VARCHAR2	No	
house_number	IN	VARCHAR2	No	
street_suffix	IN	VARCHAR2	No	
street	IN	VARCHAR2	No	
street_number	IN	VARCHAR2	No	
floor	IN	VARCHAR2	No	
suite	IN	VARCHAR2	No	
postal_plus4_code	IN	VARCHAR2	No	
position	IN	VARCHAR2	No	
location_directions	IN	VARCHAR2	No	
address_effective_date	IN	DATE	No	
address_expiration_date	IN	DATE	No	
cli_code	IN	VARCHAR2	No	
language	IN	VARCHAR2	No	Validation: Foreign key to fnd_languages.language_code (installed)
short_description	IN	VARCHAR2	No	
description	IN	VARCHAR2	No	
geometry	IN	OBJECT	No	
loc_hierarchy_id	IN	NUMBER	No	
sales_tax_geocode	IN	VARCHAR2	No	
sales_tax_inside_city_limits	IN	VARCHAR2	No	
fa_location_id	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE  Default: USER_ENTERED
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezones.timezone_id

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
actual_content_source	IN	VARCHAR2	No	Validation : Validated against AR lookup type CONTENT_SOURCE_TYPE Default : 'USER_ENTERED'
x_location_id	OUT	NUMBER	No	Comment; Return location_id of the record created

## Update Location API

### Description

This routine is used to update an Address Location. The API updates a record in the HZ\_LOCATIONS table.

### PL/SQL Procedure

```
PROCEDURE update_location (
    p_init_msg_list          IN          VARCHAR2:=FND_API.G_FALSE,
    p_location_rec          IN          LOCATION_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

### Java Method

```
public static void updateLocation(
    OracleConnection_connection,
    String p_init_msg_list,
    LocationRec p_location_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Location API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
location_id	IN	NUMBER	Yes	Validation: Valid location_id should be passed in.
orig_system_reference	IN	VARCHAR2	No	
country	IN	VARCHAR2	No	Validation: Foreign key to fnd_territories.territory_code
address1	IN	VARCHAR2	No	Validation: Cannot be set to null during update
address2	IN	VARCHAR2	No	
address3	IN	VARCHAR2	No	
address4	IN	VARCHAR2	No	
city	IN	VARCHAR2	No	
postal_code	IN	VARCHAR2	No	
state	IN	VARCHAR2	No	
province	IN	VARCHAR2	No	
county	IN	VARCHAR2	No	
address_key	IN	VARCHAR2	No	
address_style	IN	VARCHAR2	No	
validated_flag	IN	VARCHAR2	No	
address_lines_phonetic	IN	VARCHAR2	No	
po_box_number	IN	VARCHAR2	No	
house_number	IN	VARCHAR2	No	
street_suffix	IN	VARCHAR2	No	
street	IN	VARCHAR2	No	
street_number	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
floor	IN	VARCHAR2	No	
suite	IN	VARCHAR2	No	
postal_plus4_code	IN	VARCHAR2	No	
position	IN	VARCHAR2	No	
location_directions	IN	VARCHAR2	No	
address_effective_date	IN	DATE	No	
address_expiration_date	IN	DATE	No	
cli_code	IN	VARCHAR2	No	
language	IN	VARCHAR2	No	Validation: Foreign key to fnd_languages.language_code (installed)
short_description	IN	VARCHAR2	No	
description	IN	VARCHAR2	No	
geometry	IN	OBJECT	No	
loc_hierarchy_id	IN	NUMBER	No	
sales_tax_geocode	IN	VARCHAR2	No	
sales_tax_inside_city_limits	IN	VARCHAR2	No	
fa_location_id	IN	NUMBER	No	
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Not updateable
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezones.timezone_id
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_source	IN	VARCHAR2	No	Validation: Cannot be updated.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN/O UT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>■ Mandatory attribute</li><li>■ Validated against value in the database for the existing person location record</li></ul> Comment: <ul style="list-style-type: none"><li>■ Pass the current object_version_number of the location record</li><li>■ Return new value after update.</li></ul>

## Party Site APIs

**PL/SQL Package Name: HZ\_PARTY\_SITE\_V2PUB**

**Java Class Name: HzPartySiteV2Pub**

### PL/SQL Record Structure for Party Site

```

TYPE party_site_rec_type          IS RECORD(
  party_site_id                   NUMBER,
  party_id                        NUMBER,
  location_id                     NUMBER,
  party_site_number               VARCHAR2(30),
  orig_system_reference           VARCHAR2(240),
  mailstop                       VARCHAR2(60),
  identifying_address_flag       VARCHAR2(1),
  status                          VARCHAR2(1),
  party_site_name                 VARCHAR2(240),
  attribute_category              VARCHAR2(30),
  attribute1                      VARCHAR2(150),
  attribute2                      VARCHAR2(150),
  attribute3                      VARCHAR2(150),
  attribute4                      VARCHAR2(150),
  attribute5                      VARCHAR2(150),
  attribute6                      VARCHAR2(150),
  attribute7                      VARCHAR2(150),
  attribute8                      VARCHAR2(150),
  attribute9                      VARCHAR2(150),
  attribute10                    VARCHAR2(150),
  attribute11                    VARCHAR2(150),
  attribute12                    VARCHAR2(150),
  attribute13                    VARCHAR2(150),
  attribute14                    VARCHAR2(150),
  attribute15                    VARCHAR2(150),
  attribute16                    VARCHAR2(150),
  attribute17                    VARCHAR2(150),
  attribute18                    VARCHAR2(150),
  attribute19                    VARCHAR2(150),
  attribute20                    VARCHAR2(150),
  language                       VARCHAR2(4),
  addressee                      VARCHAR2(150),
  created_by_module              VARCHAR2(150),
  application_id                 NUMBER
);

```



```

        public String          attribute18;
        public String          attribute19;
        public String          attribute20;
        public String          language;
        public String          addressee;
        public String          created_by_module;
        public BigDecimal      application_id;

        public PartySiteRec();
        public PartySiteRec(boolean __RosettaUseGMISSValues);
    }

```

## Java Inner Class for Party Site Use

```

public static class PartySiteUseRec {
    public BigDecimal          party_site_use_id;
    public String              comments;
    public String              site_use_type;
    public BigDecimal          party_site_id;
    public String              primary_per_type;
    public String              status;
    public String              created_by_module;
    public BigDecimal          application_id;

    public PartySiteUseRec();
    public PartySiteUseRec(boolean __RosettaUseGMISSValues);
}

```

## Create Party Site API

### Description

This routine is used to create a Party Site for a party. Party Site relates an existing party from the HZ\_PARTIES table with an address location from the HZ\_LOCATIONS table. The API creates a record in the HZ\_PARTY\_SITES table. You can create multiple party sites with multiple locations and mark one of those party sites as identifying for that party. The identifying party site address components are denormalized into the HZ\_PARTIES table.

### PL/SQL Procedure

```

PROCEDURE create_party_site (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_party_site_rec         IN          PARTY_SITE_REC_TYPE,
    x_party_site_id         OUT         NUMBER,
    x_party_site_number     OUT         VARCHAR2,

```

```

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

```

### Java Method

```

public static void createPartySite(
    OracleConnection_connection,
    String p_init_msg_list,
    PartySiteRec p_party_site_rec,
    BigDecimal [ ] x_party_site_id,
    String [ ] x_party_site_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Create party Site API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_site_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
party_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign Key to hz_parties.party_id</li> </ul>
location_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign Key to hz_locations.location_id</li> </ul>
party_site_number	IN	VARCHAR2	Yes/No	Validation: <ul style="list-style-type: none"> <li>▪ Required when profile HZ_GENERATE_PARTY_SITE_NUMBER = N, else generated from sequence.</li> <li>▪ Unique when passed in</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
orig_system_reference	IN	VARCHAR2	No	Default: party_site_id
mailstop	IN	VARCHAR2	No	
identifying_address_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY STATUS
party_site_name	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
language	IN	VARCHAR2	No	Validation: Foreign Key to fnd_languages.language_code (installed)
addressee	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
x_party_site_id	IN	NUMBER	No	Comment: party_site_id of the party site record created
x_party_site_number	IN	NUMBER	No	Comment: party_site_number of the party site record created

### Other Validations

A party can have only one location defined as the Identifying Address.

The API automatically sets the `identifying_address_flag` to Y for the first active, visible party site created for a given party, even if you pass N for the attribute value. You cannot change the `identifying_address_flag` from Y to N. Instead, you must choose a different active, visible party site and set that party site as the identifying address. Choosing a new identifying address causes the status of the old identifying address to change to non-identifying.

Moreover, when you activate a party site, if this is the first active, visible party site for a party, the API sets this party site as the identifying address. If you inactivate a party site that is the identifying address, the API changes its status to non-identifying address, finds the first active, visible party site from the existing party sites, and then makes that one as identifying address. If the API cannot find any active, visible party site, the party has no identifying address. You cannot set an inactive party site as the identifying address.

The API denormalizes location components such as `address1`, `city`, and `country` only from the identifying party sites to `HZ_PARTIES`.

## Update Party Site API

### Description

This routine is used to update a Party Site. The API updates a record in the HZ\_PARTY\_SITES table. You cannot set the identifying address flag to 'N' to unmark the party site as identifying, rather you should set another site as identifying which makes any other party site for that party as non identifying. The identifying party site address components are denormalized into the HZ\_PARTIES table.

### PL/SQL Procedure

```
PROCEDURE update_party_site (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_party_site_rec        IN          PARTY_SITE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

### Java Method

```
public static void updatePartySite(
    OracleConnection connection,
    String p_init_msg_list,
    PartySiteRec p_party_site_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update party Site API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_site_id	IN	NUMBER	Yes	Validation: Valid party_site_id from HZ_PARTY_SITE_USES table  Comment: Pass the party_site_id from HZ_PARTY_SITES table

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	No	Validation: Not updateable
location_id	IN	NUMBER	No	Validation: Not updateable
party_site_number	IN	VARCHAR2	No	Validation: Not updateable
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
mailstop	IN	VARCHAR2	No	
identifying_address_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY STATUS</li> <li>■ Cannot be set to null during update</li> </ul>
party_site_name	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
language	IN	VARCHAR2	No	Validation: Foreign key to fnd_languages.language_code (installed)
addressee	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from HZ_PARTY_SITES</li> <li>■ Return new value after update.</li> </ul>

#### Other Validations

A party can have only one location defined as the Identifying Party Site.

## Create Party Site Use API

### Description

This routine is used to create a Party Site Use for a Party Site. The API creates a record in the HZ\_PARTY\_SITE\_USES table. Party site use defines a business

purpose for a party site such as 'BILL\_TO', 'SHIP\_TO' etc. You can create party site use for a party site that is already present in the HZ\_PARTY\_SITES table.

### PL/SQL Procedure

```
PROCEDURE create_party_site_use (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_party_site_use_rec    IN          PARTY_SITE_USE_REC_TYPE,
    x_party_site_use_id     OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

### Java Method

```
public static void createPartySiteUse(
    OracleConnection_connection,
    String                    p_init_msg_list,
    PartySiteUseRec          p_party_site_use_rec,
    BigDecimal [ ]           x_party_site_use_id,
    String [ ]               x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Party Site Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_site_use_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
comments	IN	VARCHAR2	No	
site_use_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against AR lookup type PARTY_SITE_USE_CODE</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_site_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign key to hz_party_sites.party_site_id</li> </ul>
primary_per_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
x_party_site_use_id	OUT	NUMBER	No	Comment: Return Party Site Use ID of the party site use created

### Other Validations

The combination of party\_site\_id and site\_use\_type must be unique.

## Update Party Site Use API

### Description

This routine is used to update a Party Site Use. The API updates a record in the HZ\_PARTY\_SITE\_USES table.

### PL/SQL Procedure

```
PROCEDURE update_party_site_use (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_party_site_use_rec     IN          PARTY_SITE_USE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

**Java Method**

```

public static void updatePartySiteUse(
    OracleConnection_connection,
    String                                p_init_msg_list,
    PartySiteUseRec                       p_party_site_use_rec,
    BigDecimal [ ]                        p_object_version_number,
    String [ ]                             x_return_status,
    BigDecimal [ ]                         x_msg_count,
    String [ ]                             x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Update Party Site Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_site_use_id	IN	NUMBER	Yes	Validation: Valid party_site_use_id from HZ_PARTY_SITE_USES table  Comment: Pass the party_site_use_id from HZ_PARTY_SITE_USES record
comments	IN	VARCHAR2	No	
site_use_type	IN	VARCHAR2	No	Validation: Not updateable
party_site_id	IN	NUMBER	No	Validation: Not updateable
primary_per_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS</li> <li>■ Cannot be updated to null</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

---

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>■ Mandatory attribute</li><li>■ Validated against value in the database for the existing record</li></ul> Comment: <ul style="list-style-type: none"><li>■ Pass the current object_version_number of the record from hz_party_site_uses</li><li>■ Return new value after update</li></ul>

---



---

---

## Contact Point API Use

This chapter provides information about the Contact Point application programming interface.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Contact Point APIs

**PL/SQL Package Name: HZ\_CONTACT\_POINT\_V2PUB**

**Java Class Name: HzContactPointV2Pub**

**PL/SQL Constant:**

```
G_MISS_CONTENT_SOURCE_TYPE  
(CONSTANT VARCHAR2(30) := 'USER_ENTERED');
```

**PL/SQL Record Structure for Contact Point**

```
TYPE contact_point_rec_type          IS RECORD (  
    contact_point_id                 NUMBER,  
    contact_point_type               VARCHAR2(30),  
    status                           VARCHAR2(30),  
    owner_table_name                 VARCHAR2(30),  
    owner_table_id                   NUMBER,  
    primary_flag                     VARCHAR2(1),  
    orig_system_reference            VARCHAR2(240),  
    content_source_type              VARCHAR2(30) := G_MISS_CONTENT_SOURCE_TYPE,  
    attribute_category               VARCHAR2(30),  
    attribute1                       VARCHAR2(150),  
    attribute2                       VARCHAR2(150),  
    attribute3                       VARCHAR2(150),  
    attribute4                       VARCHAR2(150),  
    attribute5                       VARCHAR2(150),  
    attribute6                       VARCHAR2(150),  
    attribute7                       VARCHAR2(150),  
    attribute8                       VARCHAR2(150),  
    attribute9                       VARCHAR2(150),  
    attribute10                      VARCHAR2(150),  
    attribute11                      VARCHAR2(150),  
    attribute12                      VARCHAR2(150),  
    attribute13                      VARCHAR2(150),  
    attribute14                      VARCHAR2(150),  
    attribute15                      VARCHAR2(150),  
    attribute16                      VARCHAR2(150),  
    attribute17                      VARCHAR2(150),  
    attribute18                      VARCHAR2(150),  
    attribute19                      VARCHAR2(150),  
    attribute20                      VARCHAR2(150),  
    contact_point_purpose              VARCHAR2(30),
```

```

        primary_by_purpose          VARCHAR2(1),
        created_by_module        VARCHAR2(150),
        application_id           NUMBER
        actual_content_source     VARCHAR2(30)
    )

```

### PL/SQL Record Structure for EDI

```

TYPE edi_rec_type IS RECORD(
    edi_transaction_handlin      VARCHAR2(25),
    edi_id_number                VARCHAR2(30),
    edi_payment_method           VARCHAR2(30),
    edi_payment_format           VARCHAR2(30),
    edi_remittance_method        VARCHAR2(30),
    edi_remittance_instruction   VARCHAR2(30),
    edi_tp_header_id             NUMBER,
    edi_ece_tp_location_code     VARCHAR2(40)
)

```

### PL/SQL Record Structure for EMAIL

```

TYPE email_rec_type IS RECORD (
    email_format                 VARCHAR2(30),
    email_address                VARCHAR2(2000)
)

```

### PL/SQL Record Structure for PHONE

```

TYPE phone_rec_type           IS RECORD (
    phone_calling_calendar     VARCHAR2(30),
    last_contact_dt_time       DATE,
    timezone_id                NUMBER,
    phone_area_code            VARCHAR2(10),
    phone_country_code         VARCHAR2(10),
    phone_number                VARCHAR2(40),
    phone_extension            VARCHAR2(20),
    phone_line_type            VARCHAR2(30),
    raw_phone_number           VARCHAR2(60)
)

```

### PL/SQL Record Structure for TELEX

```

TYPE telex_rec_type           IS RECORD (
    telex_number                VARCHAR2(50)
)

```

## PL/SQL Record Structure for WEB

```
TYPE web_rec_type IS RECORD (  
    web_type          VARCHAR2(60),  
    url              VARCHAR2(2000)  
)
```

## PL/SQL Record Structure for EFT

```
TYPE eft_rec_type          IS RECORD (  
    eft_transmission_program_id NUMBER,  
    eft_printing_program_id  NUMBER,  
    eft_user_number         VARCHAR2(30),  
    eft_swift_code          VARCHAR2(30),  
)
```

## Java Inner Class for Contact Point

```
public static class ContactPointRec {  
    public BigDecimal    contact_point_id;  
    public String       contact_point_type;  
    public String       status;  
    public String       owner_table_name;  
    public BigDecimal    owner_table_id;  
    public String       primary_flag;  
    public String       orig_system_reference;  
    public String       content_source_type;  
    public String       attribute_category;  
    public String       attribute1;  
    public String       attribute2;  
    public String       attribute3;  
    public String       attribute4;  
    public String       attribute5;  
    public String       attribute6;  
    public String       attribute7;  
    public String       attribute8;  
    public String       attribute9;  
    public String       attribute10;  
    public String       attribute11;  
    public String       attribute12;  
    public String       attribute13;  
    public String       attribute14;  
    public String       attribute15;  
    public String       attribute16;  
    public String       attribute17;  
    public String       attribute18;  
}
```

```

        public String                attribute19;
        public String                attribute20;
        public String                contact_point_purpose;
        public String                primary_by_purpose;
        public String                created_by_module;
        public BigDecimal            application_id;
        public String                actual_content_source;

        public ContactPointRec();
        public ContactPointRec(boolean__RosettaUseGMISSValues);
    }

```

### Java Inner Class for EDI

```

public static class EdiRec {
    public String                edi_transaction_handling;
    public String                edi_id_number;
    public String                edi_payment_method;
    public String                edi_payment_format;
    public String                edi_remittance_method;
    public String                edi_remittance_instruction;
    public BigDecimal            edi_tp_header_id;
    public String                edi_ece_tp_location_code;

    public EdiRec();
    public EdiRec(boolean__RosettaUseGMISSValues);
}

```

### Java Inner Class for EMAIL

```

public static class EmailRec {
    public String                email_format;
    public String                email_address;

    public EmailRec();
    public EmailRec(boolean __RosettaUseGMISSValues);
}

```

### Java Inner Class for PHONE

```

public static class PhoneRec {
    public String                phone_calling_calendar;
    public java.sql.Timestamp    last_contact_dt_time;
    public BigDecimal            timezone_id;
    public String                phone_area_code;
}

```

```
        public String                phone_country_code;
        public String                phone_number;
        public String                phone_extension;
        public String                phone_line_type;
        public String                raw_phone_number;

        public PhoneRec();
        public PhoneRec(boolean __RosettaUseGMISSValues);
    }
```

### Java Inner Class for TELEX

```
    public static class TelexRec {
        public String                telex_number;

        public TelexRec();
        public TelexRec(boolean __RosettaUseGMISSValues);
    }
```

### Java Inner Class for WEB

```
    public static class WebRec {
        public String                web_type;
        public String                url;

        public WebRec();
        public WebRec(boolean __RosettaUseGMISSValues);
    }
```

### Java Inner Class for EFT

```
    public static class EftRec {
        public BigDecimal            eft_transmission_program_id;
        public BigDecimal            eft_printing_program_id;
        public String                eft_user_number
        public String                eft_swift_code

        public EftRec() {this(true);}
        public EftRec(boolean __RosettaUseGMISSValues);
    }
```

## Create Contact Point API

### Description

Use this routine to create a Contact Point for a Party or a Party Site. The supported types of contact points are PHONE, PAGER, EMAIL, TELEX, WEB, EFT, and EDI. This routine creates a record in the HZ\_CONTACT\_POINTS table. Each contact point type has a corresponding API. You must call the relevant interface and pass the corresponding record, which depends on the type of contact point you create.

You should use the contact type-dependent APIs. A generic API, called Create Contact Point, is available but does not handle EFT contact points or any future contact point types. The generic Create Contact Point API requires that you pass the appropriate record along with the proper contact point type for the contact point that you create.

### PL/SQL Procedure for EDI Contact Points:

```
PROCEDURE create_edi_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_edi_rec               IN          EDI_REC_TYPE:=G_MISS_EDI_REC,
    x_contact_point_id     OUT         NUMBER,
    x_return_status        OUT         VARCHAR2,
    x_msg_count            OUT         NUMBER,
    x_msg_data             OUT         VARCHAR2
)
```

### PL/SQL Procedure for EMAIL Contact Points:

```
PROCEDURE create_email_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_email_rec             IN          EMAIL_REC_TYPE:=G_MISS_EMAIL_REC,
    x_contact_point_id     OUT         NUMBER,
    x_return_status        OUT         VARCHAR2,
    x_msg_count            OUT         NUMBER,
    x_msg_data             OUT         VARCHAR2
)
```

### PL/SQL Procedure for PHONE Contact Points:

```
PROCEDURE create_phone_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_phone_rec            IN          PHONE_REC_TYPE:=G_MISS_PHONE_REC,
    x_contact_point_id     OUT         NUMBER,
    x_return_status        OUT         VARCHAR2,
```

```

        x_msg_count          OUT          NUMBER,
        x_msg_data          OUT          VARCHAR2
    )

```

**PL/SQL Procedure for WEB Contact Points:**

```

PROCEDURE create_web_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_web_rec                IN          WEB_REC_TYPE:=G_MISS_WEB_REC,
    x_contact_point_id      OUT          NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)

```

**PL/SQL Procedure for TELEX Contact Points:**

```

PROCEDURE create_telex_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_telex_rec              IN          TELEX_REC_TYPE:=G_MISS_TELEX_REC,
    x_contact_point_id      OUT          NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)

```

**PL/SQL Procedure for EFT Contact Points:**

```

PROCEDURE create_eft_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_eft_rec                IN          EFT_REC_TYPE:=G_MISS_EFT_REC,
    x_contact_point_id      OUT          NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)

```

**Generic PL/SQL Procedure**

```

PROCEDURE create_contact_point (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_edi_rec                IN          EDI_REC_TYPE:= G_MISS_EDI_REC,
    p_email_rec              IN          EMAIL_REC_TYPE:= G_MISS_EMAIL_REC,
    p_phone_rec              IN          PHONE_REC_TYPE:= G_MISS_PHONE_REC,
    p_telex_rec              IN          TELEX_REC_TYPE:= G_MISS_TELEX_REC,
)

```

```

    p_web_rec                IN          WEB_REC_TYPE:= G_MISS_WEB_REC,
    x_contact_point_id       OUT          NUMBER,
    x_return_status          OUT          VARCHAR2,
    x_msg_count              OUT          NUMBER,
    x_msg_data               OUT          VARCHAR2
)

```

### Java Method for EDI Contact Points

```

public static void createEdiContactPoint(
    OracleConnection_connection,
    String
    ContactPointRec          p_init_msg_list,
    EdiRec                   p_contact_point_rec,
    BigDecimal [ ]           p_edi_rec,
    String [ ]               x_contact_point_id,
    BigDecimal [ ]           x_return_status,
    String [ ]               x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;

```

### Java Method for EMAIL Contact Points

```

public static void createEmailContactPoint(
    OracleConnection_connection,
    String
    ContactPointRec          p_init_msg_list,
    EmailRec                 p_contact_point_rec,
    BigDecimal [ ]           p_email_rec,
    String [ ]               x_contact_point_id,
    BigDecimal [ ]           x_return_status,
    String [ ]               x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;

```

### Java Method for PHONE Contact Points

```

public static void createPhoneContactPoint(
    OracleConnection_connection,
    String
    ContactPointRec          p_init_msg_list,
    PhoneRec                 p_contact_point_rec,
    BigDecimal [ ]           p_phone_rec,
    String [ ]               x_contact_point_id,
    BigDecimal [ ]           x_return_status,
    String [ ]               x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;

```

### Java Method for WEB Contact Points

```

public static void createWebContactPoint(

```

```

        OracleConnection_connection,
        String                                p_init_msg_list,
        ContactPointRec                       p_contact_point_rec,
        WebRec                                p_web_rec,
        BigDecimal [ ]                        x_contact_point_id,
        String [ ]                            x_return_status,
        BigDecimal [ ]                        x_msg_count,
        String [ ]                            x_msg_data
    ) throws SQLException;

```

### Java Method for TELEX Contact Points

```

public static void createTelexContactPoint(
    OracleConnection_connection,
    String                                p_init_msg_list,
    ContactPointRec                       p_contact_point_rec,
    TelexRec                              p_telex_rec,
    BigDecimal [ ]                        x_contact_point_id,
    String [ ]                            x_return_status,
    BigDecimal [ ]                        x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;

```

### Java Method for EFT Contact Points

```

public static void createEftContactPoint(
    OracleConnection_connection,
    String                                p_init_msg_list,
    ContactPointRec                       p_contact_point_rec,
    EftRec                                p_eft_rec,
    BigDecimal [ ]                        x_contact_point_id,
    String [ ]                            x_return_status,
    BigDecimal [ ]                        x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;

```

### Generic Java Method

```

public static void createContactPoint(
    OracleConnection_connection,
    String                                p_init_msg_list,
    ContactPointRec                       p_contact_point_rec,
    EdiRec                                p_edi_rec,
    EmailRec                              p_email_rec,
    PhoneRec                              p_phone_rec,
    TelexRec                              p_telex_rec,
    WebRec                                p_web_rec,
    BigDecimal [ ]                        x_contact_point_id,
)

```

```

    String [ ]                x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Contact Point API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
contact_point_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
contact_point_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against AR lookup type COMMUNICATION_TYPE</li> </ul>
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
owner_table_name	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against AR lookup type OWNER_TABLE_NAME</li> </ul> <p>Comment: If you are creating contact point for a party, pass HZ_PARTIES, if you are creating contact point for a party site, pass HZ_PARTY_SITES</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
owner_table_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign Key hz_parties.party_id when owner_table_name = HZ_PARTIES.</li> <li>■ Foreign Key to hz_party_sites.party_site_id when owner_table_name = HZ_PARTY_SITES</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Default: N
orig_system_reference	IN	VARCHAR2	No	Default: contact_point_id
content_source_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE Default: USER_ENTERED Comment: This parameter is no longer used. Use actual_content_source.
Attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
contact_point_purpose	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB</li> <li>■ Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type=WEB</li> </ul>
primary_by_purpose	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
actual_content_source	IN	VARCHAR2	No	Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE Default: USER_ENTERED

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_edi_rec record type				
If you use the generic validation method, validations only apply when contact_point type = EDI.				
edi_transaction_handling	IN	VARCHAR2	No	
edi_id_number	IN	VARCHAR2	Yes	Validation: Mandatory attribute
edi_payment_method	IN	VARCHAR2	No	
edi_payment_format	IN	VARCHAR2	No	
edi_remittance_method	IN	VARCHAR2	No	
edi_remittance_instruction	IN	VARCHAR2	No	
edi_tp_header_id	IN	NUMBER	No	
edi_ece_tp_location_code	IN	VARCHAR2	No	
p_email_rec record type				
If you use the generic validation method, validations only apply when contact_point_type=EMAIL.				
email_format	IN	VARCHAR2	No	Validation: Validated against AR lookup type EMAIL_FORMAT Default: MAILHTML
email_address	IN	VARCHAR2	Yes	Validation: Mandatory attribute
p_phone_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = PHONE or PAGER.				
phone_calling_calendar	IN	VARCHAR2	No	
last_contact_dt_time	IN	DATE	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id
phone_area_code	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
phone_country_code	IN	VARCHAR2	No	Validation: Foreign key to hz_phone_country_codes.phone_country_code
Phone_number	IN	VARCHAR2	Yes/No	Validation: Mandatory if raw_phone_number is not passed in
phone_extension	IN	VARCHAR2	No	
phone_line_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type PHONE_LINE_TYPE</li> </ul>
raw_phone_number	IN	VARCHAR2	Yes/No	Validation: Mandatory if phone_number is not passed in
p_telex_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = TELEX.				
telex_number	IN	VARCHAR2	Yes	Validation: Mandatory attribute
p_web_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = WEB.				
web_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute
url	IN	VARCHAR2	Yes	Validation: Mandatory attribute
p_eft_rec_record_type				
If you use the generic validation method, validations only apply when contact_point_type = EFT.				
eft_transmission_program_id	IN	NUMBER	No	
eft_printing_program_id	IN	NUMBER	No	
eft_user_number	IN	VARCHAR2	No	
eft_swift_code	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_contact_point_id	OUT	NUMBER	No	Comment: Return contact_point_id of the contact point record created

### Other Validations

- The Primary\_flag is a lookup code of lookup type YES/NO. The API automatically marks the first active, visible contact point per type for an entity to primary. When the user selects another contact point of same type to be primary, the previous primary contact point will be unset. Also, if the user inactivates the primary contact point, this primary contact point will be unset and next available active, visible contact point with same type will become primary; and if the user activates a contact point, and if this contact point becomes the first active, visible contact point of this type for this entity, the contact point will be marked as primary. Primary flag is defaulted to 'N' if none of the above scenarios occur and the user does not pass any value for primary flag.
- An inactive contact can never be marked as primary.
- Only the only the primary URL and email for given parties are denormalized to HZ\_PARTIES table.
- primary\_by\_purpose is a lookup code of lookup type YES/NO. It is defaulted to 'N' if user does not pass a value. There is only one primary per purpose contact point exist for the combination of owner\_table\_name, owner\_table\_id, contact\_point\_type, and contact\_point\_purpose. If primary\_by\_purpose is set to 'Y', we need to unset the previous primary per purpose contact point to non-primary. Because setting primary\_by\_purpose is only making sense when contact\_point\_purpose has some value, we ignore the primary\_by\_purpose (setting it to 'N') if contact\_point\_purpose is null.

## Update Contact Point API

### Description

Use this routine to update a Contact Point for a Party or a Party Site. the various types of contact points supported are PHONE, PAGER, EMAIL, TELEX, WEB, EFT, and EDI. The API updates a record in the HZ\_CONTACT\_POINTS table. Each contact point type has a corresponding API. You must call the relevant interface and pass the corresponding record, which depends on the type of contact point you create.

You should use the contact type-dependent APIs. A generic API, called Update Contact Point, is available, but it does not handle EFT contact points or any future contact point types. The generic Update Contact Point API requires that you pass the appropriate record along with the proper contact point type for the contact point that you create.

#### PL/SQL Procedure for EDI Contact Points:

```
PROCEDURE update_edi_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_edi_rec               IN          EDI_REC_TYPE:=G_MISS_EDI_REC,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

#### PL/SQL Procedure for EMAIL Contact Points:

```
PROCEDURE update_email_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_email_rec             IN          EMAIL_REC_TYPE:=G_MISS_EMAIL_REC,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

#### PL/SQL Procedure for PHONE Contact Points:

```
PROCEDURE update_phone_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_phone_rec             IN          PHONE_REC_TYPE:=G_MISS_PHONE_REC,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

#### PL/SQL Procedure for WEB Contact Points:

```
PROCEDURE update_web_contact_point(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec     IN          CONTACT_POINT_REC_TYPE,
    p_web_rec               IN          WEB_REC_TYPE:=G_MISS_WEB_REC,
```

```

    p_object_version_number    IN OUT    NUMBER,
    x_return_status            OUT        VARCHAR2,
    x_msg_count                OUT        NUMBER,
    x_msg_data                 OUT        VARCHAR2
)

```

**PL/SQL Procedure for TELEX Contact Points:**

```

PROCEDURE update_telex_contact_point(
    p_init_msg_list            IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec        IN          CONTACT_POINT_REC_TYPE,
    p_telex_rec                IN          TELEX_REC_TYPE:=G_MISS_TELEX_REC,
    p_object_version_number    IN OUT     NUMBER,
    x_return_status            OUT        VARCHAR2,
    x_msg_count                OUT        NUMBER,
    x_msg_data                 OUT        VARCHAR2
)

```

**PL/SQL Procedure for EFT Contact Points:**

```

PROCEDURE update_eft_contact_point(
    p_init_msg_list            IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec        IN          CONTACT_POINT_REC_TYPE,
    p_eft_rec                  IN          EFT_REC_TYPE:=G_MISS_EFT_REC,
    p_object_version_number    IN OUT     NUMBER,
    x_return_status            OUT        VARCHAR2,
    x_msg_count                OUT        NUMBER,
    x_msg_data                 OUT        VARCHAR2
)

```

**Generic PL/SQL Procedure**

```

PROCEDURE update_contact_point (
    p_init_msg_list            IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_point_rec        IN          CONTACT_POINT_REC_TYPE,
    p_edi_rec                  IN          EDI_REC_TYPE:= G_MISS_EDI_REC,
    p_email_rec                IN          EMAIL_REC_TYPE:= G_MISS_EMAIL_REC,
    p_phone_rec                IN          PHONE_REC_TYPE:= G_MISS_PHONE_REC,
    p_telex_rec                IN          TELEX_REC_TYPE:= G_MISS_TELEX_REC,
    p_web_rec                  IN          WEB_REC_TYPE:= G_MISS_WEB_REC,
    p_object_version_number    IN OUT     NUMBER,
    x_return_status            OUT        VARCHAR2,
    x_msg_count                OUT        NUMBER,
    x_msg_data                 OUT        VARCHAR2
)

```

**Java Method for EDI Contact Points**

```

public static void updateEdiContactPoint(

```

```

        OracleConnection_connection,
        String                               p_init_msg_list,
        ContactPointRec                      p_contact_point_rec,
        EdiRec                               p_edi_rec,
        BigDecimal [ ]                      p_object_version_number,
        String [ ]                          x_return_status,
        BigDecimal [ ]                      x_msg_count,
        String [ ]                          x_msg_data
    ) throws SQLException;

```

### Java Method for EMAIL Contact Points

```

public static void updateEmailContactPoint(
    OracleConnection_connection,
    String                               p_init_msg_list,
    ContactPointRec                      p_contact_point_rec,
    EmailRec                             p_email_rec,
    BigDecimal [ ]                      p_object_version_number,
    String [ ]                          x_return_status,
    BigDecimal [ ]                      x_msg_count,
    String [ ]                          x_msg_data
) throws SQLException;

```

### Java Method for PHONE Contact Points

```

public static void updatePhoneContactPoint(
    OracleConnection_connection,
    String                               p_init_msg_list,
    ContactPointRec                      p_contact_point_rec,
    PhoneRec                             p_phone_rec,
    BigDecimal [ ]                      p_object_version_number,
    String [ ]                          x_return_status,
    BigDecimal [ ]                      x_msg_count,
    String [ ]                          x_msg_data
) throws SQLException;

```

### Java Method for WEB Contact Points

```

public static void updateWebContactPoint(
    OracleConnection_connection,
    String                               p_init_msg_list,
    ContactPointRec                      p_contact_point_rec,
    WebRec                               p_web_rec,
    BigDecimal [ ]                      p_object_version_number,
    String [ ]                          x_return_status,
    BigDecimal [ ]                      x_msg_count,
    String [ ]                          x_msg_data
) throws SQLException;

```

### Java Method for TELEX Contact Points

```
public static void updateTelexContactPoint(  
    OracleConnection_connection,  
    String                                     p_init_msg_list,  
    ContactPointRec                          p_contact_point_rec,  
    TelexRec                                 p_telex_rec,  
    BigDecimal [ ]                          p_object_version_number,  
    String [ ]                               x_return_status,  
    BigDecimal [ ]                          x_msg_count,  
    String [ ]                               x_msg_data  
) throws SQLException;
```

### Java Method for EFT Contact Points

```
public static void updateEftContactPoint(  
    OracleConnection_connection,  
    String                                     p_init_msg_list,  
    ContactPointRec                          p_contact_point_rec,  
    EftRec                                   p_eft_rec,  
    BigDecimal [ ]                          p_object_version_number,  
    String [ ]                               x_return_status,  
    BigDecimal [ ]                          x_msg_count,  
    String [ ]                               x_msg_data  
) throws SQLException;
```

### Generic Java Method

```
public static void updateContactPoint(  
    OracleConnection_connection,  
    String                                     p_init_msg_list,  
    ContactPointRec                          p_contact_point_rec,  
    EdiRec                                   p_edi_rec,  
    EmailRec                                 p_email_rec,  
    PhoneRec                                p_phone_rec,  
    TelexRec                                 p_telex_rec,  
    WebRec                                  p_web_rec,  
    BigDecimal [ ]                          p_object_version_number,  
    String [ ]                               x_return_status,  
    BigDecimal [ ]                          x_msg_count,  
    String [ ]                               x_msg_data  
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Contact Point API. The table includes the parameter names, the type of each parameter, the

data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_id	IN	NUMBER	Yes	Validation: Valid contact_point_id should be passed in  Comment: Pass the contact_point_id from hz_contact_points table
contact_point_type	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Validated against AR lookup type REGISTRY_STATUS</li> <li>▪ Cannot be set to null during update</li> </ul>
owner_table_name	IN	VARCHAR2	No	Validation: Not updateable
owner_table_id	IN	NUMBER	No	Validation: Not updateable
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Not updateable
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
contact_point_purpose	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB</li> <li>■ Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type=WEB</li> </ul>
primary_by_purpose	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_source	IN	VARCHAR2	No	Validation : Cannot be updated.

p\_edi\_rec record type

If you use the generic validation method, validations only apply when contact\_point\_type = EDI.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
edi_transaction_handling	IN	VARCHAR2	No	
edi_id_number	IN	VARCHAR2	No	Validation: Cannot be updated to null
edi_payment_method	IN	VARCHAR2	No	
edi_payment_format	IN	VARCHAR2	No	
edi_remittance_method	IN	VARCHAR2	No	
edi_remittance_instruction	IN	VARCHAR2	No	
edi_tp_header_id	IN	NUMBER	No	
edi_ece_tp_location_code	IN	VARCHAR2	No	
p_email_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = EMAIL.				
email_format	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type EMAIL_FORMAT.</li> <li>■ Cannot set to null during update</li> </ul>
email_address	IN	VARCHAR2	No	Validation: Cannot be set to null during update
p_phone_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = PHONE or PAGER.				
phone_calling_calendar	IN	VARCHAR2	No	
last_contact_dt_time	IN	DATE	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id
phone_area_code	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
phone_country_code	IN	VARCHAR2	No	Validation: Foreign key to hz_phone_country_codes.phone_country_code
Phone_number	IN	VARCHAR2	No	Validation: Mandatory if raw_phone_number is not passed in
phone_extension	IN	VARCHAR2	No	
phone_line_type	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type PHONE_LINE_TYPE.</li> <li>■ Cannot be updated to null</li> </ul>
raw_phone_number	IN	VARCHAR2	No	Validation: Mandatory if phone_number is not passed in
p_telex_rec record type				
If you use generic validations, validations only apply when contact_point_type = TELEX.				
telex_number	IN	VARCHAR2	No	Validation: Mandatory attribute
p_web_rec record type				
If you use the generic validation method, validations only apply when contact_point_type = WEB.				
web_type	IN	VARCHAR2	No	Validation: Cannot be set to null during update
url	IN	VARCHAR2	No	Validation: Cannot be set to null during update
p_eft_rec_record_type				
eft_transmission_program_id	IN	NUMBER	No	
eft_printing_program_id	IN	NUMBER	No	
eft_user_number	IN	VARCHAR2	No	
eft_swift_code	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record.</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from hz_contact_points</li> <li>■ Return new value after update</li> </ul>

#### Other Validations

- The Primary\_flag is a lookup code of lookup type YES/NO. The API automatically marks the first active, visible contact point per type for an entity to primary. When the user selects another contact point of same type to be primary, the previous primary contact point will be unset. Also, if the user inactivates the primary contact point, this primary contact point will be unset and next available active, visible contact point with same type will become primary; and if the user activates a contact point, and if this contact point becomes the first active, visible contact point of this type for this entity, the contact point will be marked as primary. Primary flag is defaulted to *N* if none of the above scenarios occur and the user does not pass any value for primary flag.
- An inactive contact can never be marked as primary.
- Only the primary URL and email for given parties are denormalized to HZ\_PARTIES table.
- primary\_by\_purpose is a lookup code of lookup type YES/NO. It is defaulted to 'N' if user does not pass a value. There is only one primary per purpose contact point exist for the combination of owner\_table\_name, owner\_table\_id, contact\_point\_type, and contact\_point\_purpose. If primary\_by\_purpose is set to 'Y', we need to unset the previous primary per purpose contact point to non-primary. Because setting primary\_by\_purpose is only making sense when contact\_point\_purpose has some value, we ignore the primary\_by\_purpose (setting it to 'N') if contact\_point\_purpose is null.



---

---

# Relationship Type and Relationship API Use

This chapter provides information about the following application programming interfaces:

- Relationship Type
- Relationship

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Relationship Type APIs

**PL/SQL Package Name:** HZ\_RELATIONSHIP\_TYPE\_V2PUB

**Java Class Name:** HzRelationshipTypeV2Pub

### PL/SQL Record Structure for Relationship Type

```
TYPE relationship_type_rec_type IS RECORD(  
    relationship_type_id          NUMBER,  
    relationship_type             VARCHAR2(30),  
    forward_rel_code             VARCHAR2(30),  
    backward_rel_code            VARCHAR2(30),  
    direction_code               VARCHAR2(30),  
    hierarchical_flag            VARCHAR2(1),  
    create_party_flag            VARCHAR2(1),  
    allow_relate_to_self_flag    VARCHAR2(1),  
    allow_circular_relationships VARCHAR2(1),  
    subject_type                 VARCHAR2(30),  
    object_type                  VARCHAR2(30),  
    status                       VARCHAR2(1),  
    created_by_module            VARCHAR2(150),  
    application_id               NUMBER  
    multiple_parent_allowed      VARCHAR2(1),  
    incl_unrelated_entities      VARCHAR2(1)  
    forward_role                 VARCHAR2(30)  
    backward_role                VARCHAR2(30)  
)
```

### Java Inner Class for Relationship Type

```
public static class RelationshipTypeRec {  
    public BigDecimal          relationship_type_id;  
    public String              relationship_type;  
    public String              forward_rel_code;  
    public String              backward_rel_code;  
    public String              direction_code;  
    public String              hierarchical_flag;  
    public String              create_party_flag;  
    public String              allow_relate_to_self_flag;  
    public String              allow_circular_relationships;  
    public String              subject_type;  
    public String              object_type;  
}
```

```

        public String                status;
        public String                created_by_module;
        public BigDecimal            application_id;
        public String                multiple_parent_allowed;
        public String                incl_unrelated_entities;
        public String                forward_role;
        public String                backward_role;

        public RelationshipTypeRec();
        public RelationshipTypeRec(boolean __RosettaUseGMISSValues);
    }

```

## Create Relationship Type API

### Description

This routine is used to create a Relationship Type. The API creates a record in the HZ\_RELATIONSHIP\_TYPES table. The relationship type defines the possible relationships that can be created between different types of parties or other entities. The API internally creates an additional record when forward relationship code and backward relationship code are different indicating the relationship can be created in two ways.

### PL/SQL Procedure

```

PROCEDURE create_relationship_type (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_relationship_type_rec  IN          RELATIONSHIP_TYPE_REC_TYPE,
    x_relationship_type_id   OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2,
)

```

### Java Method

```

public static void createRelationshipType(
    OracleConnection_connection,
    String
    RelationshipTypeRec                p_init_msg_list,
    BigDecimal [ ]                    p_relationship_type_rec,
    String [ ]                         x_relationship_type_id,
    BigDecimal [ ]                    x_return_status,
    String [ ]                         x_msg_count,
    String [ ]                         x_msg_data
) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Relationship Type API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_type_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
relationship_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute
forward_rel_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type PARTY_RELATIONS_TYPE</li> </ul>
backward_rel_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type PARTY_RELATIONS_TYPE</li> </ul>
direction_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type DIRECTION_CODE</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
hierarchical_flag	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type YES/NO.</p> <ul style="list-style-type: none"> <li>■ If hierarchical_flag = Y, then allow_circular_relationships must be N.</li> <li>■ If hierarchical_flag = Y, then direction_code must be P or C</li> <li>■ If hierarchical_flag = N, then multiple_parent_allowed must be Y.</li> <li>■ All relationship type records with same relationship_type value must have same value for hierarchical_flag.</li> </ul> <p>Default : N</p> <p>Comment : Indicates whether the relationship type is hierarchical.</p>
create_party_flag	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type YES/NO</p> <p>Default: N</p> <p>Comment: Indicates whether a denormalized party will be created for a relationship having this relationship type</p>
allow_relate_to_self_flag	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type YES/NO</p> <p>Default: N</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
allow_circular_relationships	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO <ul style="list-style-type: none"> <li>■ If hierarchical_flag = Y, then allow_circular_relationships must be N.</li> <li>■ If direction_code = N, the allow_circular_relationships must be Y.</li> <li>■ All relationship type records with same relationship_type value must have same value for allow_circular_relationships.</li> </ul> Default: Y
subject_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute Foreign key to fnd_object_instance_sets.instance_set_name
object_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute Foreign key to fnd_object_instance_sets.instance_set_name
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type CODE_STATUS
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
multiple_parent_allowed	IN	VARCHAR2	No	<p>Validation : Validated against AR lookup type YES/NO</p> <ul style="list-style-type: none"> <li>■ If hierarchical_flag = N, then multiple_parent_allowed must be Y.</li> <li>■ All relationship type records with same relationship_type value must have same value for multiple_parent_allowed.</li> </ul> <p>Comment : Indicates whether a child can have multiple parents. This is applicable when hierarchical_flag is Y.</p> <p>Default : Y</p>
incl_unrelated_entities	IN	VARCHAR2	No	<p>Validation : Validated against AR lookup type YES/NO</p> <p>Comment : Indicates whether entities not having a relationship will be included in a hierarchy or not. This is applicable when hierarchical_flag is Y.</p> <p>Default : N</p>
forward_role	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type.</li> <li>■ Forward_role should be unique.</li> </ul> <p>Default : Defaulted to 'USER_ROLE_'    to_char(relationship_type_id)</p> <p>Comment : This describes the role a subject party plays in a relationship.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
backward_role	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type.</li> <li>■ Backward_role should be unique.</li> </ul> Comment: This describes the role an object party plays in a relationship. Default: Defaulted to 'USER_ROLE_'    to_char(relationship_type_id)
x_relationship_type_id	OUT	NUMBER	No	Comment: Return relationship_type_id of the relationship type record created

#### Other Validations

- The combination of relationship type, forward relationship code, backward relationship code, subject type, and object\_type should be unique.
- The combination of forward relationship code, subject type, and object type should be able to identify a unique backward relationship code. Thus, a second record with same combination of these three should have same backward relationship code.
- The direction code will be used to determine if a backward relationship type should be created. If the direction code is *P* or *C*, we will create a second relationship type. If the forward relationship code is the same as the backward relationship code, the direction code should be *N* - Non directional.
- If a relationship type is created with the hierarchical\_flag set to *Y*, the API does not allow a circular relationship with that relationship type.
- If a relationship type is created with the hierarchical\_flag set to *Y*, that is generally called hierarchical relationship type. If you create or update relationships of that hierarchical relationship type, relationship information is denormalized to the HZ\_HIERARCHY\_NODES table with level, date effectivity information, and so on. You can query that table at any time to view the hierarchy information.

- If `forward_rel_code` and `backward_rel_code` are different, then the `forward_role` and the `backward_role` should also be different.
- If `forward_rel_code` and `backward_rel_code` are the same, then the `forward_role` and the `backward_role` should also be same.
- When `forward_role` or `backward_role` is not passed, it defaults to `'USER_ROLE_' || to_char(relationship_type_id)` and no lookup is created through the API for this defaulted role.
- A new lookup type, `HZ_RELATIONSHIP_TYPE`, is introduced for relationship types. In the API, no validation is done for a `relationship_type` against the lookup because of backward compatibility. Before creating a relationship type, a lookup for that relationship type must be created.

## Update Relationship Type API

### Description

This routine is used to update a Relationship Type. The API updates a record in the `HZ_RELATIONSHIP_TYPES` table.

### PL/SQL Procedure:

```
PROCEDURE update_relationship_type (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_relationship_type_rec  IN          RELATIONSHIP_TYPE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count            OUT         NUMBER,
    x_msg_data             OUT         VARCHAR2
)
```

### Java Method

```
public static void updateRelationshipType(
    OracleConnection connection,
    String p_init_msg_list,
    RelationshipTypeRec p_relationship_type_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Relationship Type API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_type_id	IN	NUMBER	Yes	Validation: valid relationship_type_id should be passed in Comment: Pass the relationship_type_id from hz_relationship_types record
relationship_type	IN	VARCHAR2	No	Validation: Not updateable
forward_rel_code	IN	VARCHAR2	No	Validation: Not updateable
backward_rel_code	IN	VARCHAR2	No	Validation: Not updateable
direction_code	IN	VARCHAR2	No	Validation: Not updateable
hierarchical_flag	IN	VARCHAR2	No	Validation: Not updateable
create_party_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Can be updated. No relationship record, created with the current setup of create_party_flag, can exist with this relationship type.</li> <li>■ Validate against AR lookup type YES/NO.</li> </ul>
allow_relate_to_self_flag	IN	VARCHAR2	No	Validation: Not updateable
allow_circular_relationships	IN	VARCHAR2	No	Validation: Not updateable
subject_type	IN	VARCHAR2	No	Validation: Not updateable
object_type	IN	VARCHAR2	No	Validation: Not updateable

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>Validated against AR lookup type CODE_STATUS</li> <li>Cannot set to null during update</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
multiple_parent_allowed	IN	VARCHAR2	No	Validation : Non updateable
incl_unrelated_entities	IN	VARCHAR2	No	Validation :Validated against AR lookup type YES/NO
forward_role	IN	VARCHAR2	No	Validation : Non updateable. Non nullable
backward_role	IN	VARCHAR2	No	Validation : Non updateable. Non nullable
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>Mandatory attribute</li> <li>Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>Pass the current object_version_number of the record from hz_relationship_types</li> <li>Return new value after update</li> </ul>

### Other Validations

- The combination of relationship type, forward relationship code, backward relationship code, subject type, and object\_type should be unique.
- The combination of forward relationship code, subject type, and object type should be able to identify an unique backward relationship code. Thus, a

second record with same combination of these three should have same backward relationship code.

- The direction code will be used to determine if a backward relationship type should be created. If the direction code is 'P' or 'C', we will create a second relationship type. If the forward relationship code is the same as the backward relationship code, the direction code should be 'N' - Non directional.

## Relationship APIs

**PL/SQL Package Name: HZ\_RELATIONSHIP\_V2PUB**

**Java Class Name: HzRelationshipV2Pub**

**PL/SQL Constant:**

```
G_MISS_CONTENT_
  CONSTANT VARCHAR2(30) := USER_ENTERED;
```

### PL/SQL Record Structure for Relationship

```
TYPE relationship_rec_type IS RECORD(
  relationship_id          NUMBER,
  subject_id              NUMBER,
  subject_type            VARCHAR2(30),
  subject_table_name      VARCHAR2(30),
  object_id               NUMBER,
  object_type             VARCHAR2(30),
  object_table_name       VARCHAR2(30),
  relationship_code       VARCHAR2(30),
  relationship_type       VARCHAR2(30),
  comments                VARCHAR2(240),
  start_date              DATE,
  end_date                DATE,
  status                  VARCHAR2(1),
  content_source_type     VARCHAR2(30), := G_MISS_CONTENT_SOURCE_TYPE,
  attribute_category      VARCHAR2(30),
  attribute1              VARCHAR2(150),
  attribute2              VARCHAR2(150),
  attribute3              VARCHAR2(150),
  attribute4              VARCHAR2(150),
  attribute5              VARCHAR2(150),
  attribute6              VARCHAR2(150),
  attribute7              VARCHAR2(150),
  attribute8              VARCHAR2(150),
  attribute9              VARCHAR2(150),
  attribute10             VARCHAR2(150),
  attribute11             VARCHAR2(150),
  attribute12             VARCHAR2(150),
  attribute13             VARCHAR2(150),
  attribute14             VARCHAR2(150),
```

```

attribute15          VARCHAR2 (150) ,
attribute16          VARCHAR2 (150) ,
attribute17          VARCHAR2 (150) ,
attribute18          VARCHAR2 (150) ,
attribute19          VARCHAR2 (150) ,
attribute20          VARCHAR2 (150) ,
created_by_module    VARCHAR2 (150) ,
application_id       NUMBER,
party_rec            HZ_PARTY_V2PUB.
                    PARTY_REC_TYPE:= HZ_PARTY_V2PUB.
                    G_MISS_PARTY_REC

additional_information1  VARCHAR2 (150) ,
additional_information2  VARCHAR2 (150) ,
additional_information3  VARCHAR2 (150) ,
additional_information4  VARCHAR2 (150) ,
additional_information5  VARCHAR2 (150) ,
additional_information6  VARCHAR2 (150) ,
additional_information7  VARCHAR2 (150) ,
additional_information8  VARCHAR2 (150) ,
additional_information9  VARCHAR2 (150) ,
additional_information10 VARCHAR2 (150) ,
additional_information11 VARCHAR2 (150) ,
additional_information12 VARCHAR2 (150) ,
additional_information13 VARCHAR2 (150) ,
additional_information14 VARCHAR2 (150) ,
additional_information15 VARCHAR2 (150) ,
additional_information16 VARCHAR2 (150) ,
additional_information17 VARCHAR2 (150) ,
additional_information18 VARCHAR2 (150) ,
additional_information19 VARCHAR2 (150) ,
additional_information20 VARCHAR2 (150) ,
additional_information21 VARCHAR2 (150) ,
additional_information22 VARCHAR2 (150) ,
additional_information23 VARCHAR2 (150) ,
additional_information24 VARCHAR2 (150) ,
additional_information25 VARCHAR2 (150) ,
additional_information26 VARCHAR2 (150) ,
additional_information27 VARCHAR2 (150) ,
additional_information28 VARCHAR2 (150) ,
additional_information29 VARCHAR2 (150) ,
additional_information30 VARCHAR2 (150) ,
percentage_ownership    NUMBER
actual_content_source    VARCHAR2 (30)
)

```

**Java Inner Class for Relationship**

```
public static class RelationshipRec {
    public BigDecimal relationship_id;
    public BigDecimal subject_id;
    public String subject_type;
    public String subject_table_name;
    public BigDecimal object_id;
    public String object_type;
    public String object_table_name;
    public String relationship_code;
    public String relationship_type;
    public String comments;
    public java.sql.Timestamp start_date;
    public java.sql.Timestamp end_date;
    public String status;
    public String content_source_type;
    public String attribute_category;
    public String attribute1;
    public String attribute2;
    public String attribute3;
    public String attribute4;
    public String attribute5;
    public String attribute6;
    public String attribute7;
    public String attribute8;
    public String attribute9;
    public String attribute10;
    public String attribute11;
    public String attribute12;
    public String attribute13;
    public String attribute14;
    public String attribute15;
    public String attribute16;
    public String attribute17;
    public String attribute18;
    public String attribute19;
    public String attribute20;
    public String created_by_module;
    public BigDecimal application_id;
    public
    HzPartyV2Pub.PartyRec
    public String additional_information1;
    public String additional_information2;
    public String additional_information3;
    public String additional_information4;
```

```
public String additional_information5;
public String additional_information6;
public String additional_information7;
public String additional_information8;
public String additional_information9;
public String additional_information10;
public String additional_information11;
public String additional_information12;
public String additional_information13;
public String additional_information14;
public String additional_information15;
public String additional_information16;
public String additional_information17;
public String additional_information18;
public String additional_information19;
public String additional_information20;
public String additional_information21;
public String additional_information22;
public String additional_information23;
public String additional_information24;
public String additional_information25;
public String additional_information26;
public String additional_information27;
public String additional_information28;
public String additional_information29;
public String additional_information30;
public BigDecimal percentage_ownership;
public String actual_content_source;

public RelationshipRec();
public RelationshipRec(boolean __RosettaUseGMISSValues);
}
```

## Create Relationship API

### Description

This routine is used to create a Relationship between two parties or other entities. The API creates a record in the HZ\_RELATIONSHIPS table. This defines the relationship that exists between Parties of type PERSON, ORGANIZATION, and other entities that are defined in FND\_OBJECT\_INSTANCE\_SETS. Each relationship can be viewed from either ways. So an additional relationship record is created to store the reverse relationship. The relationship code, relationship type, subject type and object type must be a valid combination already defined in the

HZ\_RELATIONSHIP\_TYPES table. The two relationship records have the same relationship\_id, they are distinguishable by the directional\_flag column.

If a hierarchical relationship type (hierarchical\_flag = Y) is used to create a relationship, the relationship information is denormalized to the HZ\_HIERARCHY\_NODES table with level, effective date, and so on. The API ensures that no circular relationship is created, so that all of the relationships using that relationship type are hierarchical.

### PL/SQL Procedure

```
PROCEDURE create_relationship (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_relationship_rec       IN          RELATIONSHIP_REC_TYPE,
    x_relationship_id       OUT         NUMBER,
    x_party_id              OUT         NUMBER,
    x_party_number          OUT         VARCHAR2,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2,
    p_create_org_create     IN          VARCHAR:=Y
)
```

### Java Method

```
public static void createRelationship(
    OracleConnection_connection,
    String p_init_msg_list,
    RelationshipRec p_relationship_rec,
    BigDecimal [ ] x_relationship_id,
    BigDecimal [ ] x_party_id,
    String [ ] x_party_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Relationship API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_id	IN	NUMBER	No	Validation: Unique if passed in, else generated by sequence
subject_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute.</li> <li>■ Validated against Primary Key in <code>fnf_objects.obj_name</code> where <code>fnf_objects.object_id = fnf_object_instance_sets.object_id</code> and <code>fnf_object_instance_sets.instance_set_name = subject_type</code></li> </ul>
subject_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to <code>fnf_object_instance_sets.instance_set_name</code></li> </ul>
subject_table_name	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to <code>fnf_objects.obj_name</code>.</li> </ul>
object_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against Primary Key in <code>fnf_objects.obj_name</code> where <code>fnf_objects.object_id = fnf_object_instance_sets.object_id</code> and <code>fnf_object_instance_sets.instance_set_name = subject_type</code></li> </ul>
object_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to <code>fnf_object_instance_sets.instance_set_name</code></li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
object_table_name	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign Key to fnd_objects.obj_name</li> </ul>
relationship_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Validated against AR lookup type PARTY_RELATIONS_TYPE</li> <li>▪ Required to be a valid relationship code for the particular relationship type requested.</li> </ul>
relationship_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign key to hz_relationship_types.relationship_type</li> </ul>
comments	IN	VARCHAR2	No	
start_date	IN	DATE	Yes	Validation: Must be less than end_date if end_date is passed Default: sysdate
end_date	IN	DATE	No	Default: 31-DEC-4712 Validation: Must be greater than start_date
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Validated against AR lookup type CONTENT_SOURCE_TYPE Default: USER_ENTERED
attribute_category	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
additional_ information1	IN	VARCHAR2	No	
additional_ information2	IN	VARCHAR2	No	
additional_ information3	IN	VARCHAR2	No	
additional_ information4	IN	VARCHAR2	No	
additional_ information5	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_information6	IN	VARCHAR2	No	
additional_information7	IN	VARCHAR2	No	
additional_information8	IN	VARCHAR2	No	
additional_information9	IN	VARCHAR2	No	
additional_information10	IN	VARCHAR2	No	
additional_information11	IN	VARCHAR2	No	
additional_information12	IN	VARCHAR2	No	
additional_information13	IN	VARCHAR2	No	
additional_information14	IN	VARCHAR2	No	
additional_information15	IN	VARCHAR2	No	
additional_information16	IN	VARCHAR2	No	
additional_information17	IN	VARCHAR2	No	
additional_information18	IN	VARCHAR2	No	
additional_information19	IN	VARCHAR2	No	
additional_information20	IN	VARCHAR2	No	
additional_information21	IN	VARCHAR2	No	
additional_information22	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_information23	IN	VARCHAR2	No	
additional_information24	IN	VARCHAR2	No	
additional_information25	IN	VARCHAR2	No	
additional_information26	IN	VARCHAR2	No	
additional_information27	IN	VARCHAR2	No	
additional_information28	IN	VARCHAR2	No	
additional_information29	IN	VARCHAR2	No	
additional_information30	IN	VARCHAR2	No	
percentage_ownership	IN	NUMBER	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
actual_content_source	IN	VARCHAR2	No	Validation : Validated against AR lookup type CONTENT_SOURCE_TYPE Default : USER_ENTERED
party_rec	Record Type attributes			
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated by sequence

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_number	IN	VARCHAR2	Yes/No	Validation: Mandatory if HZ_GENERATE_PARTY_NUMBER=N, else generated by sequence
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_relationship_id	OUT	NUMBER	No	Comment: Return relationship_id of the relationship record created
x_party_id	OUT	NUMBER	No	Comment: Return party_id for the relationship created
x_party_number	OUT	NUMBER	No	Comment: Return party number of the party created
p_create_org_contact	IN	VARCHAR2	No	Comment: This parameter is for use only by TCA development Default: Y.

### Other Validations

- When you call the create relationship procedure, two new records will be created in HZ\_RELATIONSHIPS table and one record will be created in the HZ\_ORG\_CONTACTS table.
- You are required to specify a subject id, an object id, a subject type, an object type, a subject table name (where the subject belongs to e.g. 'HZ\_PARTIES' for subject type of 'PERSON', 'ORGANIZATION'), an object table name, a relationship type, a relationship code, and start date to create a relationship.
- The relationship code must be either a forward or backward relationship code defined in active records in the HZ\_RELATIONSHIP\_TYPES table for your relationship type.
- The end date will be default to '31-DEC-4712' if not specified. The end date must be greater than the start date.

- For a given `subject_id`, `object_id`, `relationship_code`, there can be no overlap of the start date and the end date.
- The subject type and the object type will be used to do the foreign key check for the subject id and the object id column. The subject id should be a valid object instance defined in the object instance set which is specified as the subject type. The object id should be a valid object instance defined in the object instance set which is specified as the object type.
- The subject id and the object id cannot be the same value unless the self related flag of the relationship type is set to yes.
- It will check the “denormalized to party” flag in the `HZ_RELATIONSHIP_TYPES` table to see if a denormalized party is necessary. If the flag is set to yes, a party will be created for the relationship. Only the relationship with both `subject_table_name` and `object_table_name` as 'HZ\_PARTIES' can be denormalized to HZ\_PARTIES table.
- A record will be created in the `HZ_RELATIONSHIPS` table with the relationship code passed as input parameter.
- It will find the 2<sup>nd</sup> relationship code from the `HZ_RELATIONSHIP_TYPES` table, and use it to create a second relationship. The 2<sup>nd</sup> relationship will have the same `relationship_id` and `party_id` as the first relationship. The requested relationship will be created with `directional_code` value *F* (meaning forward) and the reciprocal relationship will be created with `directional_code` value *B* (meaning backward).
- If a hierarchical relationship type is used, the API ensures no circular relationship is created during the creation of a relationship. Usually the API also ensures that a child has only one parent at a time. However, if the attribute `multiple_parents_allowed = Y`, then you can create multiple parents for a child. This is a very special case and you must cautiously use this attribute setting.
- If a hierarchical relationship type is used, API denormalizes the relationship information into `HZ_HIERARCHY_NODES` table. This denormalized table contains relationship information along with level, date effectivity of relationships etc. Simple query can be written against that table to query hierarchy information.

## Update Relationship API

### Description

This routine is used to update a Relationship. Because there are two records for each relationship (forward and backward), the API updates two records in the HZ\_RELATIONSHIPS table. Additionally you can update the denormalized party record for the relationship (if it is present) by passing party's id and party's object version number.

### PL/SQL Procedure

```
PROCEDURE update_relationship (  
    p_init_msg_list                IN          VARCHAR2:= FND_API.G_FALSE,  
    p_relationship_rec             IN          RELATIONSHIP_REC_TYPE,  
    p_object_version_number        IN OUT     NUMBER,  
    p_party_object_version_number  IN OUT     NUMBER,  
    x_return_status                OUT        VARCHAR2,  
    x_msg_count                   OUT        NUMBER,  
    x_msg_data                    OUT        VARCHAR2  
)
```

### Java Method

```
public static void updateRelationship(  
    OracleConnection_connection,  
    String                                p_init_msg_list,  
    RelationshipRec                       p_relationship_rec,  
    BigDecimal [ ]                       p_object_version_number,  
    BigDecimal [ ]                       p_party_object_version_number,  
    String [ ]                            x_return_status,  
    BigDecimal [ ]                       x_msg_count,  
    String [ ]                            x_msg_data  
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Relationship API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_id	IN	NUMBER	Yes	Validation: Valid relationship_id should be passed in Comment: Pass the relationship_id from the hz_relationships record
subject_id	IN	NUMBER	No	Validation: Not updateable
subject_type	IN	VARCHAR2	No	Validation: Not updateable
subject_table_name	IN	VARCHAR2	No	Validation: Not updateable
object_id	IN	NUMBER	No	Validation: Not updateable
object_type	IN	VARCHAR2	No	Validation: Not updateable
object_table_name	IN	VARCHAR2	No	Validation: Not updateable
relationship_code	IN	VARCHAR2	No	Validation: Not updateable
relationship_type	IN	VARCHAR2	No	Validation: Not updateable
comments	IN	VARCHAR2	No	
start_date	IN	DATE	No	Validation: Cannot be set to null during update Cannot be greater than end_date
end_date	IN	DATE	No	Validation: Must be greater than start date
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Validated against AR lookup type REGISTRY_STATUS</li> <li>■ Cannot be updated to null</li> </ul>
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Cannot be updated.
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
additional_ information1	IN	VARCHAR2	No	
additional_ information2	IN	VARCHAR2	No	
additional_ information3	IN	VARCHAR2	No	
additional_ information4	IN	VARCHAR2	No	
additional_ information5	IN	VARCHAR2	No	
additional_ information6	IN	VARCHAR2	No	
additional_ information7	IN	VARCHAR2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_information8	IN	VARCHAR2	No	
additional_information9	IN	VARCHAR2	No	
additional_information10	IN	VARCHAR2	No	
additional_information11	IN	VARCHAR2	No	
additional_information12	IN	VARCHAR2	No	
additional_information13	IN	VARCHAR2	No	
additional_information14	IN	VARCHAR2	No	
additional_information15	IN	VARCHAR2	No	
additional_information16	IN	VARCHAR2	No	
additional_information17	IN	VARCHAR2	No	
additional_information18	IN	VARCHAR2	No	
additional_information19	IN	VARCHAR2	No	
additional_information20	IN	VARCHAR2	No	
additional_information21	IN	VARCHAR2	No	
additional_information22	IN	VARCHAR2	No	
additional_information23	IN	VARCHAR2	No	
additional_information24	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
additional_information25	IN	VARCHAR2	No	
additional_information26	IN	VARCHAR2	No	
additional_information27	IN	VARCHAR2	No	
additional_information28	IN	VARCHAR2	No	
additional_information29	IN	VARCHAR2	No	
additional_information30	IN	VARCHAR2	No	
percentage_ownership	IN	NUMBER	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_source	IN	VARCHAR2	No	Validation: Cannot be updated.
party_rec Record Type attributes				
party_id	IN	NUMBER	No	Validation: Valid party_id should be passed in to update party sequence Comment: <ul style="list-style-type: none"> <li>■ Pass the party_id from the hz_parties record</li> <li>■ Pass only if you want to update the party record</li> </ul>
party_number	IN	VARCHAR2	No	Validation: Not updateable
validated_flag	IN	VARCHAR2	No	Validation: Not updateable
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Validated against AR lookup type REGISTRY_STATUS</li> <li>▪ Cannot be updated to null</li> </ul>
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute for relationship record</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number from hz_relationships table</li> <li>■ Return new value after update</li> </ul>
p_party_object_version_number	IN OUT	NUMBER	Yes/No	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute if party record for the relationship needs to be updated</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number from hz_parties if you want to update party record</li> <li>■ Return new value after update</li> </ul>

#### Other validations

- When you call the create relationship procedure, two new records will be created in HZ\_RELATIONSHIPS table.
- You are required to specify a subject id, an object id, a subject type, an object type, a subject table name (where the subject belongs to e.g. 'HZ\_PARTIES' for

subject type of 'PERSON', 'ORGANIZATION'), an object table name, a relationship type, a relationship code, and start date to create a relationship.

- The relationship code must be either a forward or backward relationship code defined in active records in the HZ\_RELATIONSHIP\_TYPES table for your relationship type.
- The end date will be default to '31-DEC-4712' if not specified. The end date must be greater than the start date.
- For a given subject\_id, object\_id, relationship\_code, there can be no overlap of the start\_date and the end\_date.
- The subject type and the object type will be used to do the foreign key check for the subject id and the object id column. The subject id should be a valid object instance defined in the object instance set which is specified as the subject type. The object id should be a valid object instance defined in the object instance set which is specified as the object type.
- The subject id and the object id cannot be the same value unless the self related flag of the relationship type is set to yes.
- It will check the “denormalized to party” flag in the HZ\_RELATIONSHIP\_TYPES table to see if a denormalized party is necessary. If the flag is set to yes, a party will be created for the relationship. Only the relationship with both subject\_table\_name and object\_table\_name as 'HZ\_PARTIES' can be denormalized to HZ\_PARTIES table.
- A record will be created in the HZ\_RELATIONSHIPS table with the relationship code passed as input parameter.
- It will find the 2<sup>nd</sup> relationship code from the HZ\_RELATIONSHIP\_TYPES table, and use it to create a second relationship. The 2<sup>nd</sup> relationship will have the same relationship\_id and party\_id as the first relationship. The requested relationship will be created with directional\_code value 'F' (meaning forward) and the reciprocal relationship will be created with directional\_code value 'B' (meaning backward).
- Information in the Create Relationship API section about hierarchical types is also valid for the Update Relationship API section.



---

---

## Classification API Use

This chapter provides information about the Classification application programming interface.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Classification APIs

**PL/SQL Package Name: HZ\_CLASSIFICATION\_V2PUB**

**Java Class Name: HzClassificationV2Pub**

### PL/SQL Record Structure for Class Category

```
TYPE class_category_rec_type IS RECORD (  
    class_category                VARCHAR2(30),  
    allow_multi_parent_flag      VARCHAR2(1),  
    allow_multi_assign_flag      VARCHAR2(1),  
    allow_leaf_node_only_flag    VARCHAR2(1),  
    created_by_module            VARCHAR2(150),  
    application_id               NUMBER  
    delimiterVARCHAR2(1),  
)
```

### PL/SQL Record Structure for Class Code Relation

```
TYPE class_code_relation_rec_type IS RECORD  
    class_category                VARCHAR2(30),  
    class_code                   VARCHAR2(30),  
    sub_class_code               VARCHAR2(30),  
    start_date_active            DATE,  
    end_date_active              DATE,  
    created_by_module            VARCHAR2(150),  
    application_id               NUMBER  
)
```

### PL/SQL Record Structure for Code Assignment

```
TYPE code_assignment_rec_type IS RECORD (  
    code_assignment_id           NUMBER,  
    owner_table_name            VARCHAR2(30),  
    owner_table_id              NUMBER,  
    class_category              VARCHAR2(30),  
    class_code                  VARCHAR2(30),  
    primary_flag                VARCHAR2(1),  
    content_source_type         VARCHAR2(30) := HZ_PARTY_V2PUB.G_MISS_  
    CONTENT_SOURCE_TYPE,  
    start_date_active           DATE,  
    end_date_active             DATE,  
    status                      VARCHAR2(1),
```

```

        created_by_module          VARCHAR2(150),
        application_id             NUMBER
        rank                       NUMBER
    )

```

## PL/SQL Record Structure for Class Category Use

```

TYPE class_category_use_rec_type IS RECORD (
    class_category          VARCHAR2(30),
    owner_table            VARCHAR2(240),
    column_name            VARCHAR2(240),
    additional_where_clause VARCHAR2(4000),
    created_by_module      VARCHAR2(150),
    application_id         NUMBER
)

```

## Java Inner Class for Class Category

```

public static class ClassCategoryRec {
    public String          class_category;
    public String          allow_multi_parent_flag;
    public String          allow_multi_assign_flag;
    public String          allow_leaf_node_only_flag;
    public String          created_by_module;
    public BigDecimal      application_id;
    public String          delimiter

    public ClassCategoryRec();
    public ClassCategoryRec(boolean __RosettaUseGMISSValues);
}

```

## Java Inner Class for Class Code Relation

```

public static class ClassCodeRelationRec {
    public String          class_category;
    public String          class_code;
    public String          sub_class_code;
    public java.sql.Timestamp start_date_active;
    public java.sql.Timestamp end_date_active;
    public String          created_by_module;
    public BigDecimal      application_id;

    public ClassCodeRelationRec();
    public ClassCodeRelationRec(boolean __RosettaUseGMISSValues);
}

```

## Java Inner Class for Code Assignment

```
public static class CodeAssignmentRec {
    public BigDecimal          code_assignment_id;
    public String              owner_table_name;
    public BigDecimal          owner_table_id;
    public String              class_category;
    public String              class_code;
    public String              primary_flag;
    public String              content_source_type;
    public java.sql.Timestamp  start_date_active;
    public java.sql.Timestamp  end_date_active;
    public String              status;
    public String              created_by_module;
    public BigDecimal          application_id;
    public String              rank;

    public CodeAssignmentRec();
    public CodeAssignmentRec(boolean __RosettaUseGMISSValues);
}
```

## Java Inner Class for Class Category Use

```
public static class ClassCategoryUseRec {
    public String              class_category;
    public String              owner_table;
    public String              column_name;
    public String              additional_where_clause;
    public String              created_by_module;
    public BigDecimal          application_id;

    public ClassCategoryUseRec();
    public ClassCategoryUseRec(boolean __RosettaUseGMISSValues);
}
```

## Create Class Category API

### Description

This routine is used to create a Class Category. The API creates a record in the HZ\_CLASS\_CATEGORIES table. A Class Category provides a way to classify parties and party sites. For example, NAICS\_1997 (1997 North American Industry Classification System) is a class category. A class category corresponds to an AR lookup type and the related class codes are lookup codes of the lookup type. Users

has to create a valid lookup type before creating the class category using that lookup type.

**PL/SQL Procedure**

```
PROCEDURE create_class_category(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_class_category_rec    IN          CLASS_CATEGORY_REC_TYPE,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

**Java Method**

```
public static void createClassCategory (
    OracleConnection_connection,
    String p_init_msg_list,
    ClassCategoryRec p_class_category_rec,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

**Parameter Description And Validation**

The following table lists information about the parameters in the Create Class Category API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	ValidationDefaultComment
class_category	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Unique</li> <li>▪ A Class Category correspond to a LOOKUP_TYPE</li> </ul>
allow_multi_parent_flag	IN	VARCHAR2	No	Validation: Validated against FND lookup type YES/NO
allow_multi_assign_flag	IN	VARCHAR2	No	Validation: Validated against FND lookup type YES/NO
allow_leaf_node_only_flag	IN	VARCHAR2	No	Validation: Validated against FND lookup type YES/NO

Parameter Name	Type	Data Type	Required	ValidationDefaultComment
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
delimiter	IN	VARCHAR2	No	Comment: Used for concatenated class code in the HZ_CLASS_CODE_DENORM table. Default value is / .

## Update Class Category API

### Description

This routine is used to update a Class Category. The API updates a record in the HZ\_CLASS\_CATEGORIES table.

### PL/SQL Procedure

```
PROCEDURE update_class_category(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_class_category_rec    IN          CLASS_CATEGORY_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
;
```

### Java Method

```
public static void updateClassCategory(
    OracleConnection_connection,
    String
    ClassCategoryRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) throws SQLException;
    p_init_msg_list,
    p_class_category_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data
```

### Parameter Description And Validation

The following table lists information about the parameters in the Update Class Category API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Unique</li> <li>▪ A Class Category correspond to a LOOKUP_TYPE</li> </ul>
allow_multi_parent_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Validated against FND lookup type YES/NO</li> <li>▪ This flag cannot be updated from Y to N, if there are any class codes related to more than one parent code</li> </ul>
allow_multi_assign_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Validated against FND lookup type YES/NO</li> <li>▪ This flag cannot be updated from Y to N, if there are any class codes assigned to more than one instance of HZ_PARTIES or HZ_RELATIONSHIP_TYPES or any entities that uses the classification model.</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
allow_leaf_node_only_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>Validated against FND lookup type YES/NO</li> <li>This flag cannot be updated from Y to N, if there are any class codes assigned to more than one instance of HZ_PARTIES or HZ_RELATIONSHIP_TYPES or any entities that uses the classification model.</li> </ul>
created_by_module	IN	VARCHAR2	Yes	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
delimiter	IN	VARCHAR2	No	Comment: Used for concatenated class code in the HZ_CLASS_CODE_DENORM table. Default value is / .
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>Mandatory attribute</li> <li>Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>Pass the current object_version_number from HZ_CLASS_CATEGORIES</li> <li>Return new value after update</li> </ul>

## Create Class Code Relation API

### Description

This routine is used to create a Class Code Relation. The API creates a record in the HZ\_CLASS\_CODE\_RELATIONS table. The class codes are related to a class category. For example, the class category NAICS\_1997 has

- the code 11 (Agriculture, Forestry, Fishing and Hunting).
- the code 111 (Crop Production)
- the code 1111 (Oilseed and Grain Farming)

You can relate the code 11 as parent code of the code 111, which in turn can be related as parent code of the code 1111. This way you can set up the class codes as parent child relationship. As a class category is a lookup type, the class codes of a class category are the lookup codes of that lookup type.

### PL/SQL Procedure

```
PROCEDURE create_class_code_relation
  p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
  p_class_code_relation_rec IN        CLASS_CODE_RELATION_REC_TYPE,
  x_return_status         OUT         VARCHAR2,
  x_msg_count            OUT         NUMBER,
  x_msg_data             OUT         VARCHAR2
)

```

### Java Method

```
public static void createClassCodeRelation(
  OracleConnection_connection,
  String p_init_msg_list,
  ClassCodeRelationRec p_class_code_relation_rec,
  String [ ] x_return_status,
  BigDecimal [ ] x_msg_count,
  String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Class Code Relation API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES.CLASS_CATEGORY

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column</li> <li>■ sub_class_code cannot be an ancestor code of class_code. This validation avoids recursive relationship</li> </ul>
sub_class_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column</li> <li>■ class_code cannot be a descendant of sub_class_code. This validation avoid recursive relationship</li> <li>■ If allow_multi_parent_flag = N then the sub_class_code must not have any class_code related to it for the period range from start_date_active to end_date_active.</li> </ul>
start_date_active	IN	DATE	No	
end_date_active	IN	DATE	No	Validation: <ul style="list-style-type: none"> <li>■ A Class Code Relation must be unique for a period of time range from start_date_active to end_date_active.</li> <li>■ The end_date_active must be null or greater then the start_date_active.</li> </ul>
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated

## Update Class Code Relation API

### Description

This routine is used to update a Class Code Relation. The API updates a record in table HZ\_CLASS\_CODE\_RELATIONS.

### PL/SQL Procedure

```
PROCEDURE update_class_code_relation(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_class_code_relation_rec IN          CLASS_CODE_RELATION_REC_TYPE,
    p_object_version_number IN OUT      NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)
```

### Java Method

```
public static void updateClassCodeRelation(
    OracleConnection_connection,
    String p_init_msg_list,
    ClassCodeRelationRec p_class_code_relation_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Class Code Relation API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES
class_code	IN	VARCHAR2	Yes	Validation: Validated the existence of the relation (class_category, class_code, sub_class_code, start_date_active)
sub_class_code	IN	VARCHAR2	Yes	Validation: Part of the existence validation
start_date_active	IN	DATE	No	Validation: Part of the existence validation
end_date_active	IN	DATE	No	Validation: <ul style="list-style-type: none"> <li>■ A Class Code Relation (class_category, class_code, sub_class_Code) must be unique for a period of time range from Start_date_Active to End_Date_Active.</li> <li>■ End_date_Active must be null or greater than Start_Date_Active.</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from hz_class_code_relations</li> <li>■ Returns new value after update</li> </ul>

## Create Code Assignment API

### Description

This routine is used to create a Code Assignment. The API creates a record in the HZ\_CODE\_ASSIGNMENTS table. An assignment of class code links an instance of the class code to an instance of the classified table. The HZ\_CODE\_ASSIGNMENTS table is an intersection table that links the classification codes in the AR\_LOOKUPS view to the instances of the parties or other entities stored in the table identified in the OWNER\_TABLE\_NAME column. The OWNER\_TABLE\_ID column holds the value of the ID column of the classified table.

### PL/SQL Procedure

```
PROCEDURE create_code_assignment(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_code_assignment_rec    IN          CODE_ASSIGNMENT_REC_TYPE,
    x_return_status          OUT         VARCHAR2,
    x_msg_count              OUT         NUMBER,
    x_msg_data               OUT         VARCHAR2,
    x_code_assignment_id     OUT         NUMBER
)
```

### Java Method

```
public static void createCodeAssignment(
    OracleConnection_connection,
    String                               p_init_msg_list,
    CodeAssignmentRec                     p_code_assignment_rec,
    String [ ]                             x_return_status,
    BigDecimal [ ]                         x_msg_count,
    String [ ]                             x_msg_data,
    BigDecimal [ ]                         x_code_assignment_id
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Code Assignment API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
code_assignment_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
owner_table_name	IN	VARCHAR2	No	Validation: Validated against fnd lookup values where lookup type = 'CODE_ASSIGN_OWNER_TABLE'
owner_table_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>Must be an Id column value of the entity which is going to use the classification model.</li> <li>The combination (owner_table_name, owner_table_id, class_category, class_code, content_source_type, start_active_date) must be unique</li> </ul>
class_category	IN	VARCHAR2	No	Validation: Validated against HZ_CLASS_CATEGORIES
class_code	IN	VARCHAR2	No	Validation: Validated against FND lookup values where the LOOKUP_TYPE = CLASS_CATEGORY. A class code must be a valid lookup_code from the lookup_type which name is the CLASS_CATEGORY.
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>Validated against FND lookup values where lookup type YES/NO</li> <li>An owner_table_id can only have one primary (PRIMARY_FLAG = Y) assignment to a class_code of one class_category for one content_source_type at one time.</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Validated against FND lookup values where lookup type CONTENT_SOURCE_TYPE  Default: USER_ENTERED
start_date_active	IN	DATE	No	
end_date_active	IN	DATE	No	Validation: must be null or greater than start_date_active
Status	IN	VARCHAR2	No	Validation: Validated against AR_LOOKUP type CODE_STATUS
created_by_module	IN	VARCHAR2	Yes	Comment: Text to indicate module from which creation of record is initiated  Validation: Mandatory attribute
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated
x_code_assignment_id	OUT	NUMBER	No	Comment: Return the code_assignment_id of the code assignment record created
rank	IN	NUMBER	No	Comment: Provides the ability to rank classes if multiple classes have been assigned to a party.

## Update Code Assignment API

### Description

This routine is used to update a Code Assignment. The API updates a record in the HZ\_CODE\_ASSIGNMENTS table.

### PL/SQL Procedure

```
PROCEDURE update_code_assignment(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_code_assignment_rec    IN          CODE_ASSIGNMENT_REC_TYPE,
```

```

    p_object_version_number      IN OUT      NUMBER,
    x_return_status              OUT         VARCHAR2,
    x_msg_count                  OUT         NUMBER,
    x_msg_data                   OUT         VARCHAR2
)

```

### Java Method

```

public static void updateCodeAssignment(
    OracleConnection_connection,
    String
    CodeAssignmentRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
    p_init_msg_list,
    p_code_assignment_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data
) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Code Assignment API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
code_assignment_id	IN	NUMBER	Yes	Validation: Validated against HZ_CODE_ASSIGNMENTS.CODE_ASSIGNMENT_ID
owner_table_name	IN	VARCHAR2	No	
owner_table_id	IN	NUMBER	No	
class_category	IN	VARCHAR2	No	
class_code	IN	VARCHAR2	No	
primary_flag	IN	VARCHAR2	No	
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source.
start_date_active	IN	DATE	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
end_date_active	IN	DATE	No	Validation: Updateable if it is null
status	IN	VARCHAR2	No	Validation: Validated against AR_LOOKUP type CODE_STATUS
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing records</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record from hz_code_assignments</li> <li>■ Returns new value after update</li> </ul>
rank	IN	NUMBER	No	Comment: Provides the ability to rank classes if multiple classes have been assigned to a party.

## Create Class Category Use API

### Description

This routine is used to create a Class Category Use. The API creates a record in the HZ\_CLASS\_CATEGORY\_USES table. The classification model is an open structure, the HZ\_CLASS\_CATEGORY\_USES table indicates which tables or subsets of tables, use which classifications. The HZ\_CLASS\_CATEGORY\_USES table stores information about the tables which are going to use a particular class category. The ADDITIONAL\_WHERE\_CLAUSE is the filter for the subsets of tables. For example the SIC 1987 class category can be used to classify the parties, which have the party type Organization. The COLUMN\_NAME column holds the value of the column of the classified table in the OWNER\_TABLE column that is used as ID column for class code assignment.

**PL/SQL Procedure**

```

PROCEDURE create_class_category_use (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_class_category_use_rec IN          CLASS_CATEGORY_USE_REC_TYPE,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

**Java Method**

```

public static void createClassCategoryUse(
    OracleConnection_connection,
    String p_init_msg_list,
    ClassCategoryUseRec p_class_category_use_rec,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Create Class Category Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES
owner_table	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Validated against FND lookup values where Lookup type = CODE_ASSIGN_OWNER_TABLE</li> <li>■ The combination (class_category, owner_table) must be unique.</li> </ul>
column_name	IN	VARCHAR2	No	
additional_where_clause	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated

## Update Class Category Use API

### Description

This routine is used to update a Class Category Use. The API updates a record in the HZ\_CLASS\_CATEGORY\_USES table.

### PL/SQL Procedure

```
PROCEDURE update_class_category_use (
    p_init_msg_list          IN          VARCHAR2:=FND_API.G_FALSE,
    p_class_category_use_rec IN          CLASS_CATEGORY_USE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
)
```

### Java Method

```
public static void updateClassCategoryUse(
    OracleConnection_connection,
    String p_init_msg_list,
    ClassCategoryUseRec p_class_category_use_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Class Category Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES
owner_table	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>Validated against FND lookup type 'CODE_ASSIGN_OWNER_TABLE'</li> <li>The combination (class_category, owner_table) must be unique.</li> </ul>
column_name	IN	VARCHAR2	No	
additional_where_clause	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>Mandatory attribute</li> <li>Validated against value in the database for the existing records</li> </ul> Comment: <ul style="list-style-type: none"> <li>Pass the current object_version_number of the record from hz_class_category_uses<sup>2</sup></li> <li>Returns new value after update</li> </ul>

## Is Valid Category API

### Description

This function determines if an ID can be assigned to a class\_category and owner\_table. The function returns *T* if an ID can be assigned to the given class\_category, otherwise the function returns *F*.

**PL/SQL Function**

```

FUNCTION is_valid_category(
    p_owner_table          IN          VARCHAR2,
    p_class_category       IN          VARCHAR2,
    p_id                   IN          NUMBER,
)

RETURN VARCHAR2

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Is Valid Category API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_class_category	IN	VARCHAR2	Yes	
p_owner_table	IN	VARCHAR2	Yes	
p_id	IN	VARCHAR2	Yes	



---

---

## Contact Preference API Use

This chapter provides information about the Contact Preference application programming interfaces.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Contact Preference APIs

**PL/SQL Package Name: HZ\_CONTACT\_PREFERENCE\_V2PUB**

**Java Class Name: HzContactPreferenceV2Pub**

### PL/SQL Record Structure for Contact Preference

TYPE contact_preference_rec_type	IS RECORD (
contact_preference_id	NUMBER,
contact_level_table	VARCHAR2(30),
contact_level_table_id	NUMBER,
contact_type	VARCHAR2(30),
preference_code	VARCHAR2(30),
preference_topic_type	VARCHAR2(30),
preference_topic_type_id	NUMBER,
preference_topic_type_code	VARCHAR2(30),
preference_start_date	DATE,
preference_end_date	DATE,
preference_start_time_hr	NUMBER,
preference_end_time_hr	NUMBER,
preference_start_time_mi	NUMBER,
preference_end_time_mi	NUMBER,
max_no_of_interactions	NUMBER,
max_no_of_interact_uom_code	VARCHAR2(30),
requested_by	VARCHAR2(30),
reason_code	VARCHAR2(30),
status	VARCHAR2(1),
created_by_module	VARCHAR2(150),
application_id	NUMBER
)	)

### Java Inner Class for Contact Preference

```
public static class ContactPreferenceRec {
    public BigDecimal    contact_preference_id;
    public String        contact_level_table;
    public BigDecimal    contact_level_table_id;
    public String        contact_type;
    public String        preference_code;
    public String        preference_topic_type;
    public BigDecimal    preference_topic_type_id;
    public String        preference_topic_type_code;
}
```

```

public java.sql.Timestamp      preference_start_date;
public java.sql.Timestamp      preference_end_date;
public BigDecimal              preference_start_time_hr;
public BigDecimal              preference_end_time_hr;
public BigDecimal              preference_start_time_mi;
public BigDecimal              preference_end_time_mi;
public BigDecimal              max_no_of_interactions;
public String                  max_no_of_interact_uom_code;
public String                  requested_by;
public String                  reason_code;
public String                  status;
public String                  created_by_module;
public BigDecimal              application_id;

public ContactPreferenceRec();
public ContactPreferenceRec(boolean __RosettaUseGMISSValues);
}

```

## Create Contact Preference API

### Description

This routine creates a Contact Preference. The API creates a record in the HZ\_CONTACT\_PREFERENCES table. Contact preference can be created for the Party, Party Site or Contact Point.

### PL/SQL Procedure

```

PROCEDURE create_contact_preference (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_preference_rec IN          CONTACT_PREFERENCE_REC_TYPE,
    x_contact_preference_id  OUT          NUMBER,
    x_return_status          OUT          VARCHAR2,
    x_msg_count              OUT          NUMBER,
    x_msg_data               OUT          VARCHAR2
)

```

### Java Method

```

public static void createContactPreference(
    OracleConnection_connection,
    String                  p_init_msg_list,
    ContactPreferenceRec    p_contact_preference_rec,
    BigDecimal [ ]          x_contact_preference_id,
    String [ ]              x_return_status,
    BigDecimal [ ]          x_msg_count,
    String [ ]              x_msg_data
)

```

) throws SQLException;

### Parameter Description and Validation

The following table lists information about the parameters in the Create Contact Preference API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
contact_preference_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
contact_level_table	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Non updateable</li> <li>■ Validated against AR lookup type SUBJECT_TABLE</li> </ul>
contact_level_table_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Not updateable</li> <li>■ Is the foreign key of contact_level_table</li> </ul>
contact_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory Attribute.</li> <li>■ Non-updateable</li> <li>■ contact_type is lookup code in lookup type CONTACT_TYPE</li> </ul>
preference_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Should be validated against the PREFERENCE_CODE lookup type</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_topic_type	IN	VARCHAR2	N	Validation: <ul style="list-style-type: none"> <li>▪ Should be validated against the PREFERENCE_TOPIC_TYPE lookup type.</li> <li>▪ The lookup contains the following lookup_codes against which the PREFERENCE_TOPIC_TYPE will be validated:</li> </ul> TABLES: <ul style="list-style-type: none"> <li>▪ AMS_SOURCE_CODES,</li> <li>▪ AS_INTEREST_TYPES_B,</li> <li>▪ AS_INTEREST_CODES_B</li> <li>▪ LOOKUP_TYPE: CONTACT_USAGE</li> </ul>
preference_topic_type_id	IN	NUMBER	N	Validation : Valid if the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values: <ul style="list-style-type: none"> <li>▪ AMS_SOURCE_CODES</li> <li>▪ AS_INTEREST_TYPES_B</li> <li>▪ AS_INTEREST_CODES_B</li> </ul> The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected PREFERENCE_TOPIC_TYPE.
preference_topic_type_code	IN	VARCHAR2	N	Validation: The PREFERENCE_TOPIC_TYPE_CODE attribute is populated if the CONTACT_USAGE attribute is selected as the PREFERENCE_TOPIC_TYPE attribute. A validation is performed to select the code only from the lookup type that you select.
preference_start_date	IN	DATE	Y	Validation: Mandatory attribute.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_end_date	IN	DATE	N	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ Cannot be updated to a day before the sysdate</li> <li>■ PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE</li> <li>■ If the STATUS column in the HZ_CONTACT_POINTS table is set to a value other than A for Active (such as I for Inactive, M for Merged, of D for deleted), then PREFERENCE_END_DATE is not passed in these situations, it should default to the system date. If a value other than the system date is passed, it should fail.</li> </ul>
preference_start_time_hr	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_end_time_hr	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_start_time_mi	IN	NUMBER	N	Validation: In 0 to 59 minute format  PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI
preference_end_time_mi	IN	NUMBER	N	Validation: In 0 to 59 minute format  PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI
max_no_of_interactions	IN	NUMBER	N	Validation: none
max_no_of_interact_uom_code	IN	VARCHAR2	N	Validation: The MAX_NO_OF_INTERACT_UOM_CODE column should be validated against the new MAX_NO_OF_INTERACT_UOM_CODE lookup.
requested_by	IN	VARCHAR2	Y	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Should be validated against the REQUESTED_BY lookup type</li> </ul>
reason_code	IN	VARCHAR2	N	Validation : Validated against the REASON_CODE lookup type.
status	IN	VARCHAR2	N	Validation :  Should be validated against the CODE_STATUS lookup type.  The PREFERENCE_END_DATE attribute should be set to the sysdate when STATUS has a value other than A.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
created_by_module	IN	VARCHAR2	Y	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated.
application_id	IN	NUMBER	N	Comment: Text to indicate application from which creation of record is initiated.
x_contact_preference_id	OUT	NUMBER	N	Comment: Returns contact_preference_id of the record created.

### Other validations

- When a contact point is created, PREFERENCE\_START\_DATE is defaulted to the system date if a value is not passed and the PREFERENCE\_CODE should be set to "Do" in HZ\_CONTACT\_PREFERENCES unless the party explicitly opts-out (in that case the PREFERENCE\_CODE should be set to "Do Not" and the PREFERENCE\_START\_DATE should be set to the system date if no date has been specified by the party)
- If a value is passed for MAX\_NO\_OF\_INTERACT\_UOM\_CODE then the PREFERENCE\_START\_DATE should have a value. If a value is not passed for PREFERENCE\_START\_DATE it should default to the system date.
- Duplication check. uniquely identify a contact preference to prevent duplication record to be created in HZ\_CONTACT\_PREFERENCES by checking the follow columns:
  - contact\_level\_table
  - contact\_level\_table\_id
  - contact\_type
  - preference\_topic\_type
  - preference\_topic\_type\_id
  - preference\_topic\_type\_code
- When the CONTACT\_LEVEL\_TABLE attribute is HZ\_CONTACT\_POINTS, the CONTACT\_TYPE cannot be *MAIL* or *VISIT*.
- CONTACT\_TYPE lookup should be validated against the HZ\_CONTACT\_POINTS.CONTACT\_POINT\_TYPE column based on the value passed in HZ\_

CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE. So if the value passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE is HZ\_CONTACT\_POINTS and the HZ\_CONTACT\_LEVEL\_TABLE\_ID signifies that the contact point is an email address, then the values that will be allowed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE will be 'EMAIL' only.

- The following table gives the values that can be passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE based on the HZ\_CONTACT\_POINT.CONTACT\_POINT\_TYPE, HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE and HZ\_CONTACT\_PREFERENCES/CONTACT\_LEVEL\_TABLE\_ID

<b>CONTACT_TYPE allowed</b>	<b>when CONTACT_POINT_TYPE is</b>	<b>and CONTACT_LEVEL_TABLE is</b>
CALL	PHONE	HZ_CONTACT_POINTS
MAIL	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
SMS	SMS	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
VISIT	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
EMAIL	EMAIL	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
FAX	FAX	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
TELEX	TLX	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES

<b>CONTACT_TYPE</b> allowed	<b>when CONTACT_POINT_TYPE</b> is	<b>and CONTACT_LEVEL_TABLE</b> is
EDI	EDI	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES

## Update Contact Preference API

### Description

This routine updates a Contact Preference. The API updates a record in the HZ\_CONTACT\_PREFERENCES table for Party, Party Site or Contact Point.

### PL/SQL Procedure

```
PROCEDURE update_contact_preference (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_contact_preference_rec IN          CONTACT_PREFERENCE_REC_TYPE,
    p_object_version_number IN OUT      NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)
)
```

### Java Method

```
public static void updateContactPreference(
    OracleConnection connection,
    String p_init_msg_list,
    ContactPreferenceRec p_contact_preference_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Contact Preference API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
contact_preference_id	IN	NUMBER	Yes	Validation: Valid contact_preference_id should be passed in  Comment: Pass contact_preference_id from hz_contact_preferences table
contact_level_table	IN	VARCHAR2	No	Validation: Non updateable
contact_level_table_id	IN	NUMBER	No	Validation: Non updateable
contact_type	IN	VARCHAR2	No	Validation: Non updateable
preference_code	IN	VARCHAR2	No	Validation: Should be validated against the PREFERENCE_CODE AR lookup type
preference_topic_type	IN	VARCHAR2	No	Validation:  Should be validated against the PREFERENCE_TOPIC_TYPE lookup type.  The lookup contains the following lookup_codes against which the PREFERENCE_TOPIC_TYPE will be validated <b>TABLES</b> <ul style="list-style-type: none"> <li>■ AMS_SOURCE_CODES</li> <li>■ AS_INTEREST_TYPES_B</li> <li>■ AS_INTEREST_CODES_B</li> <li>■ LOOKUP_TYPE</li> </ul>
preference_topic_type_id	IN	NUMBER	No	Validation:  If the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values: <ul style="list-style-type: none"> <li>■ AMS_SOURCE_CODES</li> <li>■ AS_INTEREST_TYPES_B</li> <li>■ AS_INTEREST_CODES_B</li> </ul> The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected in the attribute PREFERENCE_TOPIC_TYPE.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_topic_ type_code	IN	VARCHAR2	No	Validation: The PREFERENCE_TOPIC_ TYPE_CODE column is populated if the CONTACT_ USAGE attribute is selected as the PREFERENCE_TOPIC_TYPE attribute. Validation is performed to select the code only from the lookup type that the user selects.
preference_start_date	IN	DATE	No	Validation: The value of the PREFERENCE_ END_DATE attribute should be greater than or equal to the value of the PREFERENCE_START_ DATE attribute.
preference_end_date	IN	DATE	No	Validation: <ul style="list-style-type: none"> <li>■ PREFERENCE_END_DATE can not update to a day before sysdate</li> <li>■ PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_ START_DATE,</li> </ul>
preference_start_ time_hr	IN	NUMBER	No	Validation: In 0 to 24 hour format PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI
preference_end_time_ hr	IN	NUMBER	No	Validation: In 0 to 24 hour format PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_start_time_mi	IN	NUMBER	No	Validation: In 0 to 59 minute format  PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI
preference_end_time_mi	IN	NUMBER	No	Validation: In 0 to 59 minute format  PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_ TIME_MI: PREFERENCE_ START_TIME_MI
max_no_of_interactions	IN	NUMBER	No	Validation: none
max_no_of_interact_uom_code	IN	VARCHAR2	No	Validation: The MAX_NO_OF_INTERACT_UOM_CODE column should be validated against the new MAX_NO_OF_INTERACT_UOM_CODE lookup.
requested_by	IN	VARCHAR2	No	Validation: REQUESTED_BY should be validated against the REQUESTED_BY AR lookup type
reason_code	IN	VARCHAR2	No	Validation: Should be validated against the REASON_CODE lookup type

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Should be validated against the CODE_STATUS lookup type</li> <li>■ The PREFERENCE_END_DATE attribute should be set to sysdate when the STATUS column has a value other than A</li> <li>■ Can not be set to NULL during update</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the contact preference record</li> <li>■ Return new value after update</li> </ul>

#### Other Validations

- When a contact point is created, PREFERENCE\_START\_DATE is defaulted to the system date if a value is not passed and the PREFERENCE\_CODE should be set to "Do" in HZ\_CONTACT\_PREFERENCES unless the party explicitly opts-out (in that case the PREFERENCE\_CODE should be set to "Do Not" and the PREFERENCE\_START\_DATE should be set to the system date if no date has been specified by the party)
- If the STATUS column in HZ\_CONTACT\_POINTS is set to a value other than "A" for "Active such as "I" for Inactive, "M" for Merged, or "D" for Deleted, then PREFERENCE\_END\_DATE should be set. If a value for PREFERENCE\_

END\_DATE is not passed in these situations, it should default to the system date. If a value other than the system date is passed, it should fail.

- If a value is passed for MAX\_NO\_OF\_INTERACT\_UOM\_CODE then the PREFERENCE\_START\_DATE should have a value. If a value is not passed for PREFERENCE\_START\_DATE it should default to the system date.
- Duplication check. uniquely identify a contact preference to prevent duplication record to be created in HZ\_CONTACT\_PREFERENCES by checking the follow columns:
  - contact\_level\_table
  - contact\_level\_table\_id
  - contact\_type
  - preference\_topic\_type
  - preference\_topic\_type\_id
  - preference\_topic\_type\_code
- When the CONTACT\_LEVEL\_TABLE attribute is HZ\_CONTACT\_POINTS, the CONTACT\_TYPE cannot be *MAIL* or *VISIT*.
- CONTACT\_TYPE lookup should be validated against the HZ\_CONTACT\_POINTS.CONTACT\_POINT\_TYPE column based on the value passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE. So if the value passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE is HZ\_CONTACT\_POINTS and the HZ\_CONTACT\_LEVEL\_TABLE\_ID signifies that the contact point is an email address, then the values that will be allowed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE will be 'EMAIL' only.
- The following table gives the values that can be passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE based on the HZ\_CONTACT\_POINT.CONTACT\_POINT\_TYPE, HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE and HZ\_CONTACT\_PREFERENCES/CONTACT\_LEVEL\_TABLE\_ID

<b>CONTACT_TYPE allowed</b>	<b>when CONTACT_POINT_TYPE is</b>	<b>and CONTACT_LEVEL_TABLE is</b>
CALL	PHONE	HZ_CONTACT_POINTS
MAIL	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES

<b>CONTACT_TYPE</b> allowed	<b>when CONTACT_POINT_TYPE is</b>	<b>and CONTACT_LEVEL_TABLE is</b>
SMS	SMS	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
VISIT	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
EMAIL	EMAIL	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
FAX	FAX	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
TELEX	TLX	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES
EDI	EDI	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES

---

---

## Customer Account API Use

This chapter provides information about the Customer Account application programming interfaces.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Customer Account APIs

**PL/SQL Package Name: HZ\_CUST\_ACCOUNT\_V2PUB**

**Java Class Name: HzCustAccountV2Pub**

### PL/SQL Record Structure for Customer Account

```
TYPE cust_account_rec_type IS RECORD (  
    cust_account_id          NUMBER,  
    account_number          VARCHAR2 (30) ,  
    attribute_category       VARCHAR2 (30) ,  
    attribute1               VARCHAR2 (150) ,  
    attribute2               VARCHAR2 (150) ,  
    attribute3               VARCHAR2 (150) ,  
    attribute4               VARCHAR2 (150) ,  
    attribute5               VARCHAR2 (150) ,  
    attribute6               VARCHAR2 (150) ,  
    attribute7               VARCHAR2 (150) ,  
    attribute8               VARCHAR2 (150) ,  
    attribute9               VARCHAR2 (150) ,  
    attribute10              VARCHAR2 (150) ,  
    attribute11              VARCHAR2 (150) ,  
    attribute12              VARCHAR2 (150) ,  
    attribute13              VARCHAR2 (150) ,  
    attribute14              VARCHAR2 (150) ,  
    attribute15              VARCHAR2 (150) ,  
    attribute16              VARCHAR2 (150) ,  
    attribute17              VARCHAR2 (150) ,  
    attribute18              VARCHAR2 (150) ,  
    attribute19              VARCHAR2 (150) ,  
    attribute20              VARCHAR2 (150) ,  
    global_attribute_category VARCHAR2 (30) ,  
    global_attribute1        VARCHAR2 (150) ,  
    global_attribute2        VARCHAR2 (150) ,  
    global_attribute3        VARCHAR2 (150) ,  
    global_attribute4        VARCHAR2 (150) ,  
    global_attribute5        VARCHAR2 (150) ,  
    global_attribute6        VARCHAR2 (150) ,  
    global_attribute7        VARCHAR2 (150) ,  
    global_attribute8        VARCHAR2 (150) ,  
    global_attribute9        VARCHAR2 (150) ,  
    global_attribute10       VARCHAR2 (150) ,
```

---

global_attribute11	VARCHAR2 (150) ,
global_attribute12	VARCHAR2 (150) ,
global_attribute13	VARCHAR2 (150) ,
global_attribute14	VARCHAR2 (150) ,
global_attribute15	VARCHAR2 (150) ,
global_attribute16	VARCHAR2 (150) ,
global_attribute17	VARCHAR2 (150) ,
global_attribute18	VARCHAR2 (150) ,
global_attribute19	VARCHAR2 (150) ,
global_attribute20	VARCHAR2 (150) ,
orig_system_reference	VARCHAR2 (240) ,
status	VARCHAR2 (1) ,
customer_type	VARCHAR2 (30) ,
customer_class_code	VARCHAR2 (30) ,
primary_salesrep_id	NUMBER ,
sales_channel_code	VARCHAR2 (30) ,
order_type_id	NUMBER ,
price_list_id	NUMBER ,
tax_code	VARCHAR2 (50) ,
fob_point	VARCHAR2 (30) ,
freight_term	VARCHAR2 (30) ,
ship_via	VARCHAR2 (25) ,
warehouse_id	NUMBER ,
tax_header_level_flag	VARCHAR2 (1) ,
tax_rounding_rule	VARCHAR2 (30) ,
coterminate_day_month	VARCHAR2 (6) ,
primary_specialist_id	NUMBER ,
secondary_specialist_id	NUMBER ,
account_liable_flag	VARCHAR2 (1) ,
current_balance	NUMBER ,
account_established_date	DATE ,
account_termination_date	DATE ,
account_activation_date	DATE ,
department	VARCHAR2 (30) ,
held_bill_expiration_date	DATE ,
hold_bill_flag	VARCHAR2 (1) ,
realtime_rate_flag	VARCHAR2 (1) ,
acct_life_cycle_status	VARCHAR2 (30) ,
account_name	VARCHAR2 (240) ,
deposit_refund_method	VARCHAR2 (20) ,
dormant_account_flag	VARCHAR2 (1) ,
npa_number	VARCHAR2 (60) ,
suspension_date	DATE ,
source_code	VARCHAR2 (150) ,
comments	VARCHAR2 (240) ,

```

dates_negative_tolerance          NUMBER,
dates_positive_tolerance          NUMBER,
date_type_preference              VARCHAR2(20),
over_shipment_tolerance           NUMBER,
under_shipment_tolerance          NUMBER,
over_return_tolerance             NUMBER,
under_return_tolerance            NUMBER,
item_cross_ref_pref              VARCHAR2(30),
ship_sets_include_lines_flag      VARCHAR2(1),
arrivalsets_include_lines_flag    VARCHAR2(1),
sched_date_push_flag             VARCHAR2(1),
invoice_quantity_rule            VARCHAR2(30),
status_update_date               DATE,
autopay_flag                     VARCHAR2(1),
notify_flag                      VARCHAR2(1),
last_batch_id                    NUMBER,
selling_party_id                 NUMBER,
created_by_module                VARCHAR2(150),
application_id                   NUMBER
)

```

## PL/SQL Record Type for Customer Account Relationship

```

TYPE cust_acct_relate_rec_type    IS RECORD (
  cust_account_id                 NUMBER,
  related_cust_account_id         NUMBER,
  relationship_type               VARCHAR2(30),
  comments                        VARCHAR2(240),
  attribute_category              VARCHAR2(30),
  attribute1                      VARCHAR2(150),
  attribute2                      VARCHAR2(150),
  attribute3                      VARCHAR2(150),
  attribute4                      VARCHAR2(150),
  attribute5                      VARCHAR2(150),
  attribute6                      VARCHAR2(150),
  attribute7                      VARCHAR2(150),
  attribute8                      VARCHAR2(150),
  attribute9                      VARCHAR2(150),
  attribute10                     VARCHAR2(150),
  customer_reciprocal_flag        VARCHAR2(1),
  status                          VARCHAR2(1),
  attribute11                     VARCHAR2(150),
  attribute12                     VARCHAR2(150),
  attribute13                     VARCHAR2(150),
  attribute14                     VARCHAR2(150),

```

```
attribute15                VARCHAR2(150),
bill_to_flag               VARCHAR2(1),
ship_to_flag               VARCHAR2(1),
created_by_module          VARCHAR2(150),
application_id             NUMBER
)
```

## Java Inner Class for Customer Account

```
public static class CustAccountRec {
    public BigDecimal      cust_account_id;
    public String          account_number;
    public String          attribute_category;
    public String          attribute1;
    public String          attribute2;
    public String          attribute3;
    public String          attribute4;
    public String          attribute5;
    public String          attribute6;
    public String          attribute7;
    public String          attribute8;
    public String          attribute9;
    public String          attribute10;
    public String          attribute11;
    public String          attribute12;
    public String          attribute13;
    public String          attribute14;
    public String          attribute15;
    public String          attribute16;
    public String          attribute17;
    public String          attribute18;
    public String          attribute19;
    public String          attribute20;
    public String          global_attribute_category;
    public String          global_attribute1;
    public String          global_attribute2;
    public String          global_attribute3;
    public String          global_attribute4;
    public String          global_attribute5;
    public String          global_attribute6;
    public String          global_attribute7;
    public String          global_attribute8;
    public String          global_attribute9;
    public String          global_attribute10;
    public String          global_attribute11;
}
```

```
public String          global_attribute12;
public String          global_attribute13;
public String          global_attribute14;
public String          global_attribute15;
public String          global_attribute16;
public String          global_attribute17;
public String          global_attribute18;
public String          global_attribute19;
public String          global_attribute20;
public String          orig_system_reference;
public String          status;
public String          customer_type;
public String          customer_class_code;
public BigDecimal      primary_salesrep_id;
public String          sales_channel_code;
public BigDecimal      order_type_id;
public BigDecimal      price_list_id;
public String          tax_code;
public String          fob_point;
public String          freight_term;
public String          ship_via;
public BigDecimal      warehouse_id;
public String          tax_header_level_flag;
public String          tax_rounding_rule;
public String          coterminate_day_month;
public BigDecimal      primary_specialist_id;
public BigDecimal      secondary_specialist_id;
public String          account_liable_flag;
public BigDecimal      current_balance;
public java.sql.Timestamp account_established_date;
public java.sql.Timestamp account_termination_date;
public java.sql.Timestamp account_activation_date;
public String          department;
public java.sql.Timestamp held_bill_expiration_date;
public String          hold_bill_flag;
public String          realtime_rate_flag;
public String          acct_life_cycle_status;
public String          account_name;
public String          deposit_refund_method;
public String          dormant_account_flag;
public String          npa_number;
public java.sql.Timestamp suspension_date;
public String          source_code;
public String          comments;
public BigDecimal      dates_negative_tolerance;
```

```

    public BigDecimal          dates_positive_tolerance;
    public String              date_type_preference;
    public BigDecimal          over_shipment_tolerance;
    public BigDecimal          under_shipment_tolerance;
    public BigDecimal          over_return_tolerance;
    public BigDecimal          under_return_tolerance;
    public String              item_cross_ref_pref;
    public String              ship_sets_include_lines_flag;
    public String              arrivalsets_include_lines_flag;
    public String              sched_date_push_flag;
    public String              invoice_quantity_rule;
    public java.sql.Timestamp  status_update_date;
    public String              autopay_flag;
    public String              notify_flag;
    public BigDecimal          last_batch_id;
    public BigDecimal          selling_party_id;
    public String              created_by_module;
    public BigDecimal          application_id;

    public CustAccountRec();
    public CustAccountRec(boolean __RosettaUseGMISSValues);
}

```

## Java Inner Class for Customer Account Relationship

```

public static class CustAcctRelateRec {
    public BigDecimal          cust_account_id;
    public BigDecimal          related_cust_account_id;
    public String              relationship_type;
    public String              comments;
    public String              attribute_category;
    public String              attribute1;
    public String              attribute2;
    public String              attribute3;
    public String              attribute4;
    public String              attribute5;
    public String              attribute6;
    public String              attribute7;
    public String              attribute8;
    public String              attribute9;
    public String              attribute10;
    public String              customer_reciprocal_flag;
    public String              status;
    public String              attribute11;
    public String              attribute12;
}

```

```
public String          attributel3;
public String          attributel4;
public String          attributel5;
public String          bill_to_flag;
public String          ship_to_flag;
public String          created_by_module;
public BigDecimal      application_id;

public CustAcctRelateRec();
public CustAcctRelateRec(boolean __RosettaUseGMISValues);
}
```

## Create Customer Account API (Person or Organization)

### Description

This routine is used to create a Customer Account. The API creates a record in the HZ\_CUST\_ACCOUNTS table for party type Person or Organization. Account can be created for an existing party by passing party\_id of the party. Alternatively, this routine creates a new party and an account for the party. Customer profile record in the HZ\_CUSTOMER\_PROFILES table can also be created while calling this routine based on value passed in p\_customer\_profile\_rec. The routine is overloaded for Person and Organization.

### PL/SQL Procedure for Person Account

```
PROCEDURE create_cust_account (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_account_rec       IN          CUST_ACCOUNT_REC_TYPE,
    p_person_rec             IN          HZ_PARTY_V2PUB.PERSON_REC_TYPE,
    p_customer_profile_rec   IN          HZ_CUSTOMER_PROFILE
                                     V2PUB.CUSTOMER_PROFILE_REC_
                                     TYPE,
    p_create_profile_amt     IN          VARCHAR2:= FND_API.G_TRUE,
    x_cust_account_id        OUT         NUMBER,
    x_account_number         OUT         VARCHAR2,
    x_party_id               OUT         NUMBER,
    x_party_number           OUT         VARCHAR2,
    x_profile_id             OUT         NUMBER,
    x_return_status          OUT         VARCHAR2,
    x_msg_count              OUT         NUMBER,
    x_msg_data               OUT         VARCHAR2
)
```

---



---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to FND\_API.G\_TRUE, profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---



---

### PL/SQL Procedure for Organization Account

```

PROCEDURE create_cust_account (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_account_rec       IN          CUST_ACCOUNT_REC_TYPE,
    p_organization_rec       IN          HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE,
    p_customer_profile_rec   IN          HZ_CUSTOMER_PROFILE_V2PUB.
                                        CUSTOMER_PROFILE_REC_TYPE,
    p_create_profile_amt     IN          VARCHAR2:= FND_API.G_TRUE,
    x_cust_account_id        OUT         NUMBER,
    x_account_number         OUT         VARCHAR2,
    x_party_id               OUT         NUMBER,
    x_party_number           OUT         VARCHAR2,
    x_profile_id             OUT         NUMBER,
    x_return_status          OUT         VARCHAR2,
    x_msg_count              OUT         NUMBER,
    x_msg_data               OUT         VARCHAR2
)

```

---



---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to FND\_API.G\_TRUE, profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---



---

### Java Method for Person Account

```

public static void createCustAccount
    (OracleConnection connection,
     String p_init_msg_list,
     CustAccountRec p_cust_account_rec,
     HzPartyV2Pub.PersonRec p_person_rec,
     HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec,
     String p_create_profile_amt,

```

```

        BigDecimal [ ]          x_cust_account_id,
        String [ ]             x_account_number,
        BigDecimal [ ]        x_party_id,
        String [ ]            x_party_number,
        BigDecimal [ ]        x_profile_id,
        String [ ]            x_return_status,
        BigDecimal [ ]        x_msg_count,
        String [ ]            x_msg_data
    ) throws SQLException;

```

---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to HzConstant.getGTrue(), profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---

#### Java Method for Organization Account

```

public static void createCustAccount(
    OracleConnection_connection,
    String                          p_init_msg_list,
    CustAccountRec                  p_cust_account_rec,
    HzPartyV2Pub.OrganizationRec    p_organization_rec,
    HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec,
    String                          p_create_profile_amt,
    BigDecimal [ ]                  x_cust_account_id,
    String [ ]                      x_account_number,
    BigDecimal [ ]                  x_party_id,
    String [ ]                      x_party_number,
    BigDecimal [ ]                  x_profile_id,
    String [ ]                      x_return_status,
    BigDecimal [ ]                  x_msg_count,
    String [ ]                      x_msg_data
) throws SQLException;

```

---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to HzConstant.getGTrue(), profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Account API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	N	Validation: unique if passed in, else generated from sequence
account_number	IN	VARCHAR2	Y	Validation: <ul style="list-style-type: none"> <li>■ Mandatory Attribute. If GENERATE_CUSTOMER_NUMBER of AR_SYSTEM_PARAMETERS is on, if user has passed in an account_number, error out</li> <li>■ account_number will be generated from sequence. If autonumbering is off, if user has not passed in value, error out</li> </ul>
attribute_category	IN	VARCHAR2	N	
attribute1	IN	VARCHAR2	N	
attribute2	IN	VARCHAR2	N	
attribute3	IN	VARCHAR2	N	
attribute4	IN	VARCHAR2	N	
attribute5	IN	VARCHAR2	N	
attribute6	IN	VARCHAR2	N	
attribute7	IN	VARCHAR2	N	
attribute8	IN	VARCHAR2	N	
attribute9	IN	VARCHAR2	N	
attribute10	IN	VARCHAR2	N	
attribute11	IN	VARCHAR2	N	
attribute12	IN	VARCHAR2	N	
attribute13	IN	VARCHAR2	N	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute14	IN	VARCHAR2	N	
attribute15	IN	VARCHAR2	N	
attribute16	IN	VARCHAR2	N	
attribute17	IN	VARCHAR2	N	
attribute18	IN	VARCHAR2	N	
attribute19	IN	VARCHAR2	N	
attribute20	IN	VARCHAR2	N	
global_attribute_category	IN	VARCHAR2	N	
global_attribute1	IN	VARCHAR2	N	
global_attribute2	IN	VARCHAR2	N	
global_attribute3	IN	VARCHAR2	N	
global_attribute4	IN	VARCHAR2	N	
global_attribute5	IN	VARCHAR2	N	
global_attribute6	IN	VARCHAR2	N	
global_attribute7	IN	VARCHAR2	N	
global_attribute8	IN	VARCHAR2	N	
global_attribute9	IN	VARCHAR2	N	
global_attribute10	IN	VARCHAR2	N	
global_attribute11	IN	VARCHAR2	N	
global_attribute12	IN	VARCHAR2	N	
global_attribute13	IN	VARCHAR2	N	
global_attribute14	IN	VARCHAR2	N	
global_attribute15	IN	VARCHAR2	N	
global_attribute16	IN	VARCHAR2	N	
global_attribute17	IN	VARCHAR2	N	
global_attribute18	IN	VARCHAR2	N	
global_attribute19	IN	VARCHAR2	N	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute20	IN	VARCHAR2	N	
orig_system_reference	IN	VARCHAR2	N	Validation: unique if passed in Default: cust_account_id
status	IN	VARCHAR2	N	Validation: <ul style="list-style-type: none"> <li>▪ status is lookup code in lookup type CODE_STATUS</li> <li>▪ status cannot be set to null during update. It is defaulted to 'A' if user does not pass any value</li> </ul>
customer_type	IN	VARCHAR2	N	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE
customer_class_code	IN	VARCHAR2	N	Validation: Validated against AR lookup type CUSTOMER CLASS
primary_salesrep_id	IN	NUMBER	N	Validation: Must be valid salesrep_id from RA_SALESREPS table.
sales_channel_code	IN	VARCHAR2	N	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups
order_type_id	IN	NUMBER	N	Validation: Must be a valid order_type_id from the OE_ORDER_TYPES_V.
price_list_id	IN	NUMBER	N	Validation: Must be a valid price_list_id from SO_PRICE_LISTS table.
tax_code	IN	VARCHAR2	N	Validation: Must be a valid tax_code from the AR_VAT_TAX table.
fob_point	IN	VARCHAR2	N	Validation: Validated against AR lookup type FOB.
freight_term	IN	VARCHAR2	N	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_partial	IN	VARCHAR2	N	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR2	N	Validation: ship_via is foreign key to oe_ship_methods_v and can be passed only in single org case
warehouse_id	IN	NUMBER	N	Validation: Must be valid organization_id from the ORG_ORGANIZATION_DEFINITIONS table.
tax_header_level_flag	IN	VARCHAR2	N	Validation: tax_header_level_flag is lookup code in lookup type YES/NO. It is defaulted to 'N' if user does not pass value.
tax_rounding_rule	IN	VARCHAR2	N	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
coterminate_day_month	IN	VARCHAR2	N	
primary_specialist_id	IN	NUMBER	N	Validation: primary_specialist_id is foreign key to per_all_people_f
secondary_specialist_id	IN	NUMBER	N	Validation: secondary_specialist_id is foreign key to per_all_people_f
account_liable_flag	IN	VARCHAR2	N	Validation: account_liable_flag is lookup code in lookup type YES/NO. It is defaulted to 'N' if user does not pass value.
current_balance	IN	NUMBER	N	
account_established_date	IN	DATE	N	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
account_termination_date	IN	DATE	N	Validation: <ul style="list-style-type: none"> <li>■ account_termination_date should be greater than account_established_date</li> <li>■ account_termination_date should be greater than account_activation_date</li> </ul>
account_activation_date	IN	DATE	N	Validation: account_activation_date should be greater than account_established_date
department	IN	VARCHAR2	N	
held_bill_expiration_date	IN	DATE	N	
hold_bill_flag	IN	VARCHAR2	N	Validation: hold_bill_flag is lookup code in lookup type YES/NO Default: 'N'
realtime_rate_flag	IN	VARCHAR2	N	
acct_life_cycle_status	IN	VARCHAR2	N	
account_name	IN	VARCHAR2	N	
deposit_refund_method	IN	VARCHAR2	N	
dormant_account_flag	IN	VARCHAR2	N	Validation: dormant_account_flag is lookup code in lookup type YES/NO Default: 'N'
npa_number	IN	VARCHAR2	N	
suspension_date	IN	DATE	N	
source_code	IN	VARCHAR2	N	
comments	IN	VARCHAR2	N	
dates_negative_tolerance	IN	NUMBER	N	
dates_positive_tolerance	IN	NUMBER	N	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
date_type_preference	IN	VARCHAR2	N	Validation: Validated against OE lookup type REQUEST_DATE_TYPE
over_shipment_tolerance	IN	NUMBER	N	
under_shipment_tolerance	IN	NUMBER	N	
over_return_tolerance	IN	NUMBER	N	
under_return_tolerance	IN	NUMBER	N	
item_cross_ref_pref	IN	VARCHAR2	N	Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYPES
ship_sets_include_lines_flag	IN	VARCHAR2	N	Validation: <ul style="list-style-type: none"> <li>■ ship_sets_include_lines_flag is lookup code in lookup type YES/NO</li> <li>■ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>■ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul> Default: N
arrivalsets_include_lines_flag	IN	VARCHAR2	N	Validation: <ul style="list-style-type: none"> <li>■ arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>■ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>■ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
sched_date_push_flag	IN	VARCHAR2	N	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR2	N	Validated against OE lookup type INVOICE_BASIS.
pricing_event	IN	VARCHAR2	N	Comment: This attribute is no longer used.
status_update_date	IN	DATE	N	
autopay_flag	IN	VARCHAR2	N	Validation: autopay_flag is lookup code in lookup type YES/NO
notify_flag	IN	VARCHAR2	N	Validation: notify_flag is lookup code in lookup type YES/NO
last_batch_id	IN	NUMBER	N	
selling_party_id	IN	NUMBER	N	Validation: selling_party_id is foreign key of HZ_PARTIES
created_by_module	IN	VARCHAR2	Y	Validation: Mandatory Attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	N	Comment: Text to indicate application from which creation of record is initiated
x_cust_account_id	OUT	NUMBER	N	Comment: Returns cust_account_id of the account record create
x_account_number	OUT	NUMBER	N	Comment: Returns account_number of the account record created
x_party_id	OUT	NUMBER	N	Comment: Returns party_id of the organization or person party created
x_party_number	OUT	NUMBER	N	Comment: Returns party_number of the organization or person created

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_profile_id	OUT	NUMBER	N	Comment: Returns profile_id of the organization or person profile created

### Other validations

- If party referenced by party\_id user passes through p\_person\_rec.party\_rec or p\_organization\_rec.party\_rec exists, we will create only account for this existing party. However, if party does not exist or user does not pass party\_id (i.e. party\_id is null or FND\_API.G\_MISS\_NUM), we will create both party and account. If party has to be created, the validations on p\_person\_rec and p\_organization\_rec are same as those in create\_person, create\_organization in hz\_party\_v2pub.
- Customer profile is mandatory for an account. If user does not pass profile\_class\_name, we will create a customer profile based on default profile class, which should have ID 0 and in active status. The validations on p\_customer\_profile\_rec are same as those in hz\_customer\_profile\_v2pub.create\_customer\_profile.
- If p\_create\_profile\_amt is FND\_API.G\_TRUE, we will create customer profile amount when we create customer profile. These customer profile amounts have defaulted value from profile class amounts of the same profile class.

## Update Customer Account API

### Description

This routine is used to update a Customer Account. The API updates a record in the HZ\_CUST\_ACCOUNTS table. The account could belong to a party of type Person or Organization. The same routine updates all types of accounts whether it belongs to a person or an organization.

### PL/SQL Procedure

```
PROCEDURE update_cust_account (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_account_rec      IN          CUST_ACCOUNT_REC_TYPE,
    p_object_version_number IN OUT    NUMBER,
    x_return_status        OUT         VARCHAR2,
    x_msg_count            OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

**Java Method**

```

public static void updateCustAccount(
    OracleConnection_connection,
    String                                p_init_msg_list,
    CustAccountRec                        p_cust_account_rec,
    BigDecimal [ ]                       p_object_version_number,
    String [ ]                            x_return_status,
    BigDecimal [ ]                       x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Update Customer Account API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	Yes	Validation: valid cust_account_id should be passed in Comment: Pass cust_account_id from hz_cust_accounts table
account_number	IN	VARCHAR2	No	Validation: Not updateable
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: Cannot be set to null during update
customer_type	IN	VARCHAR2	No	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE
customer_class_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type CUSTOMER_CLASS
primary_salesrep_id	IN	NUMBER	No	Validation: Must be valid salesrep_id from RA_SALESREPS.
sales_channel_code	IN	VARCHAR2	No	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups
order_type_id	IN	NUMBER	No	Validation: Must be valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	IN	NUMBER	No	Validation: Must be valid price_list_id from SO_PRICE_LISTS.
tax_code	IN	VARCHAR2	No	Validation: Must be valid tax_code from AR_VAT_TAX.
fob_point	IN	VARCHAR2	No	Validated against AR lookup type FOB.
freight_term	IN	VARCHAR2	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups
ship_partial	IN	VARCHAR2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
ship_via	IN	VARCHAR2	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be passed only in single org case
warehouse_id	IN	NUMBER	No	Validation: Must be valid organization_id from ORG_ORGANIZATION_DEFINITIONS.
tax_header_level_flag	IN	VARCHAR2	No	Validation: tax_header_level_flag is lookup code in lookup type YES/NO
tax_rounding_rule	IN	VARCHAR2	No	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
coterminate_day_month	IN	VARCHAR2	No	
primary_specialist_id	IN	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f
secondary_specialist_id	IN	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f
account_liable_flag	IN	VARCHAR2	No	Validation: account_liable_flag is lookup code in lookup type YES/NO
current_balance	IN	NUMBER	No	
account_established_date	IN	DATE	No	
account_termination_date	IN	DATE	No	Validation: <ul style="list-style-type: none"> <li>■ account_termination_date should be greater than account_establish</li> <li>■ account_termination_date should be greater than account_activation_date</li> </ul>
account_activation_date	IN	DATE	No	Validation: account_activation_date should be greater than account_established_date

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
department	IN	VARCHAR2	No	
held_bill_expiration_date	IN	DATE	No	
hold_bill_flag	IN	VARCHAR2	No	Validation: hold_bill_flag is lookup code in lookup type YES/NO
realtime_rate_flag	IN	VARCHAR2	No	
acct_life_cycle_status	IN	VARCHAR2	No	
account_name	IN	VARCHAR2	No	
deposit_refund_method	IN	VARCHAR2	No	
dormant_account_flag	IN	VARCHAR2	No	Validation: dormant_account_flag is lookup code in lookup type YES/NO
npa_number	IN	VARCHAR2	No	
suspension_date	IN	DATE	No	
source_code	IN	VARCHAR2	No	
comments	IN	VARCHAR2	No	
dates_negative_tolerance	IN	NUMBER	No	
dates_positive_tolerance	IN	NUMBER	No	
date_type_preference	IN	VARCHAR2	No	Validation: Validated against OE lookup type REQUEST_DATE_TYPE.
over_shipment_tolerance	IN	NUMBER	No	
under_shipment_tolerance	IN	NUMBER	No	
over_return_tolerance	IN	NUMBER	No	
under_return_tolerance	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
item_cross_ref_pref	IN	VARCHAR2	No	Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYPES.
ship_sets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	IN	VARCHAR2	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR2	No	Validation: Validated against OE lookup type INVOICE_BASIS.
pricing_event	IN	VARCHAR2	No	Comment: This attribute is no longer used.
status_update_date	IN	DATE	No	
autopay_flag	IN	VARCHAR2	No	Validation: autopay_flag is lookup code in lookup type YES/NO

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
notify_flag	IN	VARCHAR2	No	Validation: notify_flag is lookup code in lookup type YES/NO
last_batch_id	IN	NUMBER	No	
selling_party_id	IN	NUMBER	No	Validation: selling_party_id should point to a organization party
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record.</li> </ul> Comment <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the customer account record</li> <li>■ Return new value after update.</li> </ul>

## Create Customer Account Relationship API

### Description

This routine is used to create a Customer Account Relationship. The API creates a record in the HZ\_CUST\_ACCT\_RELATE table. You can relate two different customer accounts in this process. This is different from Relationship API that has been discussed earlier.

### PL/SQL Procedure

```

PROCEDURE create_cust_acct_relate (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_acct_relate_rec  IN          CUST_ACCT_RELATE_REC_TYPE,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

**Java Method**

```

public static void createCustAcctRelate(
    OracleConnection_connection,
    String                                p_init_msg_list,
    CustAcctRelateRec                    p_cust_acct_relate_rec,
    String [ ]                            x_return_status,
    BigDecimal [ ]                        x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Create Customer Account Relationship API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	Yes	Validation: Mandatory attribute Comment: Pass cust_account_id hz_cust_accounts
related_cust_account_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ elated_cust_account_id is foreign key of hz_cust_accounts</li> </ul>
relationship_type	IN	VARCHAR2	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE
comments	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
customer_reciprocal_flag	IN	VARCHAR2	No	Validation: customer_reciprocal_flag is lookup code in lookup type YES/NO Default: N
status	IN	VARCHAR2	No	Validation: Status is lookup code in lookup type CODE_STATUS Default: A
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
bill_to_flag	IN	VARCHAR2	No	
ship_to_flag	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated

### Other Validations

The combination of cust\_account\_id and related\_cust\_account\_id should be unique in an organization.

## Update Customer Account Relationship API

### Description

This routine is used to update a Customer Account Relationship. The API updates a in the HZ\_CUST\_ACCT\_RELATE table.

### PL/SQL Procedure

```
PROCEDURE update_cust_acct_relate (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_acct_relate_rec  IN          CUST_ACCT_RELATE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

### Java Method

```
public static void updateCustAcctRelate(
    OracleConnection_connection,
    String                    p_init_msg_list,
    CustAcctRelateRec        p_cust_acct_relate_rec,
    BigDecimal [ ]           p_object_version_number,
    String [ ]               x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]               x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Customer Account Relationship API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	Yes	Validation: cust_account_id is foreign key of hz_cust_accounts
related_cust_account_id	IN	NUMBER	Yes	Validation: related_cust_account_id is foreign key of hz_cust_accounts

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_type	IN	VARCHAR2	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE
comments	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
customer_reciprocal_flag	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Status cannot be set to null during update</li> <li>■ Status is lookup code in lookup type CODE_STATUS</li> </ul>
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
bill_to_flag	IN	VARCHAR2	No	
ship_to_flag	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>■ Mandatory attribute</li><li>■ Validated against value in the database for the existing record</li></ul> Comment: <ul style="list-style-type: none"><li>■ Pass the current object_version_number of the cust account relate record</li><li>■ Return new value after update</li></ul>

**Other Validations**

The combination of cust\_account\_id and related\_cust\_account\_id should be unique in an organization.

---

---

## Customer Account Site API Use

This chapter provides information about the Customer Account Site application programming interfaces.

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Customer Account Site APIs

**PL/SQL Package Name: HZ\_CUST\_ACCOUNT\_SITE\_V2PUB**

**Java Class Name: HzCustAccountSiteV2Pub**

### PL/SQL Record Structure for Customer Account Site

TYPE cust_acct_site_rec_type	IS RECORD (
cust_acct_site_id	NUMBER,
cust_account_id	NUMBER,
party_site_id	NUMBER,
attribute_category	VARCHAR2(30),
attribute1	VARCHAR2(150),
attribute2	VARCHAR2(150),
attribute3	VARCHAR2(150),
attribute4	VARCHAR2(150),
attribute5	VARCHAR2(150),
attribute6	VARCHAR2(150),
attribute7	VARCHAR2(150),
attribute8	VARCHAR2(150),
attribute9	VARCHAR2(150),
attribute10	VARCHAR2(150),
attribute11	VARCHAR2(150),
attribute12	VARCHAR2(150),
attribute13	VARCHAR2(150),
attribute14	VARCHAR2(150),
attribute15	VARCHAR2(150),
attribute16	VARCHAR2(150),
attribute17	VARCHAR2(150),
attribute18	VARCHAR2(150),
attribute19	VARCHAR2(150),
attribute20	VARCHAR2(150),
global_attribute_category	VARCHAR2(30),
global_attribute1	VARCHAR2(150),
global_attribute2	VARCHAR2(150),
global_attribute3	VARCHAR2(150),
global_attribute4	VARCHAR2(150),
global_attribute5	VARCHAR2(150),
global_attribute6	VARCHAR2(150),
global_attribute7	VARCHAR2(150),
global_attribute8	VARCHAR2(150),
global_attribute9	VARCHAR2(150),

```

global_attribute10          VARCHAR2(150),
global_attribute11          VARCHAR2(150),
global_attribute12          VARCHAR2(150),
global_attribute13          VARCHAR2(150),
global_attribute14          VARCHAR2(150),
global_attribute15          VARCHAR2(150),
global_attribute16          VARCHAR2(150),
global_attribute17          VARCHAR2(150),
global_attribute18          VARCHAR2(150),
global_attribute19          VARCHAR2(150),
global_attribute20          VARCHAR2(150),
orig_system_reference        VARCHAR2(240),
status                      VARCHAR2(1),
customer_category_code       VARCHAR2(30),
language                    VARCHAR2(4),
key_account_flag            VARCHAR2(1),
tp_header_id                NUMBER,
ece_tp_location_code         VARCHAR2(40),
primary_specialist_id        NUMBER,
secondary_specialist_id      NUMBER,
territory_id                 NUMBER,
territory                    VARCHAR2(30),
translated_customer_name     VARCHAR2(50),
created_by_module            VARCHAR2(150),
application_id               NUMBER
)

```

## PL/SQL Record Structure for Customer Account Site Use

```

TYPE cust_site_use_rec_type IS RECORD (
  site_use_id              NUMBER,
  cust_acct_site_id        NUMBER,
  site_use_code            VARCHAR2(30),
  primary_flag             VARCHAR2(1),
  status                   VARCHAR2(1),
  location                 VARCHAR2(40),
  bill_to_site_use_id      NUMBER,
  orig_system_reference     VARCHAR2(240),
  sic_code                 VARCHAR2(30),
  payment_term_id          NUMBER,
  gsa_indicator            VARCHAR2(1),
  ship_via                 VARCHAR2(25),
  fob_point                VARCHAR2(30),
  order_type_id            NUMBER,
  price_list_id            NUMBER,

```

freight_term	VARCHAR2 (30) ,
warehouse_id	NUMBER,
territory_id	NUMBER,
attribute_category	VARCHAR2 (30) ,
attribute1	VARCHAR2 (150) ,
attribute2	VARCHAR2 (150) ,
attribute3	VARCHAR2 (150) ,
attribute4	VARCHAR2 (150) ,
attribute5	VARCHAR2 (150) ,
attribute6	VARCHAR2 (150) ,
attribute7	VARCHAR2 (150) ,
attribute8	VARCHAR2 (150) ,
attribute9	VARCHAR2 (150) ,
attribute10	VARCHAR2 (150) ,
tax_reference	VARCHAR2 (50) ,
sort_priority	NUMBER,
tax_code	VARCHAR2 (50) ,
attribute11	VARCHAR2 (150) ,
attribute12	VARCHAR2 (150) ,
attribute13	VARCHAR2 (150) ,
attribute14	VARCHAR2 (150) ,
attribute15	VARCHAR2 (150) ,
attribute16	VARCHAR2 (150) ,
attribute17	VARCHAR2 (150) ,
attribute18	VARCHAR2 (150) ,
attribute19	VARCHAR2 (150) ,
attribute20	VARCHAR2 (150) ,
attribute21	VARCHAR2 (150) ,
attribute22	VARCHAR2 (150) ,
attribute23	VARCHAR2 (150) ,
attribute24	VARCHAR2 (150) ,
attribute25	VARCHAR2 (150) ,
demand_class_code	VARCHAR2 (30) ,
tax_header_level_flag	VARCHAR2 (1) ,
tax_rounding_rule	VARCHAR2 (30) ,
global_attribute1	VARCHAR2 (150) ,
global_attribute2	VARCHAR2 (150) ,
global_attribute3	VARCHAR2 (150) ,
global_attribute4	VARCHAR2 (150) ,
global_attribute5	VARCHAR2 (150) ,
global_attribute6	VARCHAR2 (150) ,
global_attribute7	VARCHAR2 (150) ,
global_attribute8	VARCHAR2 (150) ,
global_attribute9	VARCHAR2 (150) ,
global_attribute10	VARCHAR2 (150) ,

```

global_attribute11          VARCHAR2 (150) ,
global_attribute12          VARCHAR2 (150) ,
global_attribute13          VARCHAR2 (150) ,
global_attribute14          VARCHAR2 (150) ,
global_attribute15          VARCHAR2 (150) ,
global_attribute16          VARCHAR2 (150) ,
global_attribute17          VARCHAR2 (150) ,
global_attribute18          VARCHAR2 (150) ,
global_attribute19          VARCHAR2 (150) ,
global_attribute20          VARCHAR2 (150) ,
global_attribute_category   VARCHAR2 (30) ,
primary_salesrep_id         NUMBER,
finchrg_receivables_trx_id NUMBER,
dates_negative_tolerance    NUMBER,
dates_positive_tolerance    NUMBER,
date_type_preference        VARCHAR2 (20) ,
over_shipment_tolerance     NUMBER,
under_shipment_tolerance    NUMBER,
item_cross_ref_pref         VARCHAR2 (30) ,
over_return_tolerance       NUMBER,
under_return_tolerance      NUMBER,
ship_sets_include_lines_flag VARCHAR2 (1) ,
arrivalsets_include_lines_flag VARCHAR2 (1) ,
sched_date_push_flag        VARCHAR2 (1) ,
invoice_quantity_rule       VARCHAR2 (30) ,
gl_id_rec                   NUMBER,
gl_id_rev                   NUMBER,
gl_id_tax                   NUMBER,
gl_id_freight               NUMBER,
gl_id_clearing               NUMBER,
gl_id_unbilled               NUMBER,
gl_id_unearned               NUMBER,
gl_id_unpaid_rec            NUMBER,
gl_id_remittance            NUMBER,
gl_id_factor                 NUMBER,
tax_classification           VARCHAR2 (30) ,
created_by_module            VARCHAR2 (150) ,
application_id               NUMBER
)

```

## Java Inner Class for Customer Account Site

```

public static class CustAcctSiteRec {
    public BigDecimal          cust_acct_site_id;
    public BigDecimal          cust_account_id;
}

```



```

        public String                customer_category_code;
        public String                language;
        public String                key_account_flag;
        public BigDecimal            tp_header_id;
        public String                ece_tp_location_code;
        public BigDecimal            primary_specialist_id;
        public BigDecimal            secondary_specialist_id;
        public BigDecimal            territory_id;
        public String                territory;
        public String                translated_customer_name;
        public String                created_by_module;
        public BigDecimal            application_id;

        public CustAcctSiteRec();
        public CustAcctSiteRec(boolean __RosettaUseGMISSValues);
    }
    public CustAcctSiteRec(boolean __RosettaUseGMISSValues);
}

```

## Java Inner Class for Customer Account Site Use

```

public static class CustSiteUseRec {
    public BigDecimal            site_use_id;
    public BigDecimal            cust_acct_site_id;
    public String                site_use_code;
    public String                primary_flag;
    public String                status;
    public String                location;
    public BigDecimal            bill_to_site_use_id;
    public String                orig_system_reference;
    public String                sic_code;
    public BigDecimal            payment_term_id;
    public String                gsa_indicator;
    public String                ship_via;
    public String                fob_point;
    public BigDecimal            order_type_id;
    public BigDecimal            price_list_id;
    public String                freight_term;
    public BigDecimal            warehouse_id;
    public BigDecimal            territory_id;
    public String                attribute_category;
    public String                attribute1;
    public String                attribute2;
    public String                attribute3;
    public String                attribute4;
}

```

```
public String          attribute5;
public String          attribute6;
public String          attribute7;
public String          attribute8;
public String          attribute9;
public String          attribute10;
public String          tax_reference;
public BigDecimal      sort_priority;
public String          tax_code;
public String          attribute11;
public String          attribute12;
public String          attribute13;
public String          attribute14;
public String          attribute15;
public String          attribute16;
public String          attribute17;
public String          attribute18;
public String          attribute19;
public String          attribute20;
public String          attribute21;
public String          attribute22;
public String          attribute23;
public String          attribute24;
public String          attribute25;
public String          demand_class_code;
public String          tax_header_level_flag;
public String          tax_rounding_rule;
public String          global_attribute1;
public String          global_attribute2;
public String          global_attribute3;
public String          global_attribute4;
public String          global_attribute5;
public String          global_attribute6;
public String          global_attribute7;
public String          global_attribute8;
public String          global_attribute9;
public String          global_attribute10;
public String          global_attribute11;
public String          global_attribute12;
public String          global_attribute13;
public String          global_attribute14;
public String          global_attribute15;
public String          global_attribute16;
public String          global_attribute17;
public String          global_attribute18;
```

```

    public String                global_attribute19;
    public String                global_attribute20;
    public String                global_attribute_category;
    public BigDecimal            primary_salesrep_id;
    public BigDecimal            finchrg_receivables_trx_id;
    public BigDecimal            dates_negative_tolerance;
    public BigDecimal            dates_positive_tolerance;
    public String                date_type_preference;
    public BigDecimal            over_shipment_tolerance;
    public BigDecimal            under_shipment_tolerance;
    public String                item_cross_ref_pref;
    public BigDecimal            over_return_tolerance;
    public BigDecimal            under_return_tolerance;
    public String                ship_sets_include_lines_flag;
    public String                arrivalsets_include_lines_flag;
    public String                sched_date_push_flag;
    public String                invoice_quantity_rule;
    public BigDecimal            gl_id_rec;
    public BigDecimal            gl_id_rev;
    public BigDecimal            gl_id_tax;
    public BigDecimal            gl_id_freight;
    public BigDecimal            gl_id_clearing;
    public BigDecimal            gl_id_unbilled;
    public BigDecimal            gl_id_uneared;
    public BigDecimal            gl_id_unpaid_rec;
    public BigDecimal            gl_id_remittance;
    public BigDecimal            gl_id_factor;
    public String                tax_classification;
    public String                created_by_module;
    public BigDecimal            application_id;

    public CustSiteUseRec();
    public CustSiteUseRec(boolean __RosettaUseGMISSValues);
}

```

## Create Customer Account Site API

### Description

This routine is used to create a Customer Account Site. The API creates a record in the HZ\_CUST\_ACCT\_SITES table. The customer account site is created using an existing customer account and an existing party site.

### PL/SQL Procedure

```
PROCEDURE create_cust_acct_site (
```

```

    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_acct_site_rec     IN          CUST_ACCT_SITE_REC_TYPE,
    x_cust_acct_site_id     OUT         NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

### Java Method

```

public static void createCustAcctSite(
    OracleConnection_connection,
    String                                p_init_msg_list,
    CustAcctSiteRec                       p_cust_acct_site_rec,
    BigDecimal [ ]                        x_cust_acct_site_id,
    String [ ]                             x_return_status,
    BigDecimal [ ]                        x_msg_count,
    String [ ]                             x_msg_data
) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Account Site API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_site_id	IN	NUMBER	Yes/No	Validation: unique if passed in, else generated from sequence
cust_account_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ cust_account_id is foreign key of hz_cust_accounts</li> </ul>
party_site_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ party_site_id is foreign key of hz_party_sites<sup>2</sup></li> <li>■ party_site_id must link to a location of content source type USER_ENTERED</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: unique if passed in Default: cust_acct_site_id
status	IN	VARCHAR2	No	Validation: status is lookup code in lookup type CODE_STATUS Default: A
customer_category_code	IN	VARCHAR2	No	Validation: customer_category_code is lookup code in lookup type ADDRESS_CATEGORY
language	IN	VARCHAR2	No	Validation: language is foreign key of fnd installed languages
key_account_flag	IN	VARCHAR2	No	
tp_header_id	IN	NUMBER	No	Validation: tp_header_id must be unique if pass in

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
ece_tp_location_code	IN	VARCHAR2	No	Validation: The ece_tp_location_code should be unique for a customer within the organization.
primary_specialist_id	IN	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f
secondary_specialist_id	IN	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f
territory_id	IN	NUMBER	No	
territory	IN	VARCHAR2	No	
translated_customer_name	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated
x_cust_acct_site_id	OUT	NUMBER	No	Comment: Returns cust_acct_site_id of the record created

### Other Validations

cust\_account\_id and party\_site\_id together should be unique for a particular organization.

## Update Customer Account Site API

### Description

This routine is used to update a Customer Account Site. The API updates a record in the HZ\_CUST\_ACCT\_SITES table.

### PL/SQL Procedure

```
PROCEDURE update_cust_acct_site (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_acct_site_rec    IN          CUST_ACCT_SITE_REC_TYPE,
```

```

    p_object_version_number      IN OUT    NUMBER,
    x_return_status              OUT       VARCHAR2,
    x_msg_count                  OUT       NUMBER,
    x_msg_data                   OUT       VARCHAR2
)

```

### Java Method

```

public static void updateCustAcctSite(
    OracleConnection_connection,
    String
    CustAcctSiteRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) throws SQLException;
    p_init_msg_list,
    p_cust_acct_site_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data

```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Customer Account Site Organization API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_site_id	IN	NUMBER	Yes	Validation: valid cust_acct_site_id must be passed in Comment: Pass cust_acct_site_id from table hz_cust_acct_sites
cust_account_id	IN	NUMBER	No	Validation: Not updateable
party_site_id	IN	NUMBER	No	Validation: Not updateable
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Status cannot be set to null during update.</li> <li>■ Status is lookup code in lookup type CODE_STATUS.</li> </ul>
customer_category_code	IN	VARCHAR2	No	Validation: customer_category_code is lookup code in lookup type ADDRESS_CATEGORY
Language	IN	VARCHAR2	No	Validation: language is foreign key of fnd installed languages
key_account_flag	IN	VARCHAR2	No	
tp_header_id	IN	NUMBER	No	Validation: tp_header_id must be unique
ece_tp_location_code	IN	VARCHAR2	No	
primary_specialist_id	IN	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f
secondary_specialist_id	IN	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f
territory_id	IN	NUMBER	No	
territory	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
translated_customer_name	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass current object_version_number of the record from hz_cust_acct_sites</li> <li>■ Return new value after update</li> </ul>

#### Other Validations

cust\_account\_id and party\_site\_id together should be unique for a particular organization.

## Create Customer Account Site Use API

#### Description

This routine is used to create a Customer Account Site Use. The API creates a record in the HZ\_CUST\_SITE\_USES table. Additionally profile information at site level can be created by this routine by passing proper value in p\_create\_profile.

#### PL/SQL Procedure

```

PROCEDURE create_cust_site_use (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_site_use_rec      IN          CUST_SITE_USE_REC_TYPE,
    p_customer_profile_rec   IN          HZ_CUSTOMER_PROFILE_V2PUB
                                     .CUSTOMER_PROFILE_REC_TYPE,
    p_create_profile         IN          VARCHAR2:= FND_API.G_TRUE,
    p_create_profile_amt     IN          VARCHAR2:= FND_API.G_TRUE,
    x_site_use_id           OUT         NUMBER,

```

x_return_status	OUT	VARCHAR2,
x_msg_count	OUT	NUMBER,
x_msg_data	OUT	VARCHAR2

)

---

---

**Note:** p\_create\_profile indicates whether to create customer profile for the site use being created. If value equals to FND\_API.G\_TRUE, a profile will be created. Similarly, p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to FND\_API.G\_TRUE, profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based. p\_create\_profile\_amt is processed only when p\_create\_profile equals to FND\_API.G\_TRUE.

---

---

### Java Method

```
public static void createCustSiteUse(
    OracleConnection_connection,
    String                                p_init_msg_list,
    CustSiteUseRec                        p_cust_site_use_rec,
    HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec,
    String                                p_create_profile,
    String                                p_create_profile_amt,
    BigDecimal [ ]                       x_site_use_id,
    String [ ]                            x_return_status,
    BigDecimal [ ]                       x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;
```

---

---

**Note:** p\_create\_profile indicates whether to create customer profile for the site use being created. If value equals to HzConstant.getGTrue(), a profile will be created. Similarly, p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to HzConstant.getGTrue(), profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based. p\_create\_profile\_amt is processed only when p\_create\_profile equals to HzConstant.getGTrue().

---

---

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Account Site Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
site_use_id	IN	NUMBER	Yes	Validation: unique if passed in, else generated from sequence
cust_acct_site_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ cust_account_id is foreign key to hz_cust_acct_sites</li> </ul>
site_use_code	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ site_use_code is lookup type YES/NO</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ primary_flag is lookup code in lookup type YES/NO</li> <li>▪ Only one primary is allowed for one site use type per account</li> </ul> Default: N
status	IN	VARCHAR2	No	Validation: status is lookup code in lookup type CODE_STATUS Default: A

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
location	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ It will be generated from sequence if user does not pass in and AUTO_SITE_NUMBERING in AR_SYSTEM_PARAMETERS is on</li> <li>■ location must be unique within a customer account/site_use_type</li> </ul>
bill_to_site_use_id	IN	NUMBER	No	
orig_system_reference	IN	VARCHAR2	No	Validation: site_use_id
sic_code	IN	VARCHAR2	No	
payment_term_id	IN	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS
gsa_indicator	IN	VARCHAR2	No	Validation: gsa_indicator is lookup code in lookup type YES/NO Default: N
ship_partial	IN	VARCHAR2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR2	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be used in both single and multi org case.
fob_point	IN	VARCHAR2	No	Validation: Validated against AR lookup type FOB.
order_type_id	IN	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	IN	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
freight_term	IN	VARCHAR2	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups
warehouse_id	IN	NUMBER	No	Validation: Valid organization_id from org_organization_definitions.
territory_id	IN	NUMBER	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	
sort_priority	IN	NUMBER	No	
tax_code	IN	VARCHAR2	No	Validation: Must be a valid tax_code from AR_VAT_TAX.
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
attribute25	IN	VARCHAR2	No	
demand_class_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type DEMAND_CLASS.
tax_header_level_flag	IN	VARCHAR2	No	
tax_rounding_rule	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_ category	IN	VARCHAR2	No	
primary_salesrep_id	IN	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_ trx_id	IN	NUMBER	No	Validation: Valid receivables_ trx_id from AR_RECEIVABLES_ TRX.
dates_negative_ tolerance	IN	NUMBER	No	
dates_positive_ tolerance	IN	NUMBER	No	
date_type_preference	IN	VARCHAR2	No	Validation: Validated against OE lookup type REQUEST_DATE_ TYPE.
over_shipment_ tolerance	IN	NUMBER	No	
under_shipment_ tolerance	IN	NUMBER	No	
item_cross_ref_pref	IN	VARCHAR2	No	Validation: Allowed values are INT, CUST, and cross_reference_ type value from MTL_CROSS_ REFERENCE_TYPES.
over_return_tolerance	IN	NUMBER	No	
under_return_ tolerance	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
ship_sets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>■ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>■ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul> Default: N
arrivalsets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>■ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>■ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul> Default: N
sched_date_push_flag	IN	VARCHAR2	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO Default: N
invoice_quantity_rule	IN	VARCHAR2	No	
pricing_event	IN	VARCHAR2	No	Comment: This attribute is no longer used.
gl_id_rec	IN	NUMBER	No	Validation: gl_id_rec is valid gl field
gl_id_rev	IN	NUMBER	No	Validation: gl_id_rev is valid gl field

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gl_id_tax	IN	NUMBER	No	Validation: gl_id_tax is valid gl field
gl_id_freight	IN	NUMBER	No	Validation: gl_id_freight is valid gl field
gl_id_clearing	IN	NUMBER	No	Validation: gl_id_clearing is valid gl field
gl_id_unbilled	IN	NUMBER	No	Validation: gl_id_unbilled is valid gl field
gl_id_unearned	IN	NUMBER	No	Validation: gl_id_unearned is valid gl field
gl_id_unpaid_rec	IN	NUMBER	No	Validation: gl_id_unpaid_rec is valid gl field
gl_id_remittance	IN	NUMBER	No	Validation: gl_id_remittance is valid gl field
gl_id_factor	IN	NUMBER	No	Validation: gl_id_factor is valid gl field
tax_classification	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
p_create_profile	IN	VARCHAR2	No	Validation: T or F Comment: If value T is passed, profile record for account site will be created.
p_create_profile_amt	IN	VARCHAR2	No	Validation: T or F Comment: If value T is passed, profile amount record for account site will be created.
x_site_use_id	OUT	NUMBER	No	Comment: Returns site_use_id of the record created

**Other Validations**

- When create cust site use, if the site use code is also in PARTY\_SITE\_USE\_CODE lookup type and the corresponding party site does not have such site use, the API internally creates party site use with the same site use code.
- A customer can have only one active DUN, STMTS, LEGAL site use
- cust\_acct\_site\_id and site\_use\_code together should be unique.
- For SHIP\_TO site use, bill\_to\_site\_use\_id should be a valid active BILL\_TO site use of one of an active account site of this account and its related account. For other type of site uses, this column should be null.
- If site\_use\_code is not 'BILL\_TO', then none of the gl\_id\_XXX fields should be populated.

**Update Customer Account Site Use API****Description**

This routine is used to update a Customer Account Site Use. The API updates a record in the HZ\_CUST\_SITE\_USES table.

**PL/SQL Procedure**

```
PROCEDURE update_cust_site_use (  
    p_init_msg_list                IN          VARCHAR2:= FND_API.G_FALSE,  
    p_cust_site_use_rec           IN          CUST_SITE_USE_REC_TYPE,  
    p_object_version_number       IN OUT    NUMBER,  
    x_return_status               OUT       VARCHAR2,  
    x_msg_count                   OUT       NUMBER,  
    x_msg_data                    OUT       VARCHAR2  
)
```

**Java Method**

```
public static void updateCustSiteUse(  
    OracleConnection_connection,  
    String                                p_init_msg_list,  
    CustSiteUseRec                        p_cust_site_use_rec,  
    BigDecimal [ ]                       p_object_version_number,  
    String [ ]                            x_return_status,  
    BigDecimal [ ]                       x_msg_count,  
    String [ ]                            x_msg_data  
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Customer Account Site Use API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
site_use_id	IN	NUMBER	Yes	Validation: valid site_use_id should be passed in  Comment: Pass the site_use_id from hz_cust_site_uses table
cust_acct_site_id	IN	NUMBER	No	Validation: Not updateable
site_use_code	IN	VARCHAR2	No	Validation: Not updateable
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ primary_flag cannot be set to null during update</li> <li>▪ primary_flag is lookup code in lookup type YES/NO<sup>2</sup></li> <li>▪ Only one primary is allowed for one site use type per account</li> </ul>
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ status cannot be set to null during update</li> <li>▪ status is lookup code in lookup type CODE_STATUS</li> </ul>
location	IN	VARCHAR2	No	Validation: Not updateable
bill_to_site_use_id	IN	NUMBER	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
sic_code	IN	VARCHAR2	No	
payment_term_id	IN	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS.
gsa_indicator	IN	VARCHAR2	No	Validation: gsa_indicator is lookup code in lookup type YES/NO

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
ship_partial	IN	VARCHAR2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR2	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be used in both single and multi org case.
fob_point	IN	VARCHAR2	No	Validation: Validated against AR lookup type FOB.
order_type_id	IN	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	IN	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.
freight_term	IN	VARCHAR2	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups
warehouse_id	IN	NUMBER	No	Validation: Valid organization_id from ORG_ORGANIZATION_DEFINITIONS.
territory_id	IN	NUMBER	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
tax_reference	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
sort_priority	IN	NUMBER	No	
tax_code	IN	VARCHAR2	No	Validation: Must be a valid tax_code from AR_VAT_TAX.
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
attribute25	IN	VARCHAR2	No	
demand_class_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type DEMAND_CLASS.
tax_header_level_flag	IN	VARCHAR2	No	
tax_rounding_rule	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_ category	IN	VARCHAR2	No	
primary_salesrep_id	IN	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_ trx_id	IN	NUMBER	No	Validation: Valid receivables_ trx_id from AR_RECEIVABLES_ TRX.
dates_negative_ tolerance	IN	NUMBER	No	
dates_positive_ tolerance	IN	NUMBER	No	
date_type_preference	IN	VARCHAR2	No	Validation: Validated against OE lookup type REQUEST_DATE_ TYPE.
over_shipment_ tolerance	IN	NUMBER	No	
under_shipment_ tolerance	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
item_cross_ref_pref	IN	VARCHAR2	No	Validation: Allowed values are INT, CUST, and cross_reference_type value from MTL_CROSS_REFERENCE_TYPES.
over_return_tolerance	IN	NUMBER	No	
under_return_tolerance	IN	NUMBER	No	
ship_sets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>▪ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>▪ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>▪ arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>▪ If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>▪ If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	IN	VARCHAR2	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR2	No	
pricing_event	IN	VARCHAR2	No	Comment: This attribute is no longer used.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gl_id_rec	IN	NUMBER	No	Validation: gl_id_rec is valid gl field
gl_id_rev	IN	NUMBER	No	Validation: gl_id_rev is valid gl field
gl_id_tax	IN	NUMBER	No	Validation: gl_id_tax is valid gl field
gl_id_freight	IN	NUMBER	No	Validation: gl_id_freight is valid gl field
gl_id_clearing	IN	NUMBER	No	Validation: gl_id_clearing is valid gl field
gl_id_unbilled	IN	NUMBER	No	Validation: gl_id_unbilled is valid gl field
gl_id_unearned	IN	NUMBER	No	Validation: gl_id_unearned is valid gl field
gl_id_unpaid_rec	IN	NUMBER	No	Validation: gl_id_unpaid_rec is valid gl field
gl_id_remittance	IN	NUMBER	No	Validation: gl_id_remittance is valid gl field
gl_id_factor	IN	NUMBER	No	Validation: gl_id_factor is valid gl field
tax_classification	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN/O UT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record.</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass current object_version_number of the record from hz_cust_site_uses</li> <li>■ Return new value after update</li> </ul>

#### Other Validations

cust\_acct\_site\_id and site\_use\_code together should be unique.



---

---

# Customer Profile and Customer Account Role API Use

This chapter provides information about the following Trading Community Architecture application programming interfaces:

- Customer Profile APIs
- Customer Account Role APIs

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Customer Profile APIs

**PL/SQL Package Name: HZ\_CUSTOMER\_PROFILE\_V2PUB**

**Java Class Name: HzCustomerProfileV2Pub**

### PL/SQL Record Structure for Customer Profile

TYPE customer_profile_rec_type	IS RECORD (
cust_account_profile_id	NUMBER,
cust_account_id	NUMBER,
status	VARCHAR2(1),
collector_id	NUMBER,
credit_analyst_id	NUMBER,
credit_checking	VARCHAR2(1),
next_credit_review_date	DATE,
tolerance	NUMBER,
discount_terms	VARCHAR2(1),
dunning_letters	VARCHAR2(1),
interest_charges	VARCHAR2(1),
send_statements	VARCHAR2(1),
credit_balance_statements	VARCHAR2(1),
credit_hold	VARCHAR2(1),
profile_class_id	NUMBER,
site_use_id	NUMBER,
credit_rating	VARCHAR2(30),
risk_code	VARCHAR2(30),
standard_terms	NUMBER,
override_terms	VARCHAR2(1),
dunning_letter_set_id	NUMBER,
interest_period_days	NUMBER,
payment_grace_days	NUMBER,
discount_grace_days	NUMBER,
statement_cycle_id	NUMBER,
account_status	VARCHAR2(30),
percent_collectable	NUMBER,
autocash_hierarchy_id	NUMBER,
attribute_category	VARCHAR2(30),
attribute1	VARCHAR2(150),
attribute2	VARCHAR2(150),
attribute3	VARCHAR2(150),
attribute4	VARCHAR2(150),
attribute5	VARCHAR2(150),

---

attribute6	VARCHAR2(150),
attribute7	VARCHAR2(150),
attribute8	VARCHAR2(150),
attribute9	VARCHAR2(150),
attribute10	VARCHAR2(150),
attribute11	VARCHAR2(150),
attribute12	VARCHAR2(150),
attribute13	VARCHAR2(150),
attribute14	VARCHAR2(150),
attribute15	VARCHAR2(150),
auto_rec_incl_disputed_flag	VARCHAR2(1),
tax_printing_option	VARCHAR2(30),
charge_on_finance_charge_flag	VARCHAR2(1),
grouping_rule_id	NUMBER,
clearing_days	NUMBER,
jgzz_attribute_category	VARCHAR2(30),
jgzz_attribute1	VARCHAR2(150),
jgzz_attribute2	VARCHAR2(150),
jgzz_attribute3	VARCHAR2(150),
jgzz_attribute4	VARCHAR2(150),
jgzz_attribute5	VARCHAR2(150),
jgzz_attribute6	VARCHAR2(150),
jgzz_attribute7	VARCHAR2(150),
jgzz_attribute8	VARCHAR2(150),
jgzz_attribute9	VARCHAR2(150),
jgzz_attribute10	VARCHAR2(150),
jgzz_attribute11	VARCHAR2(150),
jgzz_attribute12	VARCHAR2(150),
jgzz_attribute13	VARCHAR2(150),
jgzz_attribute14	VARCHAR2(150),
jgzz_attribute15	VARCHAR2(150),
global_attribute1	VARCHAR2(150),
global_attribute2	VARCHAR2(150),
global_attribute3	VARCHAR2(150),
global_attribute4	VARCHAR2(150),
global_attribute5	VARCHAR2(150),
global_attribute6	VARCHAR2(150),
global_attribute7	VARCHAR2(150),
global_attribute8	VARCHAR2(150),
global_attribute9	VARCHAR2(150),
global_attribute10	VARCHAR2(150),
global_attribute11	VARCHAR2(150),
global_attribute12	VARCHAR2(150),
global_attribute13	VARCHAR2(150),
global_attribute14	VARCHAR2(150),

```

global_attribute15          VARCHAR2 (150) ,
global_attribute16          VARCHAR2 (150) ,
global_attribute17          VARCHAR2 (150) ,
global_attribute18          VARCHAR2 (150) ,
global_attribute19          VARCHAR2 (150) ,
global_attribute20          VARCHAR2 (150) ,
global_attribute_category   VARCHAR2 (30) ,
cons_inv_flag               VARCHAR2 (1) ,
cons_inv_type               VARCHAR2 (30) ,
autocash_hierarchy_id_for_adr NUMBER ,
lockbox_matching_option     VARCHAR2 (30) ,
created_by_module           VARCHAR2 (150) ,
application_id              NUMBER
review_cycle                VARCHAR2 (30) ,
last_credit_review_date    DATE ,
party_id                    NUMBER
credit_classification        VARCHAR2 (30)
)

```

## PL/SQL Record Structure for Customer Profile Amount

```

TYPE cust_profile_amt_rec_type IS RECORD (
  cust_acct_profile_amt_id NUMBER ,
  cust_account_profile_id NUMBER ,
  currency_code           VARCHAR2 (15) ,
  trx_credit_limit        NUMBER ,
  overall_credit_limit    NUMBER ,
  min_dunning_amount      NUMBER ,
  min_dunning_invoice_amount NUMBER ,
  max_interest_charge     NUMBER ,
  min_statement_amount    NUMBER ,
  auto_rec_min_receipt_amount NUMBER ,
  interest_rate           NUMBER ,
  attribute_category      VARCHAR2 (30) ,
  attribute1              VARCHAR2 (150) ,
  attribute2              VARCHAR2 (150) ,
  attribute3              VARCHAR2 (150) ,
  attribute4              VARCHAR2 (150) ,
  attribute5              VARCHAR2 (150) ,
  attribute6              VARCHAR2 (150) ,
  attribute7              VARCHAR2 (150) ,
  attribute8              VARCHAR2 (150) ,
  attribute9              VARCHAR2 (150) ,
  attribute10             VARCHAR2 (150) ,
  attribute11             VARCHAR2 (150) ,

```

---

attribute12	VARCHAR2(150),
attribute13	VARCHAR2(150),
attribute14	VARCHAR2(150),
attribute15	VARCHAR2(150),
min_fc_balance_amount	NUMBER,
min_fc_invoice_amount	NUMBER,
cust_account_id	NUMBER,
site_use_id	NUMBER,
expiration_date	DATE,
jgzz_attribute_category	VARCHAR2(30),
jgzz_attribute1	VARCHAR2(150),
jgzz_attribute2	VARCHAR2(150),
jgzz_attribute3	VARCHAR2(150),
jgzz_attribute4	VARCHAR2(150),
jgzz_attribute5	VARCHAR2(150),
jgzz_attribute6	VARCHAR2(150),
jgzz_attribute7	VARCHAR2(150),
jgzz_attribute8	VARCHAR2(150),
jgzz_attribute9	VARCHAR2(150),
jgzz_attribute10	VARCHAR2(150),
jgzz_attribute11	VARCHAR2(150),
jgzz_attribute12	VARCHAR2(150),
jgzz_attribute13	VARCHAR2(150),
jgzz_attribute14	VARCHAR2(150),
jgzz_attribute15	VARCHAR2(150),
global_attribute1	VARCHAR2(150),
global_attribute2	VARCHAR2(150),
global_attribute3	VARCHAR2(150),
global_attribute4	VARCHAR2(150),
global_attribute5	VARCHAR2(150),
global_attribute6	VARCHAR2(150),
global_attribute7	VARCHAR2(150),
global_attribute8	VARCHAR2(150),
global_attribute9	VARCHAR2(150),
global_attribute10	VARCHAR2(150),
global_attribute11	VARCHAR2(150),
global_attribute12	VARCHAR2(150),
global_attribute13	VARCHAR2(150),
global_attribute14	VARCHAR2(150),
global_attribute15	VARCHAR2(150),
global_attribute16	VARCHAR2(150),
global_attribute17	VARCHAR2(150),
global_attribute18	VARCHAR2(150),
global_attribute19	VARCHAR2(150),
global_attribute20	VARCHAR2(150),

```
global_attribute_category      VARCHAR2(30),
created_by_module              VARCHAR2(150),
application_id                 NUMBER
)
```

## Java Inner Class for Customer Profile

```
public static class CustomerProfileRec {
    public BigDecimal          cust_account_profile_id;
    public BigDecimal          cust_account_id;
    public String              status;
    public BigDecimal          collector_id;
    public BigDecimal          credit_analyst_id;
    public String              credit_checking;
    public java.sql.Timestamp  next_credit_review_date;
    public BigDecimal          tolerance;
    public String              discount_terms;
    public String              dunning_letters;
    public String              interest_charges;
    public String              send_statements;
    public String              credit_balance_statements;
    public String              credit_hold;
    public BigDecimal          profile_class_id;
    public BigDecimal          site_use_id;
    public String              credit_rating;
    public String              risk_code;
    public BigDecimal          standard_terms;
    public String              override_terms;
    public BigDecimal          dunning_letter_set_id;
    public BigDecimal          interest_period_days;
    public BigDecimal          payment_grace_days;
    public BigDecimal          discount_grace_days;
    public BigDecimal          statement_cycle_id;
    public String              account_status;
    public BigDecimal          percent_collectable;
    public BigDecimal          autocash_hierarchy_id;
    public String              attribute_category;
    public String              attribute1;
    public String              attribute2;
    public String              attribute3;
    public String              attribute4;
    public String              attribute5;
    public String              attribute6;
    public String              attribute7;
    public String              attribute8;
}
```

```
public String attribute9;
public String attribute10;
public String attribute11;
public String attribute12;
public String attribute13;
public String attribute14;
public String attribute15;
public String auto_rec_incl_disputed_flag;
public String tax_printing_option;
public String charge_on_finance_charge_flag;
public BigDecimal grouping_rule_id;
public BigDecimal clearing_days;
public String jgzz_attribute_category;
public String jgzz_attribute1;
public String jgzz_attribute2;
public String jgzz_attribute3;
public String jgzz_attribute4;
public String jgzz_attribute5;
public String jgzz_attribute6;
public String jgzz_attribute7;
public String jgzz_attribute8;
public String jgzz_attribute9;
public String jgzz_attribute10;
public String jgzz_attribute11;
public String jgzz_attribute12;
public String jgzz_attribute13;
public String jgzz_attribute14;
public String jgzz_attribute15;
public String global_attribute1;
public String global_attribute2;
public String global_attribute3;
public String global_attribute4;
public String global_attribute5;
public String global_attribute6;
public String global_attribute7;
public String global_attribute8;
public String global_attribute9;
public String global_attribute10;
public String global_attribute11;
public String global_attribute12;
public String global_attribute13;
public String global_attribute14;
public String global_attribute15;
public String global_attribute16;
public String global_attribute17;
```

```
    public String                global_attribute18;
    public String                global_attribute19;
    public String                global_attribute20;
    public String                global_attribute_category;
    public String                cons_inv_flag;
    public String                cons_inv_type;
    public BigDecimal            autocash_hierarchy_id_for_adr;
    public String                lockbox_matching_option;
    public String                created_by_module;
    public BigDecimal            application_id;
    public String                review_cycle;
    public java.sql.Timestamp    last_credit_review_date;
    public BigDecimal            party_id;
    public String                credit_classification;

    public CustomerProfileRec();
    public CustomerProfileRec(boolean __RosettaUseGMISSValues);
}
```

## Java Inner Class for Customer Profile Amount

```
public static class CustProfileAmtRec {
    public BigDecimal            cust_acct_profile_amt_id;
    public BigDecimal            cust_account_profile_id;
    public String                currency_code;
    public BigDecimal            trx_credit_limit;
    public BigDecimal            overall_credit_limit;
    public BigDecimal            min_dunning_amount;
    public BigDecimal            min_dunning_invoice_amount;
    public BigDecimal            max_interest_charge;
    public BigDecimal            min_statement_amount;
    public BigDecimal            auto_rec_min_receipt_amount;
    public BigDecimal            interest_rate;
    public String                attribute_category;
    public String                attribute1;
    public String                attribute2;
    public String                attribute3;
    public String                attribute4;
    public String                attribute5;
    public String                attribute6;
    public String                attribute7;
    public String                attribute8;
    public String                attribute9;
    public String                attribute10;
    public String                attribute11;
}
```

```
public String          attribute12;
public String          attribute13;
public String          attribute14;
public String          attribute15;
public BigDecimal     min_fc_balance_amount;
public BigDecimal     min_fc_invoice_amount;
public BigDecimal     cust_account_id;
public BigDecimal     site_use_id;
public java.sql.Timestamp expiration_date;
public String          jgzz_attribute_category;
public String          jgzz_attribute1;
public String          jgzz_attribute2;
public String          jgzz_attribute3;
public String          jgzz_attribute4;
public String          jgzz_attribute5;
public String          jgzz_attribute6;
public String          jgzz_attribute7;
public String          jgzz_attribute8;
public String          jgzz_attribute9;
public String          jgzz_attribute10;
public String          jgzz_attribute11;
public String          jgzz_attribute12;
public String          jgzz_attribute13;
public String          jgzz_attribute14;
public String          jgzz_attribute15;
public String          global_attribute1;
public String          global_attribute2;
public String          global_attribute3;
public String          global_attribute4;
public String          global_attribute5;
public String          global_attribute6;
public String          global_attribute7;
public String          global_attribute8;
public String          global_attribute9;
public String          global_attribute10;
public String          global_attribute11;
public String          global_attribute12;
public String          global_attribute13;
public String          global_attribute14;
public String          global_attribute15;
public String          global_attribute16;
public String          global_attribute17;
public String          global_attribute18;
public String          global_attribute19;
public String          global_attribute20;
```

```
public String                global_attribute_category;
public String                created_by_module;
public BigDecimal            application_id;

public CustProfileAmtRec();
public CustProfileAmtRec(boolean __RosettaUseGMISSValues);
}
```

## Create Customer Profile API

### Description

This routine is used to create a Customer Profile. The API creates a record in the HZ\_CUSTOMER\_PROFILES table. The profile can be created at party level, at customer level, or at customer site level. It also creates profile amounts based on the value passed for p\_create\_profile\_amt.

### PL/SQL Procedure

```
PROCEDURE create_customer_profile (
    p_init_msg_list           IN          VARCHAR2:= FND_API.G_FALSE,
    p_customer_profile_rec    IN          CUSTOMER_PROFILE_REC_TYPE,
    p_create_profile_amt      IN          VARCHAR2:= FND_API.G_TRUE,
    x_cust_account_profile_id OUT         NUMBER,
    x_return_status           OUT         VARCHAR2,
    x_msg_count               OUT         NUMBER,
    x_msg_data                OUT         VARCHAR2
)
```

---

---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to FND\_API.G\_TRUE, profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---

---

### Java Method

```
public static void createCustomerProfile(
    OracleConnection_connection,
    String                p_init_msg_list,
    CustomerProfileRec    p_customer_profile_rec,
    String                p_create_profile_amt,
    BigDecimal [ ]       x_cust_account_profile_id,
    String [ ]            x_return_status,
```

```

        BigDecimal [ ]                x_msg_count,
        String [ ]                    x_msg_data
    ) throws SQLException;

```

---

**Note:** p\_create\_profile\_amt indicates whether to create profile amounts for the customer profile being created. If value equals to HzConstant.getGTrue(), profile amounts will be created by copying over the profile amounts for the profile class on which this customer profile is based.

---

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Profile API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_profile_id	IN	NUMBER	Yes/No	Validation Unique if passed in, else generated from sequence

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ Foreign key to hz_cust_accounts.cust_account_id</li> <li>■ One of the following 2 columns need to be not null : cust_account_id and party_id</li> <li>■ If cust_account_id is not null and party_id is null then: <ul style="list-style-type: none"> <li>* The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.</li> <li>*The party_id in the hz_parties table associated with this cust_account_id must be active.</li> </ul> </li> <li>■ If cust_account_id is null and party_id is not null then: <ul style="list-style-type: none"> <li>*The party_id is a foreign key to an active hz_parties.party_id.</li> <li>*The cust_account_id will be defaulted to (-1).</li> </ul> </li> <li>■ If cust_account_id is not null and party_id is not null then <ul style="list-style-type: none"> <li>*The party_id is a foreign key to an active hz_parties.party_id.</li> <li>*The party_id and the cust_account_id must be associated in hz_cust_accounts.</li> </ul> </li> <li>■ You can only have one customer profile for one account In this case you can find the party_id in the hz_customer_profiles.party_id column.</li> <li>■ You can only have one customer profile at party level for a party. In this case the cust_account_id will be defaulted to -1.</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
status	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type CODE_STATUS Default: A
collector_id	IN	NUMBER	No	Validation: foreign key to ar_collectors Default: it is defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)
credit_analyst_id	IN	NUMBER	No	
credit_checking	IN	VARCHAR2	No	Validation: It is lookup code in lookup type YES/NO Default: It is defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)
next_credit_review_date	IN	DATE	No	
tolerance	IN	NUMBER	No	Validation: Must be between -100 and 100. Default: Defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)
discount_terms	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is lookup code in lookup YES/NO.</li> <li>■ If discount_terms = Y, then discount_grace_days should be greater than or equal to 0.</li> <li>■ If discount_terms = N, then discount_grace_days should be null.</li> </ul> Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
dunning_letters	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ Dunning letters is lookup code in lookup type YES/NO.</li> <li>■ If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>■ If dunning_letters = N, then dunning_letter_set_id must be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
interest_charges	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ It is a lookup code in lookup type YES/NO.</li> <li>■ If interest_charges = Y, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>■ If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
send_statements	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ It is a lookup code in lookup type YES/NO.</li> <li>■ If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have value.</li> <li>■ If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
credit_balance_statements	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ It is a lookup code in lookup type YES/NO.</li> <li>■ Can be populated if send_statements is Y.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
credit_hold	IN	VARCHAR2	No	<p>Validation: It is a lookup code in lookup type YES/NO</p> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
profile_class_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ If passed in, profile_class_id should be positive.</li> <li>■ Foreign key to hz_cust_profile_classes and the corresponding profile class should be active</li> </ul> Default: defaulted to the corresponding value of DEFAULT_PROFILE_CLASS (PROFILE_CLASS_ID=0)
site_use_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ Foreign key to hz_cust_site_uses</li> <li>■ One site use can only have one profile.</li> <li>■ The customer which the site_use_id belongs to should have the same id as cust_account_id in this profile. In this case cust_account_id is mandatory.</li> </ul>
credit_rating	IN	VARCHAR2	No	Validation: Credit Rating is lookup code in AR lookup type CREDIT_RATING
risk_code	IN	VARCHAR2	No	Validation: Risk Code is lookup code in AR lookup type RISK_CODE
standard_terms	IN	NUMBER	No	Validation: Must be a valid term_id in RA_TERMS.
override_terms	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
dunning_letter_set_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ If dunning_letters = Y, then dunning_letter_set_id must have a value.</li> <li>■ If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>■ Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.</li> </ul>
interest_period_days	IN	NUMBER	No	Validation: Can be entered when interest_charges is Y and it is mandatory.
payment_grace_days	IN	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	IN	NUMBER	No	Validation: Can be populated only if discount_terms is Y.
statement_cycle_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ Must be a valid statment_cycle_id from AR_STATEMENT_CYCLES.</li> <li>■ Can be populated if send_statements is Y.</li> </ul>
account_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type ACCOUNT_STATUS.
percent_collectable	IN	NUMBER	No	Validation: Must be between 0 and 100.
autocash_hierarchy_id	IN	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
auto_rec_incl_disputed_flag	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO Default: Defaulted to the corresponding value of 'DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)
tax_printing_option	IN	VARCHAR2	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	IN	VARCHAR2	No	Validation: charge_on_finance_charge_flag is lookup code in lookup type YES/NO
grouping_rule_id	IN	NUMBER	No	Validation: Must be a valid grouping_rule_id from RA_GROUPING_RULES.
clearing_days	IN	NUMBER	No	Validation: Must be greater than zero.
jgzz_attribute_category	IN	VARCHAR2	No	
jgzz_attribute1	IN	VARCHAR2	No	
jgzz_attribute2	IN	VARCHAR2	No	
jgzz_attribute3	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
jpgzz_attribute4	IN	VARCHAR2	No	
jpgzz_attribute5	IN	VARCHAR2	No	
jpgzz_attribute6	IN	VARCHAR2	No	
jpgzz_attribute7	IN	VARCHAR2	No	
jpgzz_attribute8	IN	VARCHAR2	No	
jpgzz_attribute9	IN	VARCHAR2	No	
jpgzz_attribute10	IN	VARCHAR2	No	
jpgzz_attribute11	IN	VARCHAR2	No	
jpgzz_attribute12	IN	VARCHAR2	No	
jpgzz_attribute13	IN	VARCHAR2	No	
jpgzz_attribute14	IN	VARCHAR2	No	
jpgzz_attribute15	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
cons_inv_flag	IN	VARCHAR2	No	Validation: cons_inv_flag is lookup code in lookup type YES/NO
cons_inv_type	IN	VARCHAR2	No	
autocash_hierarchy_id_for_adr	IN	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
lockbox_matching_option	IN	VARCHAR2	No	Validation: Validated against AR lookup type ARLPLB_MATCHING_OPTION.
created_by_module	IN	VARCHAR2	Y	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
review_cycle	IN	VARCHAR2	No	Validation : Validated against AR lookup type PERIODIC_REVIEW_CYCLE.
last_review_date	IN	DATE	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>■ One of the following two columns need to be not null: cust_account_id and party_id.</li> <li>■ If cust_account_id is not null and party_id is null then this is a mandatory field.</li> </ul> <p>*The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.</p> <p>*The party_id in the hz_parties table associated with this cust_account_id must be active.</p> <ul style="list-style-type: none"> <li>■ If cust_account_id is null and party_id is not null then</li> </ul> <p>*The party_id is a foreign key to an active hz_parties.party_id.</p> <p>*The cust_account_id will be defaulted to (-1).</p> <ul style="list-style-type: none"> <li>■ If cust_account_id is not null and party_id is not null then</li> </ul> <p>*The party_id is a foreign key to an active hz_parties.party_id.</p> <p>*The party_id and the cust_account_id must be associated in hz_cust_accounts.</p> <ul style="list-style-type: none"> <li>■ You can only have one customer profile for one account. In this case you can find the party_id in the hz_customer_profiles.party_id column.</li> <li>■ You can only have one customer profile at party level for a party. In this case the cust_account_id will be defaulted to (-1).</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_create_profile_amt	IN	VARCHAR2	No	Validation: T or F  Comment: Indicates whether profile amounts will be created or not. Default value is FND_API.G_TRUE to create amounts.
x_cust_account_profile_id	OUT	NUMBER	No	Comment: Returns cust_account_profile_id for the record created
credit_classification	IN	VARCHAR2	No	Validation: This is a lookup code in lookup_type 'CREDIT_CLASSIFICATION'.  Default: NULL

### Other Validations

When creating customer profile, if profile class id is null or FND\_API.G\_MISS\_NUM, we create customer profile based on default profile class, which should have ID 0 and should be in active status. If profile class id is not null, we create customer profile based on this particular profile class. For those columns we can default from profile class (i.e. columns we have in both hz\_customer\_profiles and hz\_cust\_profile\_classes). If they are passed as null, we will default them from profile class; if they are passed as FND\_API.G\_MISS\_XXX which means user want to set them to null, we will set these columns to null. This rule also applies when we update customer profile and pass profile class id.

## Update Customer Profile API

### Description

This routine is used to update a Customer Profile. The API updates a record in the HZ\_CUSTOMER\_PROFILES table. The profile could be at the party level, the customer level, or the customer site level.

### PL/SQL Procedure

```
PROCEDURE update_customer_profile (
  p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
  p_customer_profile_rec  IN          CUSTOMER_PROFILE_REC_TYPE,
  p_object_version_number IN OUT     NUMBER,
  x_return_status         OUT         VARCHAR2,
  x_msg_count             OUT         NUMBER,
  x_msg_data              OUT         VARCHAR2
```

)

**Java Method**

```

public static void updateCustomerProfile(
    OracleConnection_connection,
    String p_init_msg_list,
    CustomerProfileRec p_customer_profile_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Update Customer Profile API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_profile_id	IN	NUMBER	Yes	Validation: valid cust_account_profile_id has to be passed in Comment: Pass cust_account_profile_id from table hz_customer_profiles
cust_account_id	IN	NUMBER	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type CODE_STATUS Default: A
collector_id	IN	NUMBER	No	Validation: Foreign key to ar_collectors
credit_analyst_id	IN	NUMBER	No	
credit_checking	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO
next_credit_review_date	IN	DATE	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
tolerance	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ Tolerance cannot be updated to null.</li> <li>■ Must be between -100 and 100.</li> </ul>
discount_terms	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is lookup code in lookup YES/NO.</li> <li>■ If discount_terms = Y, then discount_grace_days should be greater than or equal to 0.</li> <li>■ If discount_terms = N, then discount_grace_days should be null.</li> </ul>
dunning_letters	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is lookup code in lookup YES/NO.</li> <li>■ If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>■ If dunning_letters = N, then dunning_letter_set_id must be null.</li> </ul>
interest_charges	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is lookup code in lookup type YES/NO.</li> <li>■ If interest_charges = Y, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>■ If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
send_statements	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is lookup code in lookup type YES/NO.</li> <li>■ If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have a value.</li> <li>■ If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.</li> </ul>
credit_balance_statements	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ It is a lookup code in lookup type YES/NO.</li> <li>■ Can be populated if send_statements is Y.</li> </ul>
credit_hold	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO
profile_class_id	IN	NUMBER	No	Validation: Cannot be set to null
site_use_id	IN	NUMBER	No	Validation: Not updateable
credit_rating	IN	VARCHAR2	No	Validation: Credit Rating is lookup code in lookup type CREDIT_RATING
risk_code	IN	VARCHAR2	No	Validation: Risk Code is a lookup code in lookup type RISK_CODE
standard_terms	IN	NUMBER	No	Validation: Must be a valid term_id in RA_TERMS.
override_terms	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
dunning_letter_set_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>■ If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>■ Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.</li> </ul>
interest_period_days	IN	NUMBER	No	Validation: Must be entered when interest_charges is Y.
payment_grace_days	IN	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	IN	NUMBER	No	Validation: Can only be populated if the value of discount_terms is Y.
statement_cycle_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES.</li> <li>■ Can be populated if send_statements is Y.</li> </ul>
account_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type ACCOUNT_STATUS.
percent_collectable	IN	NUMBER	No	Validation: Must be between 0 and 100.
autocash_hierarchy_id	IN	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
auto_rec_incl_disputed_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Cannot be set to null</li> <li>■ It is a lookup code in lookup type YES/NO</li> </ul>
tax_printing_option	IN	VARCHAR2	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	IN	VARCHAR2	No	Validation: It is lookup code in lookup type YES/NO.
grouping_rule_id	IN	NUMBER	No	Validation: Must be a valid grouping_rule_id from RA_GROUPING_RULES.
clearing_days	IN	NUMBER	No	Validation: Must be greater than zero.
jgzz_attribute_category	IN	VARCHAR2	No	
jgzz_attribute1	IN	VARCHAR2	No	
jgzz_attribute2	IN	VARCHAR2	No	
jgzz_attribute3	IN	VARCHAR2	No	
jgzz_attribute4	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gzzz_attribute5	IN	VARCHAR2	No	
gzzz_attribute6	IN	VARCHAR2	No	
gzzz_attribute7	IN	VARCHAR2	No	
gzzz_attribute8	IN	VARCHAR2	No	
gzzz_attribute9	IN	VARCHAR2	No	
gzzz_attribute10	IN	VARCHAR2	No	
gzzz_attribute11	IN	VARCHAR2	No	
gzzz_attribute12	IN	VARCHAR2	No	
gzzz_attribute13	IN	VARCHAR2	No	
gzzz_attribute14	IN	VARCHAR2	No	
gzzz_attribute15	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
cons_inv_flag	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO
cons_inv_type	IN	VARCHAR2	No	
autocash_hierarchy_id_for_adr	IN	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
lockbox_matching_option	IN	VARCHAR2	No	Validation: Validated against AR lookup type ARLPLB_MATCHING_OPTION.
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass the current object_version_number of the record</li> <li>■ Return new value after update</li> </ul>
credit_classification	IN	VARCHAR2	No	Validation: Validated against AR lookup type 'CREDIT_CLASSIFICATION'

## Create Customer Profile Amount API

### Description

This routine is used to create Customer Profile Amount. The API creates record in the HZ\_CUST\_PROFILE\_AMTS table for a profile. You have to create a customer profile before you can create profile amount record.

### PL/SQL Procedure

```
PROCEDURE create_cust_profile_amt (  
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,  
    p_check_foreign_key     IN          VARCHAR2:= FND_API.G_TRUE,  
    p_cust_profile_amt_rec  IN          CUST_PROFILE_AMT_REC_TYPE,  
    x_cust_acct_profile_amt_id OUT      NUMBER,  
    x_return_status        OUT      VARCHAR2,  
    x_msg_count            OUT      NUMBER,  
    x_msg_data             OUT      VARCHAR2  
)
```

---

---

**Note:** p\_check\_foreign\_key indicates whether to do foreign key checking for the profile amount being created. If value equals to FND\_API.G\_TRUE, API will do foreign key checking on cust\_account\_id, cust\_account\_profile\_id and site\_use\_id. This was added for backward compatibility with customer form only. You should always set the parameter to FND\_API.G\_TRUE when you call API.

---

---

### Java Method

```
public static void createCustProfileAmt(  
    OracleConnection connection,  
    String p_init_msg_list,  
    String p_check_foreign_key,  
    CustProfileAmtRec p_cust_profile_amt_rec,  
    BigDecimal [ ] x_cust_acct_profile_amt_id,  
    String [ ] x_return_status,  
    BigDecimal [ ] x_msg_count,  
    String [ ] x_msg_data  
) throws SQLException;
```

---



---

**Note:** p\_check\_foreign\_key indicates whether to do foreign key checking for the profile amount being created. If value equals to HzConstant.getGTrue(), we will do foreign key checking on cust\_account\_id, cust\_account\_profile\_id and site\_use\_id. This was added for backward compatibility with customer form only. You should always set the parameter to HzConstant.getGTrue() when you call API.

---



---

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Profile Amount API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_profile_amt_id	IN	NUMBER	Yes/No	Validation: unique if passed in, else generated from sequence
cust_account_profile_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ When p_check_foreign_key is FND_API.G_TRUE, cust_account_profile_id must be a foreign key to hz_customer_profiles</li> </ul>
currency_code	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ Foreign key to fnd_currencies</li> <li>▪ For a given cust_account_profile_id and currency_code, only one record of the profile amount is allowed.</li> </ul>
trx_credit_limit	IN	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
overall_credit_limit	IN	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
min_dunning_amount	IN	NUMBER	No	
min_dunning_invoice_amount	IN	NUMBER	No	
max_interest_charge	IN	NUMBER	No	
min_statement_amount	IN	NUMBER	No	
auto_rec_min_receipt_amount	IN	NUMBER	No	
interest_rate	IN	NUMBER	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
min_fc_balance_amount	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
min_fc_invoice_amount	IN	NUMBER	No	
cust_account_id	IN	NUMBER	Yes	Validation: Mandatory attribute When p_check_foreign_key is FND_API.G_TRUE, cust_account_id must be a foreign key to hz_cust_accounts. When p_check_foreign_key is FND_API.G_TRUE, cust_account_id should be the same as cust_account_id in corresponding customer profile record.
site_use_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>▪ When p_check_foreign_key is FND_API.G_TRUE, site_use_id must be a foreign key to hz_cust_site_uses</li> <li>▪ When p_check_foreign_key is FND_API.G_TRUE, site_use_id should be the same as cust_account_id in corresponding customer profile record.</li> </ul>
expiration_date	IN	DATE	No	
jgzz_attribute_category	IN	VARCHAR2	No	
jgzz_attribute1	IN	VARCHAR2	No	
jgzz_attribute2	IN	VARCHAR2	No	
jgzz_attribute3	IN	VARCHAR2	No	
jgzz_attribute4	IN	VARCHAR2	No	
jgzz_attribute5	IN	VARCHAR2	No	
jgzz_attribute6	IN	VARCHAR2	No	
jgzz_attribute7	IN	VARCHAR2	No	
jgzz_attribute8	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gzzz_attribute9	IN	VARCHAR2	No	
gzzz_attribute10	IN	VARCHAR2	No	
gzzz_attribute11	IN	VARCHAR2	No	
gzzz_attribute12	IN	VARCHAR2	No	
gzzz_attribute13	IN	VARCHAR2	No	
gzzz_attribute14	IN	VARCHAR2	No	
gzzz_attribute15	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute_category	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
p_check_foreign_key	IN	VARCHAR2	No	Comment: Indicates whether foreign key checks will be done wherever possible.
x_cust_acct_profile_amt_id	OUT	NUMBER	No	Comment: Returns cust_acct_profile_amt_id for the record created.

## Update Customer Profile Amount API

### Description

This routine is used to update Profile Amount. The API updates a record in the HZ\_CUST\_PROFILE\_AMTS table.

### PL/SQL Procedure

```
PROCEDURE update_cust_profile_amt (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_profile_amt_rec  IN          CUST_PROFILE_AMT_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)
```

### Java Method

```
public static void updateCustProfileAmt(
    OracleConnection_connection,
    String p_init_msg_list,
    CustProfileAmtRec p_cust_profile_amt_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
```

```

        String [ ]                x_msg_data
    ) throws SQLException;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Customer Profile Amount API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_profile_amt_id	IN	NUMBER	Yes	Validation: Valid cust_acct_profile_amt_id value must be passed in Comment: Pass cust_acct_profile_amt_id from hz_cust_profile_amts
cust_account_profile_id	IN	NUMBER	No	Validation: Not updateable
currency_code	IN	VARCHAR2	No	Validation: Not updateable
trx_credit_limit	IN	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
overall_credit_limit	IN	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
min_dunning_amount	IN	NUMBER	No	
min_dunning_invoice_amount	IN	NUMBER	No	
max_interest_charge	IN	NUMBER	No	
min_statement_amount	IN	NUMBER	No	
auto_rec_min_receipt_amount	IN	NUMBER	No	
interest_rate	IN	NUMBER	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
min_fc_balance_ amount	IN	NUMBER	No	
min_fc_invoice_ amount	IN	NUMBER	No	
cust_account_id	IN	NUMBER	No	Validation: Not updateable
site_use_id	IN	NUMBER	No	Validation: Not updateable
expiration_date	IN	DATE	No	
jgzz_attribute_ category	IN	VARCHAR2	No	
jgzz_attribute1	IN	VARCHAR2	No	
jgzz_attribute2	IN	VARCHAR2	No	
jgzz_attribute3	IN	VARCHAR2	No	
jgzz_attribute4	IN	VARCHAR2	No	
jgzz_attribute5	IN	VARCHAR2	No	
jgzz_attribute6	IN	VARCHAR2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gzzz_attribute7	IN	VARCHAR2	No	
gzzz_attribute8	IN	VARCHAR2	No	
gzzz_attribute9	IN	VARCHAR2	No	
gzzz_attribute10	IN	VARCHAR2	No	
gzzz_attribute11	IN	VARCHAR2	No	
gzzz_attribute12	IN	VARCHAR2	No	
gzzz_attribute13	IN	VARCHAR2	No	
gzzz_attribute14	IN	VARCHAR2	No	
gzzz_attribute15	IN	VARCHAR2	No	
global_attribute1	IN	VARCHAR2	No	
global_attribute2	IN	VARCHAR2	No	
global_attribute3	IN	VARCHAR2	No	
global_attribute4	IN	VARCHAR2	No	
global_attribute5	IN	VARCHAR2	No	
global_attribute6	IN	VARCHAR2	No	
global_attribute7	IN	VARCHAR2	No	
global_attribute8	IN	VARCHAR2	No	
global_attribute9	IN	VARCHAR2	No	
global_attribute10	IN	VARCHAR2	No	
global_attribute11	IN	VARCHAR2	No	
global_attribute12	IN	VARCHAR2	No	
global_attribute13	IN	VARCHAR2	No	
global_attribute14	IN	VARCHAR2	No	
global_attribute15	IN	VARCHAR2	No	
global_attribute16	IN	VARCHAR2	No	
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	

---

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute20	IN	VARCHAR2	No	
global_attribute_category	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Cannot be updated if value exists
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against value in the database for the existing record</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass current object_version_number of the record from hz_cust_profile_amts table</li> <li>■ Return new value after update</li> </ul>

## Customer Account Role APIs

**PL/SQL Package Name: HZ\_CUST\_ACCOUNT\_ROLE\_V2PUB**

**Java Class Name: HzCustAccountRoleV2Pub**

### PL/SQL Record Structure for Customer Account Role

TYPE cust_account_role_rec_type	IS RECORD (
cust_account_role_id	NUMBER,
party_id	NUMBER,
cust_account_id	NUMBER,
cust_acct_site_id	NUMBER,
primary_flag	VARCHAR2 (1),
role_type	VARCHAR2 (30),
source_code	VARCHAR2 (150),
attribute_category	VARCHAR2 (30),
attribute1	VARCHAR2 (150),
attribute2	VARCHAR2 (150),
attribute3	VARCHAR2 (150),
attribute4	VARCHAR2 (150),
attribute5	VARCHAR2 (150),
attribute6	VARCHAR2 (150),
attribute7	VARCHAR2 (150),
attribute8	VARCHAR2 (150),
attribute9	VARCHAR2 (150),
attribute10	VARCHAR2 (150),
attribute11	VARCHAR2 (150),
attribute12	VARCHAR2 (150),
attribute13	VARCHAR2 (150),
attribute14	VARCHAR2 (150),
attribute15	VARCHAR2 (150),
attribute16	VARCHAR2 (150),
attribute17	VARCHAR2 (150),
attribute18	VARCHAR2 (150),
attribute19	VARCHAR2 (150),
attribute20	VARCHAR2 (150),
attribute21	VARCHAR2 (150),
attribute22	VARCHAR2 (150),
attribute23	VARCHAR2 (150),
attribute24	VARCHAR2 (150),
orig_system_reference	VARCHAR2 (240),
attribute25	VARCHAR2 (150),

```

        status                VARCHAR2(1),
        created_by_module     VARCHAR2(150),
        application_id        NUMBER
    )

```

## PL/SQL Record Structure for Role Responsibility

```

TYPE role_responsibility_rec_type IS RECORD (
    responsibility_id        NUMBER,
    cust_account_role_id    NUMBER,
    responsibility_type     VARCHAR2(30),
    primary_flag            VARCHAR2(1),
    attribute_category      VARCHAR2(30),
    attribute1              VARCHAR2(150),
    attribute2              VARCHAR2(150),
    attribute3              VARCHAR2(150),
    attribute4              VARCHAR2(150),
    attribute5              VARCHAR2(150),
    attribute6              VARCHAR2(150),
    attribute7              VARCHAR2(150),
    attribute8              VARCHAR2(150),
    attribute9              VARCHAR2(150),
    attribute10             VARCHAR2(150),
    attribute11             VARCHAR2(150),
    attribute12             VARCHAR2(150),
    attribute13             VARCHAR2(150),
    attribute14             VARCHAR2(150),
    attribute15             VARCHAR2(150),
    orig_system_reference   VARCHAR2(240),
    created_by_module       VARCHAR2(150),
    application_id          NUMBER
)

```

## Java Inner Class for Customer Account Role

```

public static class CustAccountRoleRec {
    public BigDecimal    cust_account_role_id;
    public BigDecimal    party_id;
    public BigDecimal    cust_account_id;
    public BigDecimal    cust_acct_site_id;
    public String        primary_flag;
    public String        role_type;
    public String        source_code;
    public String        attribute_category;
    public String        attribute1;
}

```

```
public String attribute2;
public String attribute3;
public String attribute4;
public String attribute5;
public String attribute6;
public String attribute7;
public String attribute8;
public String attribute9;
public String attribute10;
public String attribute11;
public String attribute12;
public String attribute13;
public String attribute14;
public String attribute15;
public String attribute16;
public String attribute17;
public String attribute18;
public String attribute19;
public String attribute20;
public String attribute21;
public String attribute22;
public String attribute23;
public String attribute24;
public String orig_system_reference;
public String attribute25;
public String status;
public String created_by_module;
public BigDecimal application_id;

public CustAccountRoleRec();
public CustAccountRoleRec(boolean __RosettaUseGMISSValues);
}
```

### Java Inner Class for Role Responsibility

```
public static class RoleResponsibilityRec {
    public BigDecimal responsibility_id;
    public BigDecimal cust_account_role_id;
    public String responsibility_type;
    public String primary_flag;
    public String attribute_category;
    public String attribute1;
    public String attribute2;
    public String attribute3;
    public String attribute4;
}
```

```

        public String          attribute5;
        public String          attribute6;
        public String          attribute7;
        public String          attribute8;
        public String          attribute9;
        public String          attribute10;
        public String          attribute11;
        public String          attribute12;
        public String          attribute13;
        public String          attribute14;
        public String          attribute15;
        public String          orig_system_reference;
        public String          created_by_module;
        public BigDecimal      application_id;

        public RoleResponsibilityRec();
        public RoleResponsibilityRec(boolean __RosettaUseGMISValues);
    }

```

## Create Customer Account Role API

### Description

This routine is used to create an Account Role. The API creates a record in the HZ\_CUST\_ACCOUNT\_ROLES table. To create a customer account role, you must have already created a customer account and an org contact for the party owning the customer account.

### PL/SQL Procedure

```

PROCEDURE create_cust_account_role (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_account_role_rec  IN          CUST_ACCOUNT_ROLE_REC_TYPE,
    x_cust_account_role_id   OUT         NUMBER,
    x_return_status          OUT         VARCHAR2,
    x_msg_count              OUT         NUMBER,
    x_msg_data               OUT         VARCHAR2
)

```

### Java Method

```

public static void createCustAccountRole(
    OracleConnection_connection,
    String          p_init_msg_list,
    CustAccountRoleRec          p_cust_account_role_rec,
    BigDecimal [ ]          x_cust_account_role_id,

```

```

    String [ ]                x_return_status,
    BigDecimal [ ]           x_msg_count,
    String [ ]                x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Customer Account Role API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_role_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence
party_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Foreign key to hz_parties</li> </ul>
cust_account_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory field</li> <li>■ Foreign key to hz_cust_accounts</li> </ul>
cust_acct_site_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>■ Foreign key to hz_cust_acct_sites</li> <li>■ The cust_account_id in hz_cust_acct_sites which cust_acct_site_id points to should be same as the cust_account_id put in the hz_cust_account_roles</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Primary flag is lookup code in lookup type YES/NO</li> <li>■ It is unique per cust_account_id or cust_acct_site_id</li> </ul> Default: N

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
role_type	IN	VARCHAR2	Y	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ It is a lookup code in lookup type ACCT_ROLE_TYPE<sup>2</sup></li> <li>▪ The combination of CUST_ACCOUNT_ID, PARTY_ID, ROLE_TYPE should be unique.I.</li> <li>▪ Or the combination of CUST_ACCT_SITE_ID, PARTY_ID, ROLE_TYPE should be unique</li> </ul>
source_code	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	Yes	Default: cust_account_role_id
attribute25	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS Default: 'A'
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
x_cust_account_role_id	OUT	NUMBER	No	Comment: Returns cust_account_role_id for the record created

**Other Validations**

When creating cust account role in cust account site level, if the contact referenced by party\_id does not have party site in the same location as this cust account site, API creates a party site for the contact.

**Update Customer Account Role API****Description**

This routine is used to update an Account Role. The API updates a record in the HZ\_CUST\_ACCOUNT\_ROLES table.

**PL/SQL Procedure**

```

PROCEDURE update_cust_account_role (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_cust_account_role_rec IN          CUST_ACCOUNT_ROLE_REC_TYPE,
    p_object_version_number IN OUT     NUMBER,
    x_return_status         OUT         VARCHAR2,
    x_msg_count             OUT         NUMBER,
    x_msg_data              OUT         VARCHAR2
)

```

**Java Method**

```

public static void updateCustAccountRole(
    OracleConnection_connection,
    String                                p_init_msg_list,
    CustAccountRoleRec                    p_cust_account_role_rec,
    BigDecimal [ ]                        p_object_version_number,
    String [ ]                             x_return_status,
    BigDecimal [ ]                         x_msg_count,
    String [ ]                             x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Update Customer Account Role API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_role_id	IN	NUMBER	Yes	Validation: Valid cust_account_role_id should be passed in  Comment: Pass cust_account_role_id from hz_cust_account_roles table
party_id	IN	NUMBER	Yes	Validation: Not updateable
cust_account_id	IN	NUMBER	Yes	Validation: Not updateable
cust_acct_site_id	IN	NUMBER	No	Validation: Not updateable
primary_flag	IN	VARCHAR2	No	Validation: Primary flag is lookup code in lookup type YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
role_type	IN	VARCHAR2	Yes	Validation: Not updateable
source_code	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	Yes	Validation: Not updateable

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute25	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Status cannot be set to null during update</li> <li>■ It is a lookup code in lookup type REGISTRY_STATUS</li> </ul>
created_by_module	IN	VARCHAR2	Yes	Validation: Cannot be updated if value exists
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
xp_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory attribute</li> <li>■ Validated against the value in the database for the existing record.</li> </ul> Comment: <ul style="list-style-type: none"> <li>■ Pass current object_version_number of the record from hz_custaccount_roles</li> <li>■ Returns new value after update</li> </ul>

## Create Role Responsibility API

### Description

This routine is used to create a Role Responsibility. The API creates a record in the HZ\_ROLE\_RESPONSIBILITY table.

### PL/SQL Procedure

```

PROCEDURE create_role_responsibility (
  p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
  p_role_responsibility_rec IN        ROLE_RESPONSIBILITY_REC_TYPE,
  x_responsibility_id      OUT        NUMBER,
  x_return_status          OUT        VARCHAR2,
  x_msg_count              OUT        NUMBER,
  x_msg_data               OUT        VARCHAR2
)

```

**Java Method**

```
public static void createRoleResponsibility(
    OracleConnection_connection,
    String                                p_init_msg_list,
    RoleResponsibilityRec                 p_role_responsibility_rec,
    BigDecimal [ ]                        x_responsibility_id,
    String [ ]                             x_return_status,
    BigDecimal [ ]                        x_msg_count,
    String [ ]                             x_msg_data
) throws SQLException;
```

**Parameter Description and Validation**

The following table lists information about the parameters in the Create Role Responsibility API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
responsibility_id	IN	NUMBER	Yes/No	Validation: unique if passed in, else generated from sequence
cust_account_role_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory field</li> <li>■ Foreign key to hz_cust_account_roles</li> </ul>
responsibility_type	IN	VARCHAR2	Yes	Validation: <ul style="list-style-type: none"> <li>■ Mandatory field</li> <li>■ It is a lookup code in lookup type SITE_USE_CODE</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Primary Flag is lookup code in lookup type YES/NO.</li> <li>■ It is unique per cust_account_role_id Default: N</li> </ul>
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Default: responsibility_id
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module from which creation of record is initiated
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
x_responsibility_id	OUT	NUMBER	No	Comment: Returns responsibility_id for the record created

#### Other Validations

The combination of cust\_account\_role\_id and responsibility\_type should be unique.

## Update Role Responsibility API

#### Description

This routine is used to update a Role Responsibility. The API updates a record in the HZ\_ROLE\_RESPONSIBILITY table.

**PL/SQL Procedure**

```

PROCEDURE update_role_responsibility (
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_role_responsibility_rec IN          ROLE_RESPONSIBILITY_REC_TYPE,
    p_object_version_number IN OUT      NUMBER,
    x_return_status         OUT          VARCHAR2,
    x_msg_count             OUT          NUMBER,
    x_msg_data              OUT          VARCHAR2
)

```

**Java Method**

```

public static void updateRoleResponsibility(
    OracleConnection_connection,
    String p_init_msg_list,
    RoleResponsibilityRec p_role_responsibility_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Update Role Responsibility API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
responsibility_id	IN	NUMBER	Yes	Validation: Valid responsibility_id should be passed in  Comment: Pass responsibility_id from the hz_role_responsibility table
cust_account_role_id	IN	NUMBER	No	Validation: Non updateable
responsibility_type	IN	VARCHAR2	No	Validation: Non updateable
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>■ Primary Flag is lookup code in lookup type YES/NO</li> <li>■ It is unique per cust_account_role_id</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
orig_system_reference	IN	VARCHAR2	No	Validation: Not updateable
created_by_module	IN	VARCHAR2	No	Validation: Cannot be updated if value exists
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>■ Mandatory attribute</li><li>■ Validated against value in the database for the existing record</li></ul> Comment: <ul style="list-style-type: none"><li>■ Pass current object_version_number of the record from hz_role_responsibility<sup>2</sup></li><li>■ Return new value after update</li></ul>

**Other Validations:**

The combination of cust\_account\_role\_id and responsibility\_type should be unique.

---

---

## Miscellaneous API Use

This chapter provides information about the following Trading Community Architecture application programming interfaces:

- Tax Assignment APIs
- Phone Parsing and Formatting APIs
- Name and Address Formatting APIs
- General Data Formatting API
- Hierarchy Retrieval APIs

The information provided for each API includes:

- PL/SQL record structure and procedure.
- Java inner class and method.
- Parameter descriptions and validations.

## Tax Assignment APIs

**PL/SQL Package Name: HZ\_TAX\_ASSIGNMENT\_V2PUB**

**Java Class Name: HzTaxAssignmentV2Pub**

### Create Location Assignment API

#### Description

This routine is used to populate loc\_id after a location is created/ modified. This is called when a customer account site is created. This is to ensure tax validation. It creates a record in the HZ\_LOC\_ASSIGNMENTS table with location\_id (foreign key to HZ\_LOCATIONS.LOCATION\_ID), loc\_id (foreign key to AR\_LOCATION\_COMBINATIONS.LOCATION\_ID) and org\_id (operating unit identifier).

#### PL/SQL Procedure

```
PROCEDURE create_loc_assignment(  
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,  
    p_location_id           IN          NUMBER,  
    p_lock_flag             IN          VARCHAR2:= FND_API.G_FALSE,  
    p_created_by_module     IN          VARCHAR2,  
    p_application_id        IN          NUMBER,  
    x_return_status         IN OUT     VARCHAR2,  
    x_msg_count             OUT        NUMBER,  
    x_msg_data              OUT        VARCHAR2,  
    x_loc_id                OUT        NUMBER  
)
```

#### Java Method

```
public static void createLocAssignment(  
    OracleConnection_connection,  
    String  
    BigDecimal  
    String  
    String  
    BigDecimal  
    String [ ]  
    BigDecimal [ ]  
    String [ ]  
    BigDecimal [ ]  
) throws SQLException;  
    p_init_msg_list,  
    p_location_id,  
    p_lock_flag,  
    p_created_by_module,  
    p_application_id,  
    x_return_status,  
    x_msg_count,  
    x_msg_data,  
    x_loc_id
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Location Assignment API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation	Default	Comment
p_location_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ valid location_id from HZ_LOCATIONS should be passed in.</li> </ul>		
p_lock_flag	IN	NUMBER	No	Validation: none		
p_created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute		Comment: Text to indicate module from which creation of record is initiated
p_application_id	IN	NUMBER	No			Comment: Text to indicate application from which creation of record is initiated
x_loc_id	OUT	NUMBER	No			Comment: Returns loc_id generated

### Other Validations

Org context should be set while calling this API

## Update Location Assignment API

### Description

This routine can be called to populate loc\_id after a location is created/ modified. This to ensure tax validation. It creates a record in the HZ\_LOC\_ASSIGNMENTS table with location\_id (foreign key to HZ\_LOCATIONS), loc\_id (foreign key to AR\_LOCATION\_COMBINATIONS) and org\_id (operating unit identifier).

### PL/SQL Procedure

```
PROCEDURE update_loc_assignment(
    p_init_msg_list          IN          VARCHAR2:= FND_API.G_FALSE,
    p_location_id           IN          NUMBER,
```

```

    p_lock_flag           IN          VARCHAR2:= FND_API.G_TRUE,
    p_created_by_module   IN          VARCHAR2,
    p_application_id      IN          NUMBER,
    x_return_status       IN OUT     VARCHAR2,
    x_msg_count           OUT         NUMBER,
    x_msg_data            OUT         VARCHAR2,
    x_loc_id              OUT         NUMBER
)

```

### Java Method

```

public static void updateLocAssignment(
    OracleConnection connection,
    String p_init_msg_list,
    BigDecimal p_location_id,
    String p_lock_flag,
    String p_created_by_module,
    BigDecimal p_application_id,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data,
    BigDecimal [ ] x_loc_id
) throws SQLException;

```

---

**Note:** p\_lock\_flag indicates whether to lock location record with p\_location\_id passed in. If value equals to HzConstant.getGTrue(), we will try to lock location record before we proceed.

---

### Parameter Description and Validation

The following table lists information about the parameters in the Update Location Assignment API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments,

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_location_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>▪ Mandatory attribute</li> <li>▪ valid location_id from HZ_LOCATIONS should be passed in.</li> </ul>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_lock_flag	IN	NUMBER	No	Validation: none
p_created_by_module	IN	VARCHAR2	Yes	Validation: Non updateable if value exists
p_application_id	IN	NUMBER	No	Validation: Non updateable if value exists
x_loc_id	OUT	NUMBER	No	Comment: Returns loc_id generated

## Phone Parsing and Formatting APIs

**PL/SQL Package Name: HZ\_FORMAT\_PHONE\_V2PUB**

### Phone Number Parsing API

#### Description

Use this routine to parse a raw phone number into the country code, area code and subscriber number based on the setup of country and user phone preferences. Raw phone numbers are an entered string of digits that must include the subscriber number, and may include the international prefix, trunk prefix, country code, and area code. Depending on the country, the phone number may be entered in multiple formats. Each user may enter a number based on his or her personal preferences or location. This API is called from the Contact Point API, when creating or updating a contact point of PHONE type and when the raw phone number is passed to the API. The API returns the parsed country code, area code and subscriber number to the Contact Point API which populates these columns in the HZ\_CONTACT\_POINTS table.

Note: Setting up user preferences is intended for future release.

#### PL/SQL Procedure

```
PROCEDURE phone_parse(  
    p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,  
    p_raw_phone_number      IN          VARCHAR2 := FND_API.G_MISS_CHAR,  
    p_territory_code        IN          VARCHAR2 := FND_API.G_MISS_CHAR,  
    x_phone_country_code    OUT         VARCHAR2,  
    x_phone_area_code      OUT         VARCHAR2,  
    x_phone_number         OUT         VARCHAR2,  
    x_mobile_flag          OUT         VARCHAR2,  
    x_return_status        OUT         VARCHAR2,  
    x_msg_count            OUT         NUMBER,  
    x_msg_data             OUT         VARCHAR2  
)
```

#### Parameter Description and Validation

The following table lists information about the parameters in the Phone Number Parsing API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_raw_phone_number	IN	VARCHAR2	No	Validation : None
p_territory_code	IN	VARCHAR2	No	Validation : None Comment: Territory code parameter should be passed if the user preferences are not set up. This is the territory code of the input Raw Phone Number
x_phone_country_code	OUT	VARCHAR2		Comment : Parsed country code
x_phone_area_code	OUT	VARCHAR2		Comment : Parsed area code
x_phone_number	OUT	VARCHAR2		Comment : Parsed phone number
x_mobile_flag	OUT	VARCHAR2		Comment : Flag indicating whether the number is mobile or not.

## Phone Number Formatting API

### Description

Use this routine to format a phone number for display, based on the appropriate country phone format and the user's preferences. This includes which number segments to display as well as the inclusion of prefixes.

Note: Setting up user preferences is intended for future release.

### PL/SQL Procedure

```

PROCEDURE phone_display(
    p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,
    p_territory_code        IN          VARCHAR2 := FND_API.G_MISS_CHAR,
    p_phone_country_code    IN          VARCHAR2 := FND_API.G_MISS_CHAR,
    p_phone_area_code       IN          VARCHAR2 := FND_API.G_MISS_CHAR,
    p_phone_number          IN          VARCHAR2 := FND_API.G_MISS_CHAR,
    x_formatted_phone_number OUT        VARCHAR2,
    x_return_status         OUT        VARCHAR2,
    x_msg_count             OUT        NUMBER,
    x_msg_data              OUT        VARCHAR2
)

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Phone Number Formatting API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_territory_code	IN	VARCHAR2	No	Comment: Territory code parameter should be passed if the user preferences are not set up. This is the territory code of the input Raw Phone Number
p_phone_country_code	IN	VARCHAR2	No	
p_phone_area_code	IN	VARCHAR2	No	
p_phone_number	IN	VARCHAR2	No	
p_contact_point_id	IN	NUMBER	No	Validation: Contact Point type must be of type Phone
x_formatted_phone_number	OUT	VARCHAR2	No	Comment : Returns the number after formatting

## Name and Address Formatting APIs

**PL/SQL Package Name:** HZ\_FORMAT\_PUB

### Name Formatting API

#### Description

The Name Formatting procedure formats the name of a person using a particular format (Style Format).

Two PL/SQL procedure signatures are available. Use the first if you know the party\_id of the name that you want to format. The procedure queries for the party and formats the name. Otherwise, use the second signature, which accepts the individual components of a person's name as input. In addition, a function version that can be used in SELECT statements or views is available, which returns a single line.

#### Style Formats

Developers usually know only the style they plan to use, for example Informal Salutation, and prefer that the API determines which localized Style Format to use. In this case, you can pass the Style Code, and the procedure attempts to identify the specific localized Style Format based on the user's territory and language.

First, the user's reference locale is determined. This represents a reference territory and language for name and address formatting. These are usually supplied as parameters. If the territory and language are not supplied, they default from the profile options HZ: Reference Territory and HZ: Reference Language. If either of the profile options are not set, then the parameters default with the NLS Territory and NLS Language.

1. Check for a Style Format (for the Style) matching both the user's reference language and territory combination.
2. If not found, check for a Style Format for the style that matches the user's reference territory.
3. If not found, check for a Style Format for the Style that matches the user's reference language.
4. If not found, use the default Style Format that is defined for the Style.

To bypass this derivation logic and force the routine to use a specific Style Format, you can pass that as a parameter instead of the Style Code.

### PL/SQL Procedure (known party\_id)

```

PROCEDURE    format_name(
-- input parameters
    p_party_id           IN           NUMBER
    p_style_code         IN           NUMBER           DEFAULT NULL,
    p_style_format_code  IN           VARCHAR2        DEFAULT NULL,
    p_line_break        IN           VARCHAR2        DEFAULT NULL,
    p_space_replace     IN           VARCHAR2        DEFAULT NULL,
-- context parameters
    p_ref_language_code IN           VARCHAR2        DEFAULT NULL,
    p_ref_territory_code IN          VARCHAR2        DEFAULT NULL,
-- output parameters
    x_return_status     OUT          VARCHAR2
    x_msg_count         OUT          NUMBER
    x_msg_data          OUT          VARCHAR2
    x_formatted_name    OUT          VARCHAR2
    x_formatted_lines_cnt OUT         NUMBER
    x_formatted_name_tbl OUT         HZ_FORMAT_PUB
    .string_tbl_type
)

```

### Parameter Description and Validation

The following table lists information about the parameters in the Name Formatting API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_party_id	In	Number	Yes	Comment: Party ID identifies the row in the HZ_PARTIES table for the person for which you wish to format the name.
p_style_code	In	Varchar2	No	Comment: Supplied if the caller knows the Style, but wants the procedure to figure out the specific Style Format to apply. This is the usual case. This parameter is ignored if p_style_format_code is passed.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_style_format_code	In	Varchar2	No	Comment: Supplied if the caller wishes to specifically use a known Style Format to the address, for example to override the one that the procedure would usually choose.
p_line_break	In	Varchar2	No	Comment: Characters to use to separate multiple lines. For example, <code>&lt;br&gt;</code> for HTML output.
p_space_replace	In	Varchar2	No	Comment: Characters to substitute for <i>blank</i> valued delimiters. For example, <code>&amp;nbsp;</code> for HTML output.
p_ref_language_code	In	Varchar2	No	Comment: Reference Language (context).  Default: Profile option HZ: Reference Language or else the current session NLS Language setting.
p_ref_territory_code	In	Varchar2	No	Comment: Reference Territory (context).  Default: Profile option HZ: Reference Territory or else the current session NLS Territory setting.
x_return_status	Out	Varchar2	Yes	Comment: Return status of API. <ul style="list-style-type: none"> <li>■ S=Success</li> <li>■ E=Error</li> <li>■ U=Unexpected Error</li> </ul>
x_msg_count	Out	Number	Yes	Comment: Number of error messages returned.
x_msg_data	Out	Number	Yes	Comment: Text of messages returned.
x_formatted_name	Out	Varchar2	Yes	Comment: The formatted name returned as a single string with line breaks.
x_formatted_lines_cnt	Out	Number	Yes	Comment: The number of lines in the formatted name.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_formatted_name_ tbl	Out	String_Tbl_ Type	Yes	Comment: The formatted name returned as multiple strings, one for each line.

### PL/SQL Procedure (unknown party\_id)

```

PROCEDURE    format_name(
-- input parameters
    p_style_code           IN          NUMBER    DEFAULT NULL,
    p_style_format_code   IN          VARCHAR2  DEFAULT NULL,
    p_line_break          IN          VARCHAR2  DEFAULT NULL,
    p_space_replace       IN          VARCHAR2  DEFAULT NULL,
-- person name components
    p_person_title        IN          VARCHAR2  DEFAULT NULL,
    p_person_first_name   IN          VARCHAR2  DEFAULT NULL,
    p_person_middle_name  IN          VARCHAR2  DEFAULT NULL,
    p_person_last_name    IN          VARCHAR2  DEFAULT NULL,
    p_person_suffix       IN          VARCHAR2  DEFAULT NULL,
    p_person_person_known_as IN        VARCHAR2  DEFAULT NULL,
    p_person_first_name_phonetic IN     VARCHAR2  DEFAULT NULL,
    p_person_middle_name_phonetic IN    VARCHAR2  DEFAULT NULL,
    p_person_last_name_phonetic IN     VARCHAR2  DEFAULT NULL,
-- context parameters
    p_ref_language_code   IN          VARCHAR2  DEFAULT NULL,
    p_ref_territory_code  IN          VARCHAR2  DEFAULT NULL,
-- output parameters
    x_return_status       OUT         VARCHAR2
    x_msg_count           OUT         NUMBER
    x_msg_data            OUT         VARCHAR2
    x_formatted_name      OUT         VARCHAR2
    x_formatted_lines_cnt OUT         NUMBER
    x_formatted_name_tbl  OUT         HZ_FORMAT_PUB.string_tbl_type
)

```

### Parameter Description and Validation

The following table only lists information specifically about the second of the two signatures available for this procedure. Refer to this table and the table for the first signature for information about parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_person_title	In	Varchar2	No	Comment: conventional, professional, or honorific title such as Mrs. or Dr.
p_person_first_name	In	Varchar2	No	Comment: person's first name
p_person_middle_name	In	Varchar2	No	Comment: person's middle name
p_person_last_name	In	Varchar2	No	Comment: person's last name
p_person_suffix	In	Varchar2	No	Comment: name suffix. For example, in English, a common custom to distinguish offspring with same given names is to use a generation indicator. The suffix can be a Roman numeral such as <i>II</i> or <i>III</i> , or a string such as <i>Jr.</i> or <i>Sr.</i>
p_person_known_as	In	Varchar2	No	Comment: alternative or also-known-as (AKA) name
p_first_name_phonetic	In	Varchar2	No	Comment: phonetic representation of person's first name
p_first_middle_phonetic	In	Varchar2	No	Comment: phonetic representation of person's middle name
p_first_last_phonetic	In	Varchar2	No	Comment: phonetic representation of person's last name

### PL/SQL Function

```

FUNCTION    format_name(
-- input parameters
    p_party_id          IN          NUMBER
    p_style_code        IN          NUMBER          DEFAULT NULL,
    p_style_format_code IN          VARCHAR2       DEFAULT NULL,
    p_line_break        IN          VARCHAR2       DEFAULT NULL,
    p_space_replace     IN          VARCHAR2       DEFAULT NULL,
-- context parameters
    p_ref_language_code IN          VARCHAR2       DEFAULT NULL,
    p_ref_territory_code IN         VARCHAR2       DEFAULT NULL

```

) RETURN VARCHAR2

### Parameter Description and Validation

Refer to the parameter descriptions and validations for the `format_name` procedure for information about parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

## Address Formatting API

### Description

The Address Formatting procedure formats an address using a particular format (Style Format).

Two PL/SQL procedure signatures are available. Use the first if you know the `location_id` of the address you want to format. The procedure queries for the location and formats the address. Otherwise, use the second signature, which accepts the individual address components as input. In addition, a function version that can be used in SELECT statements or views is available, which returns a single line.

### Style Formats

The API will accept one of two parameters for determining format: the Style code or the Style Format code. Normally, the developer would know only the style they wish to use, such as Mailing Address with multiple lines, and would prefer to have the API figure out which localized Style Format to use. In this case, you can pass the Style, and the procedure attempts to find the specific Style Format based on the territory where the address is located.

If you know the specific Style Format you wish to use, such as the France Mailing Address, and you want the API to use this Style Format, regardless of having the API determine which Style Format to use, then you can pass it as a parameter. This mode is provided for flexibility, but this way of using the procedure is generally not the case.

Both parameters are optional. If you do not pass either parameter, then the HZ: Default Address Style profile option determines the default address Style. The Style Format will be determined by the address's territory.

If you do want to pass style information, you should pass either the Style Format or the Style parameter, but not both. If you pass both, the Style Format parameter takes precedence, and the Style parameter will be ignored.

**PL/SQL Procedure**

```

PROCEDURE    format_address(
-- input parameters
    p_location_id          IN          NUMBER
    p_style_code           IN          NUMBER          DEFAULT NULL,
    p_style_format_code    IN          VARCHAR2       DEFAULT NULL,
    p_line_break           IN          VARCHAR2       DEFAULT NULL,
    p_space_replace        IN          VARCHAR2       DEFAULT NULL,
-- context parameters
    p_to_language_code     IN          VARCHAR2       DEFAULT NULL,
    p_country_name_lang    IN          VARCHAR2       DEFAULT NULL,
    p_from_territory_code  IN          VARCHAR2       DEFAULT NULL,
-- output parameters
    x_return_status        OUT         VARCHAR2
    x_msg_count            OUT         NUMBER
    x_msg_data             OUT         VARCHAR2
    x_formatted_address    OUT         VARCHAR2
    x_formatted_lines_cnt  OUT         NUMBER
    x_formatted_address_tbl OUT         HZ_FORMAT_PUB
                                                .string_tbl_type
)

```

**Parameter Description and Validation**

The following table lists information about the parameters in the Address Formatting API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_location_id	In	Number	Yes	Comment: Location ID identifies the row in the HZ_LOCATIONS table for which you wish the address formatted.
p_style_code	In	Varchar2	No	Comment: Supplied if the caller knows the Style, but wants the procedure to figure out the specific Style Format to apply. This is the usual case.  This parameter is ignored if p_style_format_code is passed.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_style_format_code	In	Varchar2	No	Comment: Supplied if the caller wishes to specifically use a known Style Format to the address, for example to override the one that the procedure would usually choose.
p_line_break	In	Varchar2	No	Comment: Characters to use to separate multiple lines. For example, <code>&lt;br&gt;</code> for HTML output.
p_space_replace	In	Varchar2	No	Comment: Characters to substitute for <i>blank</i> valued delimiters. For example, <code>&amp;nbsp;</code> for HTML output.
p_to_language_code	In	Varchar2	No	Comment: Language that is used at the destination location.
p_country_name_lang	In	Varchar2	No	Comment: Language used to display the country name Default: Profile option HZ: Language for country name, or if not set then the current session NLS Language setting.
p_from_territory_code	In	Varchar2	No	Comment: Territory of the sender. Default: Profile option HZ: Reference Territory, or if not set then current session NLS Territory setting.
x_return_status	Out	Varchar2	Yes	Comment: Return status of API. <ul style="list-style-type: none"> <li>■ S=Success</li> <li>■ E=Error</li> <li>■ U=Unexpected Error</li> </ul>
x_msg_count	Out	Number	Yes	Comment: Number of error messages returned.
x_msg_data	Out	Number	Yes	Comment: Text of messages returned.
x_formatted_address	Out	Varchar2	Yes	Comment: The formatted address returned as a single string with line breaks.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_formatted_lines_cnt	Out	Number	Yes	Comment: The number of lines in the formatted address.
x_formatted_address_tbl	Out	String_Tbl_Type	Yes	Comment: The formatted address returned as multiple strings, one for each line.

### Parameter Description and Validation

The following table only lists information specifically about the second of the two signatures available for this procedure. Refer to this table and the table for the first signature for information about parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_address_line_1	In	Varchar2	No	Comment: address line 1
p_address_line_2	In	Varchar2	No	Comment: address line 2
p_address_line_3	In	Varchar2	No	Comment: address line 3
p_address_line_4	In	Varchar2	No	Comment: address line 4
p_city	In	Varchar2	No	Comment: City
p_postal_code	In	Varchar2	No	Comment: Postal code or ZIP code
p_state	In	Varchar2	No	Comment: State
p_province	In	Varchar2	No	Comment: Province
p_county	In	Varchar2	No	Comment: County
p_country	In	Varchar2	No	Comment: Country of the address
p_address_lines_phonetic	In	Varchar2	No	Comment: Phonetic representation of the address

### PL/SQL Function

```

FUNCTION    format_address (
-- context parameters
    p_location_id          IN          NUMBER
    p_style_code           IN          NUMBER          DEFAULT NULL,

```

```
    p_style_format_code      IN      VARCHAR2      DEFAULT NULL,
    p_line_break             IN      VARCHAR2      DEFAULT NULL,
    p_space_replace          IN      VARCHAR2      DEFAULT NULL,
-- input parameters
    p_to_language_code       IN      VARCHAR2      DEFAULT NULL,
    p_country_name_lang      IN      VARCHAR2      DEFAULT NULL,
    p_from_territory_code    IN      VARCHAR2      DEFAULT NULL
) RETURN VARCHAR2
```

### **Parameter Description and Validation**

The information about the parameters of this signature is the same as the information about the parameters for the first `format_address` signature. Refer to the table above for the appropriate parameter information.

## General Data Formatting API

### Description

The solutions provided for name and address formatting can be used to format any type of information from any data source. This generic formatting routine provides a way for you set up the Style Metadata for the data you wish to format.

### PL/SQL Procedure

```
PROCEDURE    format_name(
-- input parameters
    p_object_code          IN      VARCHAR2
    p_object_key_1         IN      VARCHAR2
    p_object_key_2         IN      VARCHAR2
    p_object_key_3         IN      VARCHAR2
    p_object_key_4         IN      VARCHAR2
    p_style_code           IN      NUMBER          DEFAULT NULL,
    p_style_format_code    IN      VARCHAR2       DEFAULT NULL,
    p_line_break           IN      VARCHAR2       DEFAULT NULL,
    p_space_replace        IN      VARCHAR2       DEFAULT NULL,
-- context parameters
    p_ref_language_code    IN      VARCHAR2       DEFAULT NULL,
    p_ref_territory_code   IN      VARCHAR2       DEFAULT NULL,
-- output parameters
    x_return_status        OUT     VARCHAR2
    x_msg_count            OUT     NUMBER
    x_msg_data             OUT     VARCHAR2
    x_formatted_data       OUT     VARCHAR2
    x_formatted_lines_cnt  OUT     NUMBER

    x_formatted_data_tbl   OUT     HZ_FORMAT_PUB.string_tbl_type
)

```

## Hierarchy Retrieval APIs

**PL/SQL Package Name : HZ\_HIERARCHY\_V2PUB**

### PL/SQL Record Structure for Location

```

TYPE related_nodes_list_rec          IS RECORD (
  related_node_id                    NUMBER(15),
  related_node_table_name            VARCHAR2(30),
  related_node_object_type           VARCHAR2(30),
  level_number                       NUMBER(15),
  top_parent_flag                    VARCHAR2(1),
  leaf_child_flag                    VARCHAR2(1),
  effective_start_date               DATE,
  effective_end_date                 DATE,
  relationship_id                    NUMBER(15),
)
TYPE related_nodes_list_type IS TABLE OF related_nodes_list_rec INDEX BY BINARY_
INTEGER;

```

### Parameter Description and Validation

The following table lists information about the parameters in the Hierarchy Retrieval API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_init_msg_list	IN	VARCHAR2	No	Comment : Indicates whether message stack should initialized Default : FND_API.G_FALSE
p_hierarchy_type	IN	VARCHAR2	Yes	Validation : Must be a hierarchical relationship type from hz_relationship_types table
p_parent_id	IN	NUMBER	Yes	
p_parent_table_name	IN	VARCHAR2	No	Default : 'HZ_PARTIES'
p_parent_object_type	IN	VARCHAR2	No	Default : 'ORGANIZATION'
p_child_id	IN	NUMBER	Yes	
p_child_table_name	IN	VARCHAR2	No	Default : 'HZ_PARTIES'

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_child_object_type	IN	VARCHAR2	No	Default : 'ORGANIZATION'
p_effective_date	IN	DATE	No	Default : SYSDATE
p_include_node	IN	VARCHAR2	No	Comment : Indicates whether the parent/child itself should be included in the result set Default : Y
p_no_of_records	IN	NUMBER	No	Comment : Number of records to be returned. Maximum is 100 .Default : 100
x_result	OUT	VARCHAR2	Yes	Comment : Y/N
x_return_status	OUT	VARCHAR2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR2	Yes	Comment : Message text if x_msg_count = 1
Record related_nodes_list_rec :				
related_node_id		NUMBER		Id of the related entity
related_node_table_name		VARCHAR2		Table name for the related entity
related_node_object_type		VARCHAR2		Object type of the related entity
level_number		NUMBER		Level at which the entity is related to
top_parent_flag		VARCHAR2		Indicates whether related entity is top parent
leaf_child_flag		VARCHAR2		Indicates whether related entity is a leaf node
effective_start_date		DATE		Start date of the relationship
effective_end_date		DATE		End date of the relationship
relationship_id		NUMBER		Relationship Id if it is a direct link

## Top Parent Check API

### Description

Use this routine to check whether an entity is top parent, or root, in a hierarchy or not. Please pass a valid hierarchy relationship type and any necessary parent information. The API will return Y or N.

### PL/SQL Procedure

```
PROCEDURE is_top_parent(  
  p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,  
  p_hierarchy_type        IN          VARCHAR2 ,  
  p_parent_id             IN          NUMBER ,  
  p_parent_table_name     IN          VARCHAR2 := 'HZ_PARTIES' ,  
  p_parent_object_type    IN          VARCHAR2 := 'ORGANIZATION' ,  
  p_effective_date        IN          DATE := SYSDATE ,  
  x_result                OUT         VARCHAR2 ,  
  x_return_status         OUT         VARCHAR2 ,  
  x_msg_count             OUT         NUMBER ,  
  x_msg_data              OUT         VARCHAR2  
)
```

## Parent Child Relationship Check API

### Description

Use this routine to check whether two entities have a parent-child relationship in a hierarchy. Pass a valid hierarchy relationship type and any necessary parent and child information. The API will return Y or N.

### PL/SQL Procedure

```
PROCEDURE check_parent_child(  
  p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,  
  p_hierarchy_type        IN          VARCHAR2 ,  
  p_parent_id             IN          NUMBER ,  
  p_parent_table_name     IN          VARCHAR2 := 'HZ_PARTIES' ,  
  p_parent_object_type    IN          VARCHAR2 := 'ORGANIZATION' ,  
  p_child_id              IN          NUMBER ,  
  p_child_table_name      IN          VARCHAR2 := 'HZ_PARTIES' ,  
  p_child_object_type     IN          VARCHAR2 := 'ORGANIZATION' ,  
  p_effective_date        IN          DATE := SYSDATE ,  
  x_result                OUT         VARCHAR2 ,  
  x_level_number          OUT         NUMBER ,  
  x_return_status         OUT         VARCHAR2 ,
```

```

        x_msg_count          OUT          NUMBER,
        x_msg_data          OUT          VARCHAR2
    )

```

## Get Parent Nodes API

### Description

Use this routine to retrieve the parent nodes of a child in a hierarchy. Pass a valid hierarchy relationship type and any necessary parent type and child node information. The API returns a set of parent nodes in that hierarchy.

### PL/SQL Procedure

```

PROCEDURE get_parent_nodes(
    p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,
    p_hierarchy_type        IN          VARCHAR2,
    p_child_id              IN          NUMBER,
    p_child_table_name     IN          VARCHAR2,
    p_child_object_type    IN          VARCHAR2,
    p_parent_table_name    IN          VARCHAR2,
    p_parent_object_type   IN          VARCHAR2,
    p_include_node         IN          VARCHAR2 := 'Y',
    p_effective_date       IN          DATE := SYSDATE,
    p_no_of_records        IN          NUMBER := 100,
    x_related_nodes_list   OUT NOCOPY  RELATED_NODES_LIST_TYPE,
    x_return_status        OUT          VARCHAR2,
    x_msg_count            OUT          NUMBER,
    x_msg_data             OUT          VARCHAR2
)

```

## Get Child Nodes API

### Description

Use this routine to retrieve the child nodes of a parent in a hierarchy. Pass a valid hierarchy relationship type and any necessary child type and parent node information. The API returns a set of child nodes in that hierarchy.

### PL/SQL Procedure

```

PROCEDURE get_child_nodes(
    p_init_msg_list          IN          VARCHAR2 := FND_API.G_FALSE,
    p_hierarchy_type        IN          VARCHAR2,
    p_parent_id             IN          NUMBER,
    p_parent_table_name     IN          VARCHAR2,
    p_parent_object_type   IN          VARCHAR2,

```

```
p_child_table_name      IN          VARCHAR2,
p_child_object_type     IN          VARCHAR2,
p_include_node         IN          VARCHAR2 := 'Y',
p_effective_date       IN          DATE := SYSDATE,
p_no_of_records        IN          NUMBER := 100,
x_related_nodes_list   OUT NOCOPY  RELATED_NODES_LIST_TYPE,
x_return_status        OUT          VARCHAR2,
x_msg_count            OUT          NUMBER,
x_msg_data             OUT          VARCHAR2
)

```

### Get Top Parent Nodes API

#### Description

Use this routine to retrieve the top parent nodes in a hierarchy. Pass a valid hierarchy relationship type. The API returns a set of the top parent nodes in that hierarchy.

#### PL/SQL Procedure

```
PROCEDURE get_top_parent_nodes(
  p_init_msg_list      IN          VARCHAR2 := FND_API.G_FALSE,
  p_hierarchy_type     IN          VARCHAR2,
  p_parent_table_name  IN          VARCHAR2 := 'HZ_PARTIES',
  p_parent_object_type IN          VARCHAR2 := 'ALL',
  p_effective_date     IN          DATE := SYSDATE,
  p_no_of_records      IN          NUMBER := 100,
  x_top_parent_list    OUT NOCOPY  RELATED_NODES_LIST_TYPE,
  x_return_status      OUT          VARCHAR2,
  x_msg_count          OUT          NUMBER,
  x_msg_data           OUT          VARCHAR2
)

```

# A

---

---

## Sample Code

This appendix provides sample code that demonstrates the use of the TCA APIs.

## Sample Code - Demonstrating the APIs

This section provides many sample codes to demonstrate how TCA API can be called to create and update entities. The sample code is provided for PL/SQL API as well as Java API. These are just sample code and do not attempt to cover any complete business case. These sample codes will help developers to quickly code required functionality using TCA API.

### Create an Organization

#### PL/SQL Example:

```
DECLARE
    p_organization_rec  HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE;
    x_return_status     VARCHAR2(2000);
    x_msg_count         NUMBER;
    x_msg_data          VARCHAR2(2000);
    x_party_id          NUMBER;
    x_party_number      VARCHAR2(2000);
    x_profile_id        NUMBER;

BEGIN
    p_organization_rec.organization_name := 'ABC Corporation';
    p_organization_rec.created_by_module := 'TCA_EXAMPLE';

    hz_party_v2pub.create_organization (
        'T',
        p_organization_rec,
        x_return_status,
        x_msg_count,
        x_msg_data,
        x_party_id,
        x_party_number,
        x_profile_id);

    dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
    dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
    dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

    IF x_msg_count > 1 THEN

        FOR I IN 1..x_msg_count

            LOOP
```

```

        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_API.G_
        FALSE ), 1, 255));

        END LOOP;
    END IF;
END;
```

### Java Example:

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzPartyV2Pub;

class CreateOrganization {
    public static void main(String[] args) throws Exception {

        HzPartyV2Pub.OrganizationRec p_organization_rec = new HzPartyV2Pub.OrganizationRec();
        BigDecimal[]      x_party_id = new BigDecimal[1];
        String[]          x_party_number = new String[1];
        BigDecimal[]      x_profile_id = new BigDecimal[1];

        String[]          x_return_status = new String[1];
        BigDecimal[]      x_msg_count = new BigDecimal[1];
        String[]          x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartyV2Pub partyV2Pub = new HzPartyV2Pub();

            p_organization_rec.organization_name = "ABC Corporation";
            p_organization_rec.created_by_module = "TCA_EXAMPLE";

            partyV2Pub.createOrganization(
                conn
                , "T"
```

```
, p_organization_rec
, x_return_status
, x_msg_count
, x_msg_data
, x_party_id
, x_party_number
, x_profile_id
);

System.out.println( "x_return_status = " + x_return_status[0] );
System.out.println( "x_msg_count = " + x_msg_count[0] );
System.out.println( "x_msg_data = " + x_msg_data[0] );

if (x_msg_count[0].intValue() > 1) {
    OracleCallableStatement ocs = null;
    for (int i=0; i<x_msg_count[0].intValue(); i++) {
        ocs = (OracleCallableStatement)conn.prepareCall(
            "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
        ocs.registerOutParameter(1, OracleTypes.VARCHAR);
        ocs.execute();
        System.out.println((i + 1) + ". " + ocs.getString(1));
    }
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
}
```

**Out Parameters:**

```
x_return_status = S
x_msg_count = 0
x_msg_data =
x_party_id = 1272023
x_party_number = 1268621
x_profile_id = 99782
```

**Select Statement to Check Data:**

```
select party_id, party_number, party_name, object_version_number, created_by_module
from hz_parties
where party_id = 1272023;
```

The following table provides information about this select statement.

<b>PARTY_ID</b>	<b>PARTY_NUMBER</b>	<b>PARTY_NAME</b>	<b>OBJ_VER</b>	<b>CREATED_BY_MOD</b>
1272023	1268621	ABC Corporation	1	TCA_EXAMPLE

```
select organization_profile_id, party_id, organization_name, object_version_number, created_by_module
from hz_organization_profiles
where party_id = 1272023;
```

The following table provides information about this select statement.

<b>ORG_PROFILE_ID</b>	<b>PARTY_ID</b>	<b>ORGANIZATION_NAME</b>	<b>OBJ_VER_NUM</b>	<b>CREATED_BY_M</b>
99782	1272023	ABC Corporation	1	TCA_EXAMPLE

Similarly you can call `hz_party_v2pub.create_person` to create record in the `HZ_PARTIES` and `HZ_PERSON_PROFILES` tables and `hz_party_v2pub.create_group` to create record in the `HZ_PARTIES` table.

## Create a Location

### PL/SQL Example

```
DECLARE
    p_location_rec          HZ_LOCATION_V2PUB.LOCATION_REC_TYPE;
    x_location_id          NUMBER;
    x_return_status        VARCHAR2(2000);
    x_msg_count            NUMBER;
    x_msg_data             VARCHAR2(2000);
BEGIN
    p_location_rec.country := 'US';
    p_location_rec.address1 := '300 Oracle Parkway';
    p_location_rec.address2 := '13th Floor';
    p_location_rec.city := 'Redwood Shores';
    p_location_rec.postal_code := '94065';
    p_location_rec.state := 'CA';
    p_location_rec.created_by_module := 'TCA_EXAMPLE';

    hz_location_v2pub.create_location(
        dbms_output.put_line(SubStr('x_return_status='||x_return_status,1,255))
```

```

                                T,
                                p_location_rec,
                                x_location_id,
                                x_return_status,
                                x_msg_count,
                                x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_API.G_
            FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzLocationV2Pub;

class CreateLocation {
    public static void main(String[] args) throws Exception {

        HzLocationV2Pub.LocationRec p_location_rec = new HzLocationV2Pub.LocationRec();
        BigDecimal[]          x_location_id = new BigDecimal[1];

        String[]              x_return_status = new String[1];
        BigDecimal[]          x_msg_count = new BigDecimal[1];
        String[]              x_msg_data = new String[1];

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
```

```

        "jdbc:oracle:thin:@ap103fam:1521:findv115",
        "apps", "apps");

    HzLocationV2Pub locationV2Pub = new HzLocationV2Pub();

    p_location_rec.country = "US";
    p_location_rec.address1 = "300 Oracle Parkway";
    p_location_rec.address2 = "13th Floor";
    p_location_rec.city = "Redwood Shores";
    p_location_rec.postal_code = "94065";
    p_location_rec.state = "CA";
    p_location_rec.created_by_module = "TCA_EXAMPLE";

    locationV2Pub.createLocation(
        conn
        , "T"
        , p_location_rec
        , x_location_id
        , x_return_status
        , x_msg_count
        , x_msg_data
        );

    System.out.println( "x_return_status = " + x_return_status[0] );
    System.out.println( "x_msg_count = " + x_msg_count[0] );
    System.out.println( "x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? := fnd_msg_pub.get( p_encoded => F ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + ". " + ocs.getString(1));
        }
    }

    conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error: " + e);
}
}
}

```

```
}

```

### Out Parameters

```
x_location_id = 359086
x_return_status = S
x_msg_count = 0
x_msg_data =
```

### Select Statement to Check Data

```
select location_id, address1, address2, city, state, postal_code, country,
       object_version_number, created_by_module
from hz_locations
where location_id = 359086;
```

The following tables provide information about this select statement.

LOCATION_ID	ADDRESS1	ADDRESS2	CITY	STATE	POSTAL_CODE
359086	300 Oracle Parkway	13th Floor	Redwood Shores	CA	94065

COUNTRY	OBJECT_VERSION_NUMBER	CREATED_BY_MODULE
US	1	TCA_EXAMPLE

## Create a Party Site

The following examples use the organizations and locations created in the previous examples.

### PL/SQL Example

```
DECLARE
    p_party_site_rec          HZ_PARTY_SITE_V2PUB.PARTY_SITE_REC_TYPE;
    x_party_site_id          NUMBER;
    x_party_site_number      VARCHAR2(2000);
    x_return_status         VARCHAR2(2000);
    x_msg_count             NUMBER;
    x_msg_data              VARCHAR2(2000);
BEGIN
    p_party_site_rec.party_id := 1272023;
    p_party_site_rec.location_id := 359086;
    p_party_site_rec.identifying_address_flag := 'Y';
    p_party_site_rec.created_by_module := 'TCA_EXAMPLE';
```

```

hz_party_site_v2pub.create_party_site(
    T,
    p_party_site_rec,
    x_party_site_id,
    x_party_site_number,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count > 1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
            API.G_FALSE ), 1, 255));
    END LOOP;
END IF;

END;

```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzPartySiteV2Pub;

class CreatePartySite {
    public static void main(String[] args) throws Exception {

        HzPartySiteV2Pub.PartySiteRec p_party_site_rec = new HzPartySiteV2Pub.PartySiteRec();
        BigDecimal[]      x_party_site_id = new BigDecimal[1];
        String[]          x_party_site_number = new String[1];

        String[]          x_return_status = new String[1];
        BigDecimal[]      x_msg_count = new BigDecimal[1];
        String[]          x_msg_data = new String[1];
    }
}

```

```

try {
    Class.forName ("oracle.jdbc.driver.OracleDriver");
    OracleConnection conn = (OracleConnection)
        DriverManager.getConnection (
            "jdbc:oracle:thin:@ap103fam:1521:findv115",
            "apps", "apps");

    HzPartySiteV2Pub partysiteV2Pub = new HzPartySiteV2Pub();

    p_party_site_rec.party_id = new BigDecimal("1272023");
    p_party_site_rec.location_id = new BigDecimal ("359086");
    p_party_site_rec.identifying_address_flag = "Y";
    p_party_site_rec.created_by_module = "TCA_EXAMPLE";

    partysiteV2Pub.createPartySite(
        conn
        , "T"
        , p_party_site_rec
        , x_party_site_id
        , x_party_site_number
        , x_return_status
        , x_msg_count
        , x_msg_data
        );

    System.out.println( "x_return_status = " + x_return_status[0] );
    System.out.println( "x_msg_count = " + x_msg_count[0] );
    System.out.println( "x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? := find_msg_pub.get( p_encoded => 'F' ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + " " + ocs.getString(1));
        }
    }

    conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}

```

```

}
}
}

```

### Out Parameters

```

x_party_site_id = 349327
x_party_site_number = 347092
x_return_status = S
x_msg_count = 0
x_msg_data =

```

### Select Statement to Check Data

```

select party_site_id, party_id, location_id, party_site_number
from hz_party_sites
where party_site_id = 349327;

```

The following table provides information about this select statement.

PARTY_SITE_ID	PARTY_ID	LOCATION_ID	PARTY_SITE_NUMBER
349327	1272023	359086	347092

Because the party site is created with the identifying\_address\_flag column set to Y, location information is denormalized in the HZ\_PARTIES table.

### Select Statement to Check Data

```

select address1, address2, city, state, postal_code, country
from hz_parties
where party_id = 1272023;

```

The following table provides information about this select statement.

ADDRESS1	ADDRESS2	CITY	STATE	POSTAL_CODE	COUNTRY
300 Oracle Parkway	13th Floor	Redwood Shores	CA	940655	US

## Create a Party Site Use

Using the same party site created in the previous example.

The following example uses the the party site created in the previous example.

### PL/SQL Example

```

DECLARE
    p_party_site_use_rec          HZ_PARTY_SITE_V2PUB.PARTY_SITE_USE_REC_TYPE;
    x_party_site_use_id          NUMBER;
    x_return_status              VARCHAR2(2000);
    x_msg_count                  NUMBER;
    x_msg_data                   VARCHAR2(2000);
BEGIN
    p_party_site_use_rec.site_use_type := 'SHIP_TO';
    p_party_site_use_rec.party_site_id := 349327;
    p_party_site_use_rec.created_by_module := 'TCA_EXAMPLE';

    hz_party_site_v2pub.create_party_site_use(
        T,
        p_party_site_use_rec,
        x_party_site_use_id,
        x_return_status,
        x_msg_count,
        x_msg_data);

    dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
    dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
    dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

    IF x_msg_count >1 THEN
        FOR I IN 1..x_msg_count
        LOOP
            dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
                API.G_FALSE ), 1, 255));
        END LOOP;
    END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzPartySiteV2Pub;

class CreatePartySiteUse {
```

```

public static void main(String[] args) throws Exception {

    HzPartySiteV2Pub.PartySiteUseRec p_party_site_use_rec = new HzPartySiteV2Pub.PartySiteUseRec();
    BigDecimal[]          x_party_site_use_id = new BigDecimal[1];

    String[]              x_return_status = new String[1];
    BigDecimal[]          x_msg_count = new BigDecimal[1];
    String[]              x_msg_data = new String[1];

    try {
        Class.forName ("oracle.jdbc.driver.OracleDriver");
        OracleConnection conn = (OracleConnection)
            DriverManager.getConnection (
                "jdbc:oracle:thin:@ap103fam:1521:findv115",
                "apps", "apps");

        HzPartySiteV2Pub partysiteV2Pub = new HzPartySiteV2Pub();

        p_party_site_use_rec.site_use_type = "SHIP_TO";
        p_party_site_use_rec.party_site_id = new BigDecimal("349327");
        p_party_site_use_rec.created_by_module = "TCA_EXAMPLE";

        partysiteV2Pub.createPartySiteUse(
            conn
            , "T"
            , p_party_site_use_rec
            , x_party_site_use_id
            , x_return_status
            , x_msg_count
            , x_msg_data
            );

        System.out.println( "x_return_status = " + x_return_status[0] );
        System.out.println( "x_msg_count = " + x_msg_count[0] );
        System.out.println( "x_msg_data = " + x_msg_data[0] );

        if (x_msg_count[0].intValue() > 1) {
            OracleCallableStatement ocs = null;
            for (int i=0; i<x_msg_count[0].intValue(); i++) {
                ocs = (OracleCallableStatement)conn.prepareCall(
                    "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
                ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                ocs.execute();
                System.out.println((i + 1) + ". " + ocs.getString(1));
            }
        }
    }
}

```

```

    }

    conn.close();
  } catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
  } catch (SQLException e) {
    System.out.println("SQL Error." + e);
  }
}
}
}

```

**Out Parameters:**

```

x_party_site_use_id = 13523754
x_return_status = S
x_msg_count = 0
x_msg_data =

```

**Select Statement to Check Data**

```

select party_site_use_id, party_site_id, site_use_type, primary_per_type
from hz_party_site_uses
where party_site_use_id = 13523754;

```

The following table provides information about this select statement.

PARTY_SITE_USE_ID	PARTY_SITE_ID	USE_TYPE	PRIMARY_PER_TYPE
13523754	349327	SHIP_TO	Y

Because the example above is the first SHIP\_TO site use created for party ID 1272023, the primary\_per\_type attribute is set to Y.

**Create a Contact Point (Phone)**

This procedure creates a phone number for the same organization created in the first example.

**PL/SQL Example**

```

DECLARE
    p_contact_point_rec          HZ_CONTACT_POINT_V2PUB.CONTACT_POINT_
                                REC_TYPE;
    p_edi_rec                   HZ_CONTACT_POINT_V2PUB.EDI_REC_TYPE;
    p_email_rec                 HZ_CONTACT_POINT_V2PUB.EMAIL_REC_TYPE;
    p_phone_rec                 HZ_CONTACT_POINT_V2PUB.PHONE_REC_TYPE;
    p_telex_rec                 HZ_CONTACT_POINT_V2PUB.TELEX_REC_TYPE;

```

```

p_web_rec                HZ_CONTACT_POINT_V2PUB.WEB_REC_TYPE;
x_return_status          VARCHAR2(2000);
x_msg_count              NUMBER;
x_msg_data               VARCHAR2(2000);
x_contact_point_id      NUMBER;

BEGIN

p_contact_point_rec.contact_point_type := 'PHONE';
p_contact_point_rec.owner_table_name := 'HZ_PARTIES';
p_contact_point_rec.owner_table_id := '1272023';
p_contact_point_rec.primary_flag := 'Y';
p_contact_point_rec.contact_point_purpose := 'BUSINESS';
p_phone_rec.phone_area_code := '650';
p_phone_rec.phone_country_code := '1';
p_phone_rec.phone_number := '506-7000';
p_phone_rec.phone_line_type := 'GEN';
p_contact_point_rec.created_by_module := 'TCA_EXAMPLE';

hz_contact_point_v2pub.create_contact_point(
    'T',
    p_contact_point_rec,
    p_edt_rec,
    p_email_rec,
    p_phone_rec,
    p_telex_rec,
    p_web_rec,
    x_contact_point_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count > 1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('||SubStr(FND_MSG_PUB.Get(p_encoded => FND_API.G_
        FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```
import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzContactPointV2Pub;

class CreateContactPoint {
    public static void main(String[] args) throws Exception {

        HzContactPointV2Pub.ContactPointRec p_contact_point_rec = new HzContactPointV2Pub.ContactPointRec();
        HzContactPointV2Pub.EdiRec p_edi_rec = new HzContactPointV2Pub.EdiRec();
        HzContactPointV2Pub.EmailRec p_email_rec = new HzContactPointV2Pub.EmailRec();
        HzContactPointV2Pub.PhoneRec p_phone_rec = new HzContactPointV2Pub.PhoneRec();
        HzContactPointV2Pub.TelexRec p_telex_rec = new HzContactPointV2Pub.TelexRec();
        HzContactPointV2Pub.WebRec p_web_rec = new HzContactPointV2Pub.WebRec();
        BigDecimal[] x_contact_point_id = new BigDecimal[1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzContactPointV2Pub contactpointV2Pub = new HzContactPointV2Pub();

            p_contact_point_rec.contact_point_type = "PHONE";
            p_contact_point_rec.owner_table_name = "HZ_PARTIES";
            p_contact_point_rec.owner_table_id = new BigDecimal("1272023");
            p_contact_point_rec.primary_flag = "Y";
            p_contact_point_rec.contact_point_purpose = "BUSINESS";
            p_phone_rec.phone_area_code = "650";
            p_phone_rec.phone_country_code = "1";
            p_phone_rec.phone_number = "506-7000";
            p_phone_rec.phone_line_type = "GEN";
            p_contact_point_rec.created_by_module = "TCA_EXAMPLE";
```

```

contactpointV2Pub.createContactPoint(
    conn
    , "T"
    , p_contact_point_rec
    , p_edi_rec
    , p_email_rec
    , p_phone_rec
    , p_telex_rec
    , p_web_rec
    , x_contact_point_id
    , x_return_status
    , x_msg_count
    , x_msg_data
);

System.out.println( "x_return_status = " + x_return_status[0] );
System.out.println( "x_msg_count = " + x_msg_count[0] );
System.out.println( "x_msg_data = " + x_msg_data[0] );

if (x_msg_count[0].intValue() > 1) {
    OracleCallableStatement ocs = null;
    for (int i=0; i<x_msg_count[0].intValue(); i++) {
        ocs = (OracleCallableStatement)conn.prepareCall(
            "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
        ocs.registerOutParameter(1, OracleTypes.VARCHAR);
        ocs.execute();
        System.out.println((i + 1) + ". " + ocs.getString(1));
    }
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
}

```

### Out Parameters

```

x_contact_point_id = 429523
x_return_status = S
x_msg_count = 0
x_msg_data =

```

### Select Statement to Check Data

```
select contact_point_id, contact_point_type, owner_table_name, owner_table_id, primary_flag,
phone_area_code, phone_country_code, phone_number, phone_line_type, raw_phone_number
from hz_contact_points
where contact_point_id = 429523;
```

The following tables provide information about this select statement.

CONTACT_POINT_ID	CONTACT_POINT_TYPE	OWNER_TABLE_NAME
429523	PHONE	HZ_PARTIES

OWNER_TABLE_ID	PRIMARY_FLAG	PHONE_AREA_CODE
1272023	Y	650

PHONE_COUNTRY_CODE	PHONE_NUMBER	PHONE_LINE_TYPE
1	506-7000	GEN

RAW_PHONE_NUMBER
650-506-7000

### Create an Organization Contact

This procedure assumes that a person with party ID 16077 exists in the database.

#### PL/SQL Example

```
DECLARE
    p_org_contact_rec          HZ_PARTY_CONTACT_V2PUB.ORG_CONTACT_REC_TYPE;
    x_org_contact_id          NUMBER;
    x_party_rel_id            NUMBER;
    x_party_id                NUMBER;
    x_party_number            VARCHAR2(2000);
    x_return_status           VARCHAR2(2000);
    x_msg_count                NUMBER;
    x_msg_data                VARCHAR2(2000);
BEGIN
    p_org_contact_rec.department_code := 'ACCOUNTING';
    p_org_contact_rec.job_title := 'ACCOUNTS OFFICER';
    p_org_contact_rec.decision_maker_flag := 'Y';
```

```

p_org_contact_rec.job_title_code := 'APC';
p_org_contact_rec.created_by_module := 'TCA_EXAMPLE';
p_org_contact_rec.party_rel_rec.subject_id := 16077;
p_org_contact_rec.party_rel_rec.subject_type := 'PERSON';
p_org_contact_rec.party_rel_rec.subject_table_name := 'HZ_PARTIES';
p_org_contact_rec.party_rel_rec.object_id := 1272023;
p_org_contact_rec.party_rel_rec.object_type := 'ORGANIZATION';
p_org_contact_rec.party_rel_rec.object_table_name := 'HZ_PARTIES';
p_org_contact_rec.party_rel_rec.relationship_code := 'CONTACT_OF';
p_org_contact_rec.party_rel_rec.relationship_type := 'CONTACT';
p_org_contact_rec.party_rel_rec.start_date := SYSDATE;

hz_party_contact_v2pub.create_org_contact(
    T,
    p_org_contact_rec,
    x_org_contact_id,
    x_party_rel_id,
    x_party_id,
    x_party_number,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr(x_return_status = 'l|x_return_status,1,255));
dbms_output.put_line(x_msg_count = 'l|TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr(x_msg_data = 'l|x_msg_data,1,255));

IF x_msg_count > 1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('l|l|SubStr(FND_MSG_PUB.Get(p_encoded => FND_API.G_
            FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.TimeStamp;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
```

```

import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzPartyContactV2Pub;

class CreateOrgContact {
    public static void main(String[] args) throws Exception {

        HzPartyContactV2Pub.OrgContactRec p_org_contact_rec = new HzPartyContactV2Pub.OrgContactRec();
        BigDecimal[]      x_org_contact_id = new BigDecimal[1];
        BigDecimal[]      x_party_rel_id = new BigDecimal[1];
        BigDecimal[]      x_party_id = new BigDecimal[1];
        String[]          x_party_number = new String[1];

        String[]          x_return_status = new String[1];
        BigDecimal[]      x_msg_count = new BigDecimal[1];
        String[]          x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartyContactV2Pub partycontactV2Pub = new HzPartyContactV2Pub();

            p_org_contact_rec.department_code = "ACCOUNTING";
            p_org_contact_rec.job_title = "ACCOUNTS OFFICER";
            p_org_contact_rec.decision_maker_flag = "Y";
            p_org_contact_rec.job_title_code = "APC";
            p_org_contact_rec.created_by_module = "TCA_EXAMPLE";
            p_org_contact_rec.party_rel_rec.subject_id = new BigDecimal("16077");
            p_org_contact_rec.party_rel_rec.subject_type = "PERSON";
            p_org_contact_rec.party_rel_rec.subject_table_name = "HZ_PARTIES";
            p_org_contact_rec.party_rel_rec.object_id = new BigDecimal("1272023");
            p_org_contact_rec.party_rel_rec.object_type = "ORGANIZATION";
            p_org_contact_rec.party_rel_rec.object_table_name = "HZ_PARTIES";
            p_org_contact_rec.party_rel_rec.relationship_code = "CONTACT_OF";
            p_org_contact_rec.party_rel_rec.relationship_type = "CONTACT";
            p_org_contact_rec.party_rel_rec.start_date = Timestamp.valueOf("2001-09-26 00:00:00.0");

            partycontactV2Pub.createOrgContact(
                conn
                , "T"
                , p_org_contact_rec
                , x_org_contact_id
    
```

```

        , x_party_rel_id
        , x_party_id
        , x_party_number
        , x_return_status
        , x_msg_count
        , x_msg_data
    );

    System.out.println( "x_return_status = " + x_return_status[0] );
    System.out.println( "x_msg_count = " + x_msg_count[0] );
    System.out.println( "x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? = fnd_msg_pub.get( p_encoded => 'F' ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + ". " + ocs.getString(1));
        }
    }

    conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
}

```

### Out Parameters

```

x_org_contact_id = 411650
x_party_rel_id = 10091074
x_party_id = 1272029
x_party_number = 1268627
x_return_status = S
x_msg_count = 0
x_msg_data =

```

### Select Statement to Check Data

Organization contact record is created.

```
select org_contact_id, party_relationship_id, department_code, title, job_title, job_title_code
from hz_org_contacts
where org_contact_id = 411650;
```

The following tables provide information about this select statement.

ORG_CONTACT_ID	PARTY_RELATIONSHIP_ID	DEPARTMENT_CODE	TITLE
411650	10091074	ACCOUNTING	DR.

JOB_TITLE	JOB_TITLE_CODE
ACCOUNTS OFFICER	APC

Two relationship records are created.

```
select relationship_id, subject_id, subject_type, subject_table_name,
       object_id, object_type, object_table_name, directional_flag
       relationship_type, relationship_code, party_id
from hz_relationships
where relationship_id = 10091074;
```

The following tables provide information about this select statement.

RELATIONSHIP_ID	SUBJECT_ID	SUBJECT_TYPE	SUBJECT_TABLE_NAME
10091074	16077	PERSON	HZ_PARTIES
10091074	1272023	ORGANIZATION	HZ_PARTIES

OBJECT_ID	OBJECT_TYPE	OBJECT_TABLE_NAME	DIRECTIONAL_FLAG
1272023	ORGANIZATION	HZ_PARTIES	F
16077	PERSON	HZ_PARTIES	B

RELATIONSHIP_TYPE	RELATIONSHIP_CODE	PARTY_ID
CONTACT	CONTACT_OF	1272029
CONTACT	CONTACT	1272029

A party record is created.

```
select party_id, party_number, party_name, party_type
from hz_parties
where party_id = 1272029;
```

The following table provides information about this select statement.

<b>PARTY_ID</b>	<b>PARTY_NUMBER</b>	<b>PARTY_NAME</b>	<b>PARTY_TYPE</b>
1272029	1268627	John Doe -ABC Corporation-1268627	PARTY_RELATIONSHIP

## Update a Location Record

This procedure updates the new location and sets address2 to null and county to 'San Mateo'.

### PL/SQL Example

To update an attribute to null in the PL/SQL API, you must set the attribute's value to FND\_API.G\_MISS\_XXX.

```
DECLARE
    p_location_rec          HZ_LOCATION_V2PUB.LOCATION_REC_TYPE;
    p_object_version_number NUMBER;
    x_return_status        VARCHAR2(2000);
    x_msg_count            NUMBER;
    x_msg_data             VARCHAR2(2000);
BEGIN
    p_location_rec.location_id := 359086;
    p_location_rec.address2 := FND_API.G_MISS_CHAR;
    p_location_rec.county := 'San Mateo';
    -- get the object version number of the current record
    select object_version_number
    into p_object_version_number
    from hz_locations
    where location_id = 359086;

    hz_location_v2pub.update_location(
        T,
        p_location_rec,
        p_object_version_number,
        x_return_status,
        x_msg_count,
        x_msg_data);
```

```

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
API.G_FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

To update an attribute to null, you must set the attribute's value to null. This procedure is different from that of the PL/SQL API.

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzLocationV2Pub;
import oracle.apps.ar.hz.v2api.HzConstant;

class UpdateLocation {
    public static void main(String[] args) throws Exception {

        HzLocationV2Pub.LocationRec p_location_rec = new HzLocationV2Pub.LocationRec();

        String[]          x_return_status = new String[1];
        BigDecimal[]      x_msg_count    = new BigDecimal[1];
        String[]          x_msg_data    = new String[1];
        BigDecimal[]      p_object_version_number = new BigDecimal[1];

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");
```

```
HzLocationV2Pub locationV2Pub = new HzLocationV2Pub();

p_location_rec.location_id = new BigDecimal("359086");
p_location_rec.address2 = null;
p_location_rec.county = "San Mateo";

-- please note, you should pass the current value of object
-- version number column for this record
p_object_version_number[0] = new BigDecimal("1");

locationV2Pub.updateLocation(
    conn
    , "T"
    , p_location_rec
    , p_object_version_number
    , x_return_status
    , x_msg_count
    , x_msg_data
);

System.out.println( "x_return_status = " + x_return_status[0] );
System.out.println( "x_msg_count = " + x_msg_count[0] );
System.out.println( "x_msg_data = " + x_msg_data[0] );

if (x_msg_count[0].intValue() > 1) {
    OracleCallableStatement ocs = null;
    for (int i=0; i<x_msg_count[0].intValue(); i++) {
        ocs = (OracleCallableStatement)conn.prepareCall(
            "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
        ocs.registerOutParameter(1, OracleTypes.VARCHAR);
        ocs.execute();
        System.out.println((i + 1) + ". " + ocs.getString(1));
    }
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
```

### Select Statement to Check Data

```
select address1, address2, county, object_version_number
from hz_locations
where location_id = 359086;
```

The following table provides information about this select statement.

ADDRESS1	ADDRESS2	COUNTY	OBJECT_VERSION_NUMBER
300 Oracle Parkway		San Mateo	2

## Create a Customer Account

### PL/SQL Example

```
DECLARE
    p_cust_account_rec          HZ_CUST_ACCOUNT_V2PUB.CUST_ACCOUNT_REC_TYPE;
    p_person_rec                HZ_PARTY_V2PUB.PERSON_REC_TYPE;
    p_customer_profile_rec      HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE;
    x_cust_account_id           NUMBER;
    x_account_number            VARCHAR2(2000);
    x_party_id                  NUMBER;
    x_party_number              VARCHAR2(2000);
    x_profile_id                NUMBER;
    x_return_status              VARCHAR2(2000);
    x_msg_count                  NUMBER;
    x_msg_data                   VARCHAR2(2000);
BEGIN
    p_cust_account_rec.account_name := 'John's A/c';
    p_cust_account_rec.created_by_module := 'TCA_EXAMPLE';
    p_person_rec.person_first_name := 'John';
    p_person_rec.person_last_name := 'Smith';

    hz_cust_account_v2pub.create_cust_account(
        T,
        p_cust_account_rec,
        p_person_rec,
        p_customer_profile_rec,
        F,
        x_cust_account_id,
        x_account_number,
        x_party_id,
        x_party_number,
        x_profile_id,
        x_return_status,
```

```

        x_msg_count,
        x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
        API.G_FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzCustAccountV2Pub;
import oracle.apps.ar.hz.v2api.HzPartyV2Pub;
import oracle.apps.ar.hz.v2api.HzCustomerProfileV2Pub;

class CreateCustAccount {
    public static void main(String[] args) throws Exception {

        HzCustAccountV2Pub.CustAccountRec p_cust_account_rec = new HzCustAccountV2Pub.CustAccountRec();
        HzPartyV2Pub.PersonRec p_person_rec = new HzPartyV2Pub.PersonRec();
        HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec = new
        HzCustomerProfileV2Pub.CustomerProfileRec();
        BigDecimal[]    x_cust_account_id = new BigDecimal[1];
        String[]        x_account_number = new String[1];
        BigDecimal[]    x_party_id = new BigDecimal[1];
        String[]        x_party_number = new String[1];
        BigDecimal[]    x_profile_id = new BigDecimal[1];

        String[]        x_return_status = new String[1];
        BigDecimal[]    x_msg_count = new BigDecimal[1];
        String[]        x_msg_data = new String[1];
```

```

try {
    Class.forName ("oracle.jdbc.driver.OracleDriver");
    OracleConnection conn = (OracleConnection)
        DriverManager.getConnection (
            "jdbc:oracle:thin:@ap103fam:1521:findv115",
            "apps", "apps");

    HzCustAccountV2Pub custaccountV2Pub = new HzCustAccountV2Pub();

    p_cust_account_rec.account_name = "John A/c";
    p_person_rec.person_first_name = "John";
    p_person_rec.person_last_name = "Smith";
    p_cust_account_rec.created_by_module = "TCA_EXAMPLE";

    custaccountV2Pub.createCustAccount(
        conn
        , "T"
        , p_cust_account_rec
        , p_person_rec
        , p_customer_profile_rec
        , "F"
        , x_cust_account_id
        , x_account_number
        , x_party_id
        , x_party_number
        , x_profile_id
        , x_return_status
        , x_msg_count
        , x_msg_data
    );

    System.out.println( "x_return_status = " + x_return_status[0] );
    System.out.println( "x_msg_count = " + x_msg_count[0] );
    System.out.println( "x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + ". " + ocs.getString(1));
        }
    }
}

```

```

    }

    conn.close();
  } catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
  } catch (SQLException e) {
    System.out.println("SQL Error: " + e);
  }
}
}
}

```

### Out Parameters

```

x_cust_account_id = 3472
x_account_number = 1745
x_party_id = 3575
x_party_number = 1647
x_profile_id = 1483
x_return_status = S
x_msg_count = 0
x_msg_data =

```

### Select Statement to Check Data

Created a record in the HZ\_CUST\_ACCOUNTS table.

```

select cust_account_id, party_id, account_number, account_name
from hz_cust_accounts
where cust_account_id = 3472;

```

The following table provides information about this select statement.

CUST_ACCOUNT_ID	PARTY_ID	ACCOUNT_NUMBER	ACCOUNT_NAME
3472	3575	1745	John's A/c

Created a record in the HZ\_PARTIES table

```

select party_id, party_number, party_name, party_type
from hz_parties
where party_id = 3575;

```

The following table provides information about this select statement.

PARTY_ID	PARTY_NUMBER	PARTY_NAME	PARTY_TYPE
3575	1647	John Smith	PERSON

Created a record in hz\_person\_profiles table.

```
select person_profile_id, party_id, person_first_name, person_last_name
from hz_person_profiles
where party_id = 3575;
```

The following table provides information about this select statement.

PERSON_PROFILE_ID	PARTY_ID	PERSON_FIRST_NAME	PERSON_LAST_NAME
1483	3575	John	Smith

Created a record in hz\_customer\_profiles.

```
select cust_account_profile_id, cust_account_id, profile_class_id, collector_id
from hz_customer_profiles
where cust_account_id = 3472;
```

The following table provides information about this select statement.

CUST_ACCOUNT_PROFILE_ID	CUST_ACCOUNT_ID	PROFILE_CLASS_ID	COLLECTOR_ID
3691	3472	0	1000

## Create a Customer Account Site

This procedure creates a site for the account created in previous example, using an existing site.

### PL/SQL Example

```
DECLARE
    p_cust_acct_site_rec hz_cust_account_site_v2pub.cust_acct_site_rec_type;
    x_return_status      VARCHAR2(2000);
    x_msg_count          NUMBER;
    x_msg_data           VARCHAR2(2000);
    x_cust_acct_site_id  NUMBER;
BEGIN
    p_cust_acct_site_rec.cust_account_id := 3472;
    p_cust_acct_site_rec.party_site_id := 1024;
```

```

p_cust_acct_site_rec.language := 'US';
p_cust_acct_site_rec.created_by_module := 'TCA-EXAMPLE';

hz_cust_account_site_v2pub.create_cust_acct_site(
    T,
    p_cust_acct_site_rec,
    x_cust_acct_site_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count > 1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
            APLG_FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzCustAccountSiteV2Pub;

class CreateCustAcctSite {
    public static void main(String[] args) throws Exception {

        HzCustAccountSiteV2Pub.CustAcctSiteRec p_cust_acct_site_rec = new
        HzCustAccountSiteV2Pub.CustAcctSiteRec();
        BigDecimal[]          x_cust_account_site_id = new BigDecimal[1];

        String[]              x_return_status = new String[1];
        BigDecimal[]          x_msg_count = new BigDecimal[1];
        String[]              x_msg_data = new String[1];
```

```

try {
    Class.forName ("oracle.jdbc.driver.OracleDriver");
    OracleConnection conn = (OracleConnection)
        DriverManager.getConnection (
            "jdbc:oracle:thin:@ap103fam:1521:findv115",
            "apps", "apps");

    HzCustAccountSiteV2Pub custaccountsiteV2Pub = new HzCustAccountSiteV2Pub();

    p_cust_acct_site_rec.cust_account_id = new BigDecimal("3472");
    p_cust_acct_site_rec.party_site_id = new BigDecimal("1024");
    p_cust_acct_site_rec.language = "US";
    p_cust_acct_site_rec.created_by_module = "TCA_EXAMPLE";

    custaccountsiteV2Pub.createCustAcctSite(
        conn
        , "T"
        , p_cust_acct_site_rec
        , x_cust_account_site_id
        , x_return_status
        , x_msg_count
        , x_msg_data
    );

    System.out.println("x_return_status = " + x_return_status[0] );
    System.out.println("x_msg_count = " + x_msg_count[0] );
    System.out.println("x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + ". " + ocs.getString(1));
        }
    }

    conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {

```

```

        System.out.println("SQL Error." + e);
    }
}
}

```

### Out parameters

```

x_return_status = S
x_msg_count = 0
x_msg_data =
x_cust_acct_site_id = 3580

```

### Select statement to check data

```

select cust_acct_site_id, cust_account_id, party_site_id, org_id
from hz_cust_acct_sites_all
where cust_acct_site_id = 3580;

```

The following table provides information about this select statement.

CUST_ACCT_SITE_ID	CUST_ACCOUNT_ID	PARTY_SITE_ID	ORG_ID
3580	3472	1024	204

## Create a Customer Account Site Use

This procedure creates a customer account site use for the customer account site created in the previous example.

### PL/SQL Example

```

DECLARE
    p_cust_site_use_rec          HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_SITE_USE_
                                REC_TYPE;
    p_customer_profile_rec       HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_
                                REC_TYPE;
    x_site_use_id                NUMBER;
    x_return_status              VARCHAR2(2000);
    x_msg_count                  NUMBER;
    x_msg_data                   VARCHAR2(2000);
BEGIN
    p_cust_site_use_rec.cust_acct_site_id := 3580;
    p_cust_site_use_rec.site_use_code := INV;
    p_cust_site_use_rec.location := 'TCA';
    p_cust_site_use_rec.created_by_module := 'TCA_EXAMPLE';

```

```

hz_cust_account_site_v2pub.create_cust_site_use(
    'T',
    p_cust_site_use_rec,
    p_customer_profile_rec,
    ',
    ',
    x_site_use_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('!!! '||SubStr(FND_MSG_PUB.Get(p_encoded => FND_
        API.G_FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.apps.ar.hz.v2api.HzCustAccountSiteV2Pub;
import oracle.apps.ar.hz.v2api.HzCustomerProfileV2Pub;;

class CreateCustSiteUse {
    public static void main(String[] args) throws Exception {

        HzCustAccountSiteV2Pub.CustSiteUseRec p_cust_site_use_rec = new HzCustAccountSiteV2Pub.CustSiteUseRec();
        HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec = new
        HzCustomerProfileV2Pub.CustomerProfileRec();
        BigDecimal[] x_site_use_id = new BigDecimal[1];
```

```

String[]          x_return_status = new String[1];
BigDecimal[]     x_msg_count = new BigDecimal[1];
String[]         x_msg_data = new String[1];

try {
    Class.forName ("oracle.jdbc.driver.OracleDriver");
    OracleConnection conn = (OracleConnection)
        DriverManager.getConnection (
            "jdbc:oracle:thin:@ap103fam:1521:findv115",
            "apps", "apps");

    HzCustAccountSiteV2Pub custaccountsiteV2Pub = new HzCustAccountSiteV2Pub();

    p_cust_site_use_rec.cust_acct_site_id = new BigDecimal("3580");
    p_cust_site_use_rec.site_use_code = "INV";
    p_cust_site_use_rec.location = "TCA";
    p_cust_site_use_rec.created_by_module = "TCA_EXAMPLE";

    custaccountsiteV2Pub.createCustSiteUse(
        conn
        , "T"
        , p_cust_site_use_rec
        , p_customer_profile_rec
        , "T"
        , "T"
        , x_site_use_id
        , x_return_status
        , x_msg_count
        , x_msg_data
    );

    System.out.println( "x_return_status = " + x_return_status[0] );
    System.out.println( "x_msg_count = " + x_msg_count[0] );
    System.out.println( "x_msg_data = " + x_msg_data[0] );

    if (x_msg_count[0].intValue() > 1) {
        OracleCallableStatement ocs = null;
        for (int i=0; i<x_msg_count[0].intValue(); i++) {
            ocs = (OracleCallableStatement)conn.prepareCall(
                "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
            ocs.registerOutParameter(1, OracleTypes.VARCHAR);
            ocs.execute();
            System.out.println((i + 1) + ". " + ocs.getString(1));
        }
    }
}

```

```

    conn.close();
  } catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
  } catch (SQLException e) {
    System.out.println("SQL Error: " + e);
  }
}
}
}

```

### Out Parameters

```

x_site_use_id = 3756
x_return_status = S
x_msg_count = 0
x_msg_data =

```

### Select Statement to Check Data

```

select site_use_id, cust_acct_site_id, site_use_code, location, org_id
from hz_cust_site_uses_all
where site_use_id = 3756;

```

The following table provides information about this select statement.

SITE_USE_ID	CUST_ACCT_SITE_ID	SITE_USE_CODE	LOCATION	ORG_ID
3756	3580	INV	TCA	204

## Update a Customer Account Record

This example updates the customer account record that was created earlier by setting the `account_name` column to null and the `customer_type` column to *R*.

### PL/SQL Example

To update an attribute to null in the PL/SQL API, you need to set the attribute's value to `FND_API.G_MISS_XXX`.

```

DECLARE
    p_cust_account_rec          HZ_CUST_ACCOUNT_V2PUB.CUST_ACCOUNT_
                                REC_TYPE;
    p_object_version_number    NUMBER;
    x_return_status            VARCHAR2(2000);
    x_msg_count                NUMBER;
    x_msg_data                  VARCHAR2(2000);
BEGIN

```

```

p_cust_account_rec.cust_account_id := 3472;
p_cust_account_rec.customer_type := 'R';
p_cust_account_rec.account_name := FND_API.G_MISS_CHAR;

-- get the object version number of the current record
select object_version_number
into p_object_version_number
from hz_cust_accounts
where cust_account_id = 3472;

hz_cust_account_v2pub.update_cust_account(
    T',
    p_cust_account_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data);

dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

IF x_msg_count > 1 THEN
    FOR I IN 1..x_msg_count
    LOOP
        dbms_output.put_line('||SubStr(FND_MSG_PUB.Get(p_encoded => FND_API.G_
        FALSE ), 1, 255));
    END LOOP;
END IF;

END;
```

### Java Example

You must set the attribute's value to null to update an attribute to null. This procedure is different from that of the PL/SQL API.

```

import java.math.BigDecimal;
import java.sql.Timestamp;
import java.sql.DriverManager;
import java.sql.SQLException;
import oracle.jdbc.driver.OracleConnection;
import oracle.jdbc.driver.OracleCallableStatement;
import oracle.jdbc.driver.OracleTypes;
import oracle.jdbc.driver.OracleResultSet;
import oracle.apps.ar.hz.v2api.HzCustAccountV2Pub;
```

```

class UpdateCustAccount {
    public static void main(String[] args) throws Exception {

        HzCustAccountV2Pub.CustAccountRec p_cust_account_rec = new HzCustAccountV2Pub.CustAccountRec();

        String[]          x_return_status = new String[1];
        BigDecimal[]      x_msg_count = new BigDecimal[1];
        String[]          x_msg_data = new String[1];
        BigDecimal[]      p_object_version_number = new BigDecimal[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzCustAccountV2Pub custAccountV2Pub = new HzCustAccountV2Pub();

            p_cust_account_rec.cust_account_id = new BigDecimal("3472");
            p_cust_account_rec.customer_type = "R";
            p_cust_account_rec.account_name = null;

            -- please note, you should pass the current value of object
            -- version number column for this record
            p_object_version_number [0]= new BigDecimal("1");

            custAccountV2Pub.updateCustAccount(
                conn
                , "F"
                , p_cust_account_rec
                , p_object_version_number
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {

```

```

    ocs = (OracleCallableStatement)conn.prepareCall(
        "begin ? := fnd_msg_pub.get( p_encoded => 'F'); end;");
    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
    ocs.execute();
    System.out.println((i + 1) + ". " + ocs.getString(1));
}
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
}

```

### Out Parameters

```

    p_object_version_number = 2
    x_return_status = S
    x_msg_count = 0
    x_msg_data =

```

### Select Statement to Check Data

```

select cust_account_id, customer_type, account_name
from hz_cust_accounts
where cust_account_id = 3472;

```

The following table provides information about this select statement.

CUST_ACCOUNT_ID	CUSTOMER_TYPE	ACCOUNT_NAME
3472	R	

## Difference Between Version 1.0 and Version 2.0 Public API for Update Procedures

This section is specifically for Oracle internal users who have developed codes using old TCA PL/SQL API. The section demonstrates what change should be made to the existing code for uptaking Public API. There is no difference between the Version 1.0 and Version 2.0 Java APIs from the coding perspective.

### Update ECE\_TP\_LOCATION\_CODE to Null through Version 1.0 API

You must set the corresponding field in the record structure to null in order to set this column to null.

```
DECLARE
    account_site_rec          HZ_CUSTOMER_ACCOUNTS_PUB.ACCT_SITE_REC_
                               TYPE;
    x_return_status           VARCHAR2(100);
    x_msg_count               NUMBER;
    x_msg_data                VARCHAR2(2000);

    l_last_update_date        DATE := sysdate;
    l_location_code           VARCHAR2(100);
BEGIN
    UPDATE hz_cust_acct_sites
    SET ece_tp_location_code = 'Location Code'
    WHERE cust_acct_site_id = 1001;

    SELECT ece_tp_location_code into l_location_code
    FROM hz_cust_acct_sites
    WHERE cust_acct_site_id = 1001;

    dbms_output.put_line('Before Update : ece_tp_location_code = ' || l_location_code );

    account_site_rec.cust_acct_site_id := 1001;
    account_site_rec.ece_tp_location_code := NULL;

    SELECT last_update_date INTO l_last_update_date
    FROM hz_cust_acct_sites
    WHERE cust_acct_site_id = 1001;

    hz_customer_accounts_pub.update_acct_site(
    l,
    'T');
```

```

F,
account_site_rec,
l_last_update_date,
x_return_status,
x_msg_count,
x_msg_data);

SELECT ece_tp_location_code into l_location_code
FROM hz_cust_acct_sites
WHERE cust_acct_site_id = 1001;

dbms_output.put_line( 'After Update: ece_tp_location_code = '|| l_location_code );
END;
```

### Update ECE\_TP\_LOCATION\_CODE to Null through Version 2.0 API

You must set the corresponding field in the record structure to FND\_API.G\_MISS\_XXX in order to set this column to null.

```

DECLARE
    account_site_rec                HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_ACCT_
                                   SITE_REC_TYPE;
    x_return_status                 VARCHAR2(100);
    x_msg_count                     NUMBER;
    x_msg_data                      VARCHAR2(2000);
    l_object_version_number         NUMBER;
    l_location_code                 VARCHAR2(100);
BEGIN
    UPDATE hz_cust_acct_sites
    SET ece_tp_location_code = 'Location Code'
    WHERE cust_acct_site_id = 1001;

    SELECT ece_tp_location_code into l_location_code
    FROM hz_cust_acct_sites
    WHERE cust_acct_site_id = 1001;

    dbms_output.put_line( 'Before Update : ece_tp_location_code = '|| l_location_code );

    account_site_rec.cust_acct_site_id := 1001;
    account_site_rec.ece_tp_location_code := FND_API.G_MISS_CHAR;

    SELECT object_version_number INTO l_object_version_number
    FROM hz_cust_acct_sites
    WHERE cust_acct_site_id = 1001;
```

```
hz_cust_account_site_v2pub.update_cust_acct_site(  
  T,  
  account_site_rec,  
  l_object_version_number,  
  x_return_status,  
  x_msg_count,  
  x_msg_data);  
  
  SELECT ece_tp_location_code into l_location_code  
  FROM hz_cust_acct_sites  
  WHERE cust_acct_site_id = 1001;  
  dbms_output.put_line( 'After Update: ece_tp_location_code = ' || l_location_code );  
END;
```

# B

---

---

## List of Messages

This appendix describes messages that may be raised during the processing of API calls. For each message the number, code, and text of the message are provided.

## Messages for the TCA APIs

Messages play an important role in the effectiveness of API calls. The appropriate messages must be raised at the right points in the processing to accurately describe the error that has occurred. In the TCA APIs all error messages are put on the message stack during execution and can be retrieved by the caller.

This table displays a list of messages that are raised by different APIs.

Number	Code	Message
96242	HZ_API_DUPLICATE_COLUMN	Value for &COLUMN must be unique.
96243	HZ_API_MISSING_COLUMN	Column &COLUMN must have a value.
96244	HZ_API_INVALID_LOOKUP	Value for &COLUMN must be a value defined in lookup type &LOOKUP_TYPE.
96245	HZ_API_RECORD_CHANGED	This record in table &TABLE cannot be locked as it has been updated by another user.
96246	HZ_API_NONUPDATEABLE_COLUMN	You cannot update column &COLUMN.
96247	HZ_API_NO_RECORD	No &RECORD was found for ID &VALUE.
96248	HZ_API_START_DATE_GREATER	End date cannot be earlier than the start date.
96249	HZ_API_INVALID_FK	Invalid value for &FK. Please enter &COLUMN value from &TABLE.
96250	HZ_API_SUBJECT_OBJECT_IDS	Subject ID and object ID cannot refer to the same party ID.
96271	HZ_API_PARTY_NUMBER_AUTO_ON	You cannot pass the party number because the profile option HZ:Generate Party Number is either null or is set to Yes.

<b>Number</b>	<b>Code</b>	<b>Message</b>
96272	HZ_API_PARTY_SITE_NUM_AUTO_ON	You cannot pass the party site number because the profile option HZ:Generate Party Site Number is either null or is set to Yes.
96273	HZ_API_ACCOUNT_NUMBER_AUTO_ON	You cannot pass the account number because account number auto-generation is enabled.
96274	HZ_API_DATE_GREATER	&DATE2 cannot be earlier than &DATE1.
96277	HZ_API_NONUPDATEABLE_TO_NULL	You cannot update column &COLUMN to null.
96278	HZ_API_OTHERS_EXCEP	The following SQL error occurred:
96279	HZ_API_HOOK_ERROR	The following error occurred in procedure &PROCEDURE:
96291	HZ_API_UNIQUE_PRIMARY_ROLE	A Primary role is already defined for this org contact.
96292	HZ_API_UNIQUE_ROLE_TYPE	A role of this type is already defined for this org contact.
96695	HZ_API_UNIQUE_SITE_USE_TYPE	A site use of this type is already defined for this party site.
96814	HZ_PHONE_FORMAT_NOT_DEFINED	You have passed a Raw Phone Number and the Country Code. No Phone Format could be found. Please check the Phone Format setup.
96815	HZ_COUNTRY_CODE_NOT_DEFINED	You have passed a Raw Phone Number without a Country Code or the Country Code you have passed does not exist.
96816	HZ_INVALID_PHONE_PARAMETER	Either a Raw Phone Number or the Area Code and Phone Number should be passed.
96820	HZ_API_SIC_CODE_TYPE_REQUIRED	Please set both the sic_code and sic_code_type columns if one of these columns already has a value.

<b>Number</b>	<b>Code</b>	<b>Message</b>
96821	HZ_API_SIC_CODE_TYPE_OTHER	This value for the column sic_code_type is allowed only if the column content_source_type has the value USER_ENTERED.
96965	HZ_API_UNIQUE_PRIMARY_ORG_CONT	For this party, a primary contact is already defined for this type of role.
294005	HZ_API_LEAF_ONLY_NOT_ALLOWED	One or more associations use the parent codes of &CLASS_CATEGORY class category.
294006	HZ_API_SIN_PAR_NOT_ALLOWED	The &CLASS_CATEGORY category cannot update because the &CLASS_CODE3 class code in the &CLASS_CATEGORY category cannot have multiple parents during a specific date range. The &CLASS_CATEGORY category has parent class codes &CLASS_CODE1 from &START1 to &END
294007	HZ_API_SIN_ASS_NOT_ALLOWED	You cannot have any records with more than one class code from the &CLASS_CATEGORY category during a specific date range. The &OWNER_TABLE_ID ID from the &OWNER_TABLE table from the &CONTENT_SOURCE_TYPE source has class codes &CLASS_CODE1 from &START1 to
294009	HZ_API_MULTI_PARENT_FORBID	The &CLASS_CODE3 class code cannot be a child of the &CLASS_CODE2 class code because the &CLASS_CATEGORY category does not allow multiple parents during a specific date range. The &CLASS_CODE2 class code is a child of the &CLASS_CODE1 class code in the &

<b>Number</b>	<b>Code</b>	<b>Message</b>
294014	HZ_API_USE_TAB_CAT	There is no association between &OWNER_TABLE owner table and &CLASS_CATEGORY class category in HZ_CLASS_CATEGORY_USES. Please contact the applications administrator.
294015	HZ_API_CLA_CAT_WHERE	The &OWNER_TABLE_ID record ID from the &OWNER_TABLE owner table cannot be validated against the database. Most of the time
294065	HZ_INVALID_DIRECTION_CODE	If the forward and backward relationship codes are the same, the direction code must be N which indicates a non-directional relationship.
294066	HZ_NO_REL_TYPE	This relationship type is not valid for the given subject type and object type.
294067	HZ_API_INVALID_COMBINATION	Backward relationship code must be the same for all records of a specified combination of forward relationship code
294068	HZ_INVALID_DIRECTION_CODE1	If the forward relationship code and the backward relationship code are the same, then the direction code must be N (Non-directional).
294069	HZ_INVALID_DIRECTION_CODE2	If the forward relationship code and the backward relationship code are different, then the direction code must be either P (Parent) or C(Child).
294070	HZ_VALUES_NOT_SUPPORTED	You cannot set the &FIELD to &VALUE. This option is not currently available.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294072	HZ_RELATIONSHIP_DATE_OVERLAP	You cannot create a relationship for a time period that overlaps the time period of an existing relationship with the same subject id
294112	HZ_MULTIPLE_PREFERENCES	You cannot pass multiple preference values. Please pass only one preference value from one of the VALUE_VARCHAR2
294113	HZ_NO_PREFERENCE	You must pass at least one preference value. Please pass one preference value from one of the VALUE_VARCHAR2
294114	HZ_SINGLE_VALUE_PREFERENCE	You cannot create multiple preference values for a single-value preference code.
294115	HZ_API_INACTIVE_NOT_PRIMARY	An inactive contact point cannot be the primary contact point.
294218	HZ_API_NO_DEFAULT_PROF_CLASS	Cannot find the default customer profile class (profile class ID = 0). Please use a different profile class.
294219	HZ_API_INACTIVE_PROFILE_CLASS	The customer profile class &NAME is inactive. Please select another profile class for the customer profile.
294220	HZ_API_COLUMN_SHOULD_BE_NULL	The &COLUMN column in the &TABLE table must be null. Please change the value of the column to null.
294221	HZ_API_OBSOLETE_COLUMN	The &COLUMN column is obsolete. You are not allowed to pass a value to this column.
294222	HZ_API_PARTY_OBJECT_MISMATCH	For a contact, the party ID of the party site must be the same as the object ID of the relationship. Please change either the party site or the object ID.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294223	HZ_API_NEGATIVE_PROFILE_CLASS	The profile class ID must be positive. Please enter a positive number for the profile class ID.
294224	HZ_API_BILL_TO_SITE_USE_F	The site use referenced by this bill_to_site_use_id must be active bill-to
294225	HZ_API_BILL_TO_SITE_USE_S	You can only assign bill_to_site_use_id when the current account site use purpose is ship-to.
294227	HZ_API_ACCT_SITEUSE_MISMATCH	The site use must belong to the customer account.
294228	HZ_API_ACCT_SITE_MISMATCH	The site must belong to the customer account.
294229	HZ_API_PROF_AMT_IDS_MISMATCH	The &ENTITY ID of the profile amount must be the same as the &ENTITY ID of the customer profile.
294230	HZ_CUST_ACCT_ROLE_PRIMARY	A primary account role already exists for this &ENTITY. You cannot make this account role primary.
294232	HZ_INVALID_CONTACT_TYPE	The &INCORRECT_CONTACT_TYPE contact type can not be assigned to the contact point of type &CONTACT_POINT_TYPE. Please assign the &CORRECT_CONTACT_TYPE to this contact point type.
294233	HZ_INVALID_CONTACT_LEVEL_TABLE	The contact level table for the &CONTACT_TYPE type of contact can not be &CONTACT_LEVEL_TABLE. Please assign a valid contact level table to the &CONTACT_TYPE type of contact.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294237	HZ_API_DUPLICATE_RECORD	The time range of this preference overlaps with the existing preference &COLUMN=&ID in the &TABLE table. Please enter a different preference or update the existing preference (&COLUMN=&ID).
294238	HZ_NONUPDATEABLE_PREF_DATE	Dates cannot be updated to a time earlier than the current date. Please enter a date after &SYS_DATE.
294239	HZ_CP_REC_NOT_FOUND	Updates to this field require that an update be made to a corresponding record in the HZ_CONTACT_PREFERENCES table. This field cannot be updated because no record in the HZ_CONTACT_PREFERENCES table corresponds to record (&PRIMARY_KEY= &ID) in the &SOURC
294249	HZ_API_ONE_NATIVE_PER_PARTY	Each party can have only one native language. Please change any existing native language to non-native before making this the native language
294250	HZ_API_PRIMARY_PER_ENTITY	Each &ENTITY2 can have only one primary &ENTITY1. Please change any existing primary &ENTITY1 to non-primary before making this record primary
294251	HZ_API_SIZE_ERROR	The size of the &COLUMN column cannot be more than &SIZE characters. Please pass a value of &SIZE or fewer characters.
294252	HZ_API_NONSUPPORT_COLUMN	The &COLUMN column is reserved for future use.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294253	HZ_API_INACTIVE_CANNOT_PRIM	An inactive &ENTITY cannot be the primary &ENTITY. Please change the &COLUMN1 to N or change &COLUMN2 to active.
294283	HZ_DQM_API_ERROR	Unexpected SQL error encountered when performing search. Procedure: &PROC Error: &ERROR. Please contact the system administrator.
294284	HZ_NO_MATCH_RULE	No match rule was specified for performing the search. Please pass match rule name from the application or from user profile defaults.
294285	HZ_INVALID_MATCH_RULE	The match rule specified for search is invalid. Please confirm if the match rule exists and if it has been compiled.
294286	HZ_MATCH_RULE_TX_NOT_STAGED	One or more transformations used by the match rule have not been staged. Please confirm that all required transformations have been staged.
294287	HZ_NO_PRIMARY_COND	Search cannot be executed. Please enter a value for at least one of the required search criteria.
294288	HZ_SEARCH_CRIT_LARGE_ERROR	Search cannot be executed. The length of the interMedia query string for the &ENTITY entity is too long. Please enter an interMedia query string equal to or fewer than 4000 characters.
294289	HZ_MAX_SECONDARY_EXCEEDED	Too many Scoring attributes have been defined for the match rule. Please reduce the number of Scoring attributes to equal to or fewer than 50.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294290	HZ_DQM_NO_SCORE_DETAILS	Score details for this party is not available. Please execute the search again in the same database session.
294291	HZ_TRANSFORM_PROC_ERROR	Unexpected error encountered when executing the transformation procedure: Procedure: &PROC Error: &ERROR Please contact the system administrator.
294292	HZ_NO_CONTACT_POINT_TYPE	Contact point type is NULL in the contact point search variable. You must enter a contact point type for each contact point search record.
294293	HZ_PARTY_QUERY_ERROR	Unexpected error encountered when querying party details for party (ID: &PARTY_ID): Procedure: &PROC Error: &ERROR Please contact the system administrator.
294294	HZ_RESTRICT_SQL_ERROR	Unexpected error encountered when executing SQL restriction clause from p_restrict_sql: Procedure: &PROC Error: &ERROR Please examine the SQL restriction clause.
294307	HZ_DQM_TRANSFORMATION_ERROR	Unexpected SQL error encountered in transformation function: Procedure: &PROC Error: &ERROR Please contact the system administrator.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294340	HZ_CREATED_BY_MISMATCH	You do not have permission to update this record. Please contact an Oracle Applications user who has permission to update human resources records.
294371	HZ_API_CPF_NON_UPDATEABLE	You cannot update the party creation setting for this relationship type because relationships were already created with the current setting.
294376	HZ_API_VALUE_BETWEEN	Please enter a value for &COLUMN between &VALUE1 and &VALUE2, inclusive.
294377	HZ_API_VAL_DEP_FIELDS	The value of &COLUMN1 is &VALUE1. The value of &COLUMN2 must be &VALUE2.
294378	HZ_API_MAND_DEP_FIELDS	The value of &COLUMN1 is &VALUE1. Please enter a value for &COLUMN2.
294379	HZ_API_VAL_INT_CHARGES_Y	To charge interest the number of days in the interest period must be greater than zero. Please enter the number of days in the interest period.
294380	HZ_API_LESS_THAN_ZERO	Please enter a value for &COLUMN equal to or greater than zero.
294381	HZ_API_GREATER_THAN_ZERO	Please enter a value greater than zero for &COLUMN.
294382	HZ_API_VAL_OVERALL_CREDIT	The transaction credit limit must be less than or equal to the overall credit limit. Please change the transaction credit limit or the overall credit limit.
294383	HZ_API_VAL_CREDIT_LIMIT	Please enter a transaction credit limit and an overall credit limit, or make the values for both limits empty.

<b>Number</b>	<b>Code</b>	<b>Message</b>
294389	HZ_REL_NOT_ALLOW_SELF_RELATE	This relationship type does not allow you to relate a party to itself. Please either select a different party as the object of the relationship or select a different relationship type.
294391	HZ_API_NO_ACTIVE_SITE	You cannot inactivate this primary address because no other active party sites exist for this party.
294418	HZ_DIFF_VALUE_NOT_ALLOWED	All &ENTITY records with same &COLUMN1 value must have the same &COLUMN2 value.
294421	HZ_API_MULTIPLE_PARENT	Multiple parents are not permitted for the &RELTYPE relationship type. Please use a different relationship type.
294420	HZ_API_CIRCULAR_REL	Circular relationships are not permitted for the &RELTYPE relationship type. Please use a different relationship type.
294422	HZ_NON_HIER_REL_TYPE	The relationship type passed is not hierarchical. Please use a hierarchical relationship type.
294450	HZ_NOTALLOW_UPDATE_THIRD_PARTY	You do not have permission to update third party data.
294451	HZ_API_CANT_ACTIVATE_SITE	You cannot activate this party site because it was inactivated and replaced by a new party site with a new address from D&B.
294452	HZ_DUP_PARTY_WITH_PARTY_TYPE	A party with ID &PARTY_ID and type &PARTY_TYPE already exists.

---

<b>Number</b>	<b>Code</b>	<b>Message</b>
294458	HZ_DISALLOW_USER_CREATION	You do not have permission to create records for the &ENTITY entity. Please contact your system administrator.
294459	HZ_SST_INVALID_SOURCE	You cannot use the SST data source for this entity. Please update the &COLUMN column with another, valid data source.
294460	HZ_API_SST_NONUPDATEABLE_COL	You do not have permission to overwrite the information in the &COLUMN column, which comes from a third party data source. Please contact your system administrator.
294461	HZ_DNB_MOVED_COLUMN	The &COLUMN column was moved from the HZ_ORGANIZATION_PROFILES table to the HZ_CREDIT_RATINGS table. Please use the Credit Rating API to access these attributes.

---



# C

---

---

## API Mapping

This appendix provides mapping from the first, internal-only version of the API to the second, public version of the TCA API.

## Mapping Version 1 to Version 2 APIs

The following table maps the list of non-public Version 1.0 TCA APIs to the public Version 2.0 TCA APIs.

Version 1 API	Version 2 API	Comments	File Name
HZ_ CLASSIFICATION_ PUB	HZ_ CLASSIFICATION_ V2PUB	All procedure names identical to Version 1	ARH2CLSS/B.pls
HZ_CONTACT_ POINT_PUB  Create/update contact_point	HZ_CONTACT_ POINT_V2PUB  create/update contact_point		ARH2CPSS/B.pls
HZ_CUSTOMER_ ACCOUNTS_PUB	HZ_CUST_ ACCOUNT_V2PUB	Account, account relate  All procedure names identical to Version 1	ARH2CASS/B.pls
create/update cust acct site, create/update cust_ site_use site_use	HZ_CUST_ ACCOUNT_SITE_ V2PUB  create/update cust acct site, create/update cust_ site_use site_use	Account site, site uses	ARH2CSSS/B.pls
create/update cust_ account_role  create/update role_ responsibility	HZ_CUST_ ACCOUNT_ROLE_ V2PUB  create/update cust_ account_role  create/update role_ responsibility	Account role, role responsibility	ARH2CRSS/B.pls
	HZ_CUSTOMER_ PROFILE_V2PUB	Customer profile, profile amt	ARH2CFSS/B.pls
HZ_LOCATION_PUB  create/update location	HZ_LOCATION_ V2PUB  create/update location		ARH2LOSS/B.pls

<b>Version 1 API</b>	<b>Version 2 API</b>	<b>Comments</b>	<b>File Name</b>
HZ_PARTY_PUB create/update person create/update organization create/update group	HZ_PARTY_V2PUB	Person, group, organization	ARH2PASS/B.pls
create/update party site create/update party site use	HZ_PARTY_SITE_ V2PUB	Site, site uses	ARH2PSSS/B.pls
create update org_ contact create/update org contact roles	HZ_PARTY_ CONTACT_V2PUB	Contact, contact role	ARH2PCSS/B.pls
HZ_PER_INFO_PUB. Create/update person_language	HZ_PERSON_INFO_ V2PUB. Create/update person_language		ARH2PISS/B.pls
HZ_RELATIONSHIP_ PUB	HZ_RELATIONSHIP_ V2PUB		ARH2RESS/B.pls
HZ_RELATIONSHIP_ TYPE_PUB	HZ_RELATIONSHIP_ TYPE_V2PUB		ARH2RTSS/B.pls
HZ_TAX_ ASSIGNMENT_PUB	HZ_TAX_ ASSIGNMENT_ V2PUB		ARH2TASS/B.pls
	HZ_CONTACT_ PREFERENCE_ V2PUB		ARH2CTSS/B.pls
	HZ_REGISTRY_ VALIDATE_V2PUB	Public validation package for API in party level	ARH2RGVS/B.pls
	HZ_ACCOUNT_ VALIDATE_V2PUB	Public validation package for API in account level	ARH2ACVS/B.pls
	HZ_FORMAT_ PHONE_V2PUB	Phone Parsing and formatting	ARHPHFMS/B. pls

