Oracle® Trading Community Architecture

Technical Implementation Guide

Release 11*i*

Part No. B13890-03

July 2005

ORACLE

Oracle Trading Community Architecture Technical Implementation Guide, Release 11*i*

Part No. B13890-03

Contributing Author: Charlie Ahern, Rajib Borah, Uday Jose, Vinoo Nedungadan, Essan Ni, Sai Rangarajan, Viswanathan Ravichandran, Indrajit Sen, Colathur Vijayan

Contents

5 Person and Organization Information, Location, and Party Site API Use

6 Contact Point API Use

7 Relationship and Hierarchy API Use

8 Classification API Use

9 Contact Preference API Use

10 Customer Account API Use

11 Customer Account Site API Use

12 Customer Profile and Customer Account Role API Use

13 Source System Management API Use

14 Bulk Import API Use

15 Address and Tax Validation API Use

16 Data Quality Management API Use

17 Extensions API Use

18 Formatting API Use

19 Business Event System (BES) Callouts

A Sample TCA API Code

B List of TCA API Messages

C API Mapping

Send Us Your Comments

Oracle Trading Community Architecture Technical Implementation Guide, Release 11*i*

Part No. B13890-03

Oracle 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 title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

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

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

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

Preface

Intended Audience

Welcome to Release 11*i* of the *Oracle Trading Community Architecture Technical Implementation Guide*.

This guide contains the information needed to use the public Oracle Trading Community Architecture application programming interfaces (APIs) and develop callouts based on Oracle Workflow Business Events System (BES). For each API, this guide provides a description of the API, the PL/SQL procedure, and the Java method, as well as a table of the parameter descriptions and validations. For each BES callout, this guide provides the name of the logical entity, its description, and the ID parameter name. Also included are setup instructions and sample code.

See Related Documents on page xiv for more Oracle Applications product information.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ .

Accessibility of Code Examples in Documentation

Screen readers 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, some screen readers 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 does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

This chapter describes Customer Account Site APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

12 Customer Profile and Customer Account Role API Use

This chapter describes Customer Profile and Customer Account Role APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

13 Source System Management API Use

This chapter describes Source System Management APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

14 Bulk Import API Use

This chapter describes Bulk Import APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

15 Address and Tax Validation API Use

This chapter describes Location Service, Adapter, and Tax Assignment APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

16 Data Quality Management API Use

This chapter describes Data Quality Management Search and Duplicate Identification, Availability, and Transformation APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

17 Extensions API Use

This chapter describes Extensions APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

18 Formatting API Use

This chapter describes Phone Parsing and Formatting, Name and Address Formatting, and General Data Formatting APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

19 Business Event System (BES) Callouts

This chapter includes information about the use of Business Event System Callouts.

A Sample TCA API Code

This appendix provides sample code that uses the Oracle Trading Community Architecture APIs.

B List of TCA API Messages

This appendix provides the number, code, and text of the Oracle Trading Community Architecture API messages.

C API Mapping

This appendix provides information about mapping the first, internal-only version of the Oracle Trading Community Architecture APIs to the second, public version of the APIs.

Related Documents

Online Documentation

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

- **PDF Documentation:** See the Oracle Applications Documentation Library CD for current PDF documentation for your product with each release. The Oracle Applications Documentation Library is also available on Oracle*MetaLink* and is updated frequently.
- **Online Help:** Online help patches (HTML) are available on Oracle*MetaLink*.
- **About Documents:** Refer to the About Document for the mini-pack or family pack that you have installed to learn about new documentation or documentation patches that you can download. About Documents are available on Oracle*MetaLink*.

Guides Related to All Products

- **Oracle Applications User's Guide:** This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes. You can access this user's guide online by choosing *Getting Started with Oracle Applications* from any Oracle Applications help file.

Guides Related to This Product

- **Oracle Trading Community Architecture User Guide:** Use this guide to learn about Oracle Trading Community Architecture and how to use features from the Trading Community Manager responsibility to create, update, enrich, and cleanse the data in the TCA Registry.
- **Oracle Trading Community Architecture Administration Guide:** This guide describes how to administer and implement Oracle Trading Community Architecture. You set up, control, and manage functionality that affect data in the TCA Registry.
- **Oracle Trading Community Architecture Reference Guide:** This reference guide contains seeded relationship types, seeded Data Quality Management data, D&B data elements, Bulk Import interface table fields and validations, and a comprehensive glossary. This guide supplements the documentation for Oracle Trading Community Architecture and all products in the Oracle Customer Data Management family.
- **Oracle Customers Online User Guide:** This guide describes how to use Oracle Customers Online to view, create, and maintain your customer information. Oracle Customers Online is based on Oracle Trading Community Architecture data model and functionality, and is also part of the Oracle Customer Data Management product family.
- **Oracle Customers Online Implementation Guide:** This guide describes how to implement Oracle Customers Online.
- **Oracle Customer Data Librarian User Guide:** Use this guide to learn how to use Oracle Customer Data Librarian to establish and maintain the quality of the TCA Registry, focusing on consolidation, cleanliness, and completeness. Oracle Customer Data Librarian has all of the features in Oracle Customers Online, and is also part of the Oracle Customer Data Management product family.

- **Oracle Customer Data Librarian Implementation Guide:** This guide describes how to implement Oracle Customer Data Librarian. As part of implementing Customer Data Librarian, you must also complete all the implementation steps for Oracle Customers Online.
- **Oracle Receivables User Guide:** This user guide describes how to enter and maintain party and customer account information, define and use customer profiles, run reports for customer information, merge customer accounts or sites within an account, and import party and customer accounts.

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 installing 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: A Guide to Using Rapid Install:** This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications and the technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.
- **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:** Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle Applications file system and database.
- **Oracle Applications System Administrator's Documentation Set:** 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 Flexfields Guide:** This manual provides flexfields planning, setup, and reference information, as well as information on creating custom reports on flexfields data.

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 Administrator's Guide:** This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.
- **Oracle Workflow Developer's Guide:** This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.
- **Oracle Workflow User's Guide:** This guide describes how Oracle Applications users can view and respond to workflow notifications and monitor the progress of their workflow processes.
- **Oracle Workflow API Reference:** This guide describes the APIs provided for developers and administrators to access Oracle Workflow.
- **Oracle Applications 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 *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 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 and integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*MetaLink*.
- **Oracle Applications 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 tells you how to apply this UI to the design of an application built by using Oracle Forms.

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 an Oracle Applications form 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.

1

Overview

This chapter provides an overview of the features and benefits of the Oracle Trading Community Architecture application programming interfaces (APIs).

This chapter covers the following topics:

- Introduction
- Business Needs Met

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.

2

Features, Parameters, Attributes and Messages

This chapter describes the features, parameters, attributes and messages that are part of the Oracle Trading Community Architecture APIs.

This chapter covers the following topics:

- Major Features and Solution Outline
- Standard Parameters and Missing Attributes (PL/SQL API)
- Debug Messages (PL/SQL API)
- Standard Parameters and Missing Attributes (Java API)
- Debug Messages (Java API)
- Security of Oracle Human Resources Data
- Third Party Data Integration

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: Messages for the TCA APIs, page B-1.

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.

> **Caution:** 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/tca11
5/');
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 through, 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.prepa
reCall(
"begin fnd_profile.put('HZ_API_DEBUG_FILE_PATH', '/sqlcom/out/tca1
15/'); 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. Unless the privilege is provided through the user create and update rule for Other entities, users cannot update third-party locations, except for data populated in descriptive flexfields.

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.

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

Phone component information from data sources other than USER_ENTERED cannot be updated if the user does not have update privilege specified through rules. D&B data is always nonupdateable, as update rules cannot be set for PURCHASED systems.

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.

3

Party and Party Information API Use

This chapter describes Party and Party Information APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Party APIs
- Party Information APIs

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),
    orig_system                     VARCHAR2(30),
    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_natl_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_CONTEN
T_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_T
YPE,
    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,
    deceased_ind
    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_CONTEN
T_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_SOUR
CE_TYPE,
    party_rec                           PARTY_REC_TYPE:= G_MISS_PA
RTY_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),
    mission_statement               VARCHAR2(2000),
    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                       orig_system;
    public String                       status;
    public String                       category_code;
    public String                       salutation;
    public String                       attribute_category;
    public String                       attribute1;
    public String                       attribute2;
    public String                       atribute3;
    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 _RosettaUseGMISSValues);
}
```

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 BigDecima             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 String                    deceased_ind
    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 String                   mission_statement;
    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. If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key.

Classification code assignment records are also created in the HZ_CODE_ASS IGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.
- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

PL/SQL Procedure

```
PROCEDURE create_organization (
    p_init_msg_list                     IN            VARCHAR2:= FN
D_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 tables list information about the parameters in the Create Organization API. The tables include 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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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.
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 OWN_RENT_IND

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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.
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
jgzz_fiscal_code	IN	VARCHAR2	No	
analysis_fy	IN	VARCHAR2	No	
fiscal_yearend_month	IN	VARCHAR2	No	Validation: Validated against AR lookup type MONTH

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	
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 the TOTAL_EM PLOYEES_INDICATOR lookup type.
total_emp_est_ind	IN	VARCHAR2	No	Validation: Validated against the TOTAL_EMP_ EST_IND lookup type.
total_emp_min_ind	IN	VARCHAR2	No	Validation: Validated against the TOTAL_EMP_M IN_IND lookup type.
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	Validation: Validated against AR lookup type = value of local_activity_code_type.
emp_at_primary_adr	IN	VARCHAR2	No	
emp_at_primary_adr_text	IN	VARCHAR2	No	
emp_at_primary_adr_est_ind	IN	VARCHAR2	No	Validation: Validated against the EMP_AT_PR IMARY_ADR_EST_IND lookup type.
emp_at_primary_adr_min_ind	IN	VARCHAR2	No	Validation: Validated against the EMP_AT_PR IMARY_ADR_MIN_IND lookup type.
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.
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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.
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.
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: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. 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: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. Default - SST

party_rec Record 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_ reference	IN	VARCHAR2	No	Default: party_id Validation: If orig_system is passed in, then orig_sysatem_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_ REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

Classification code assignment records are also created in the HZ_CODE_ASS IGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.
- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

PL/SQL Procedure

```
PROCEDURE update_organization (
    p_init_msg_list               IN          VARCHAR2:= FND_API.G
_FALSE,
    p_organization_rec            IN          ORGANIZATION_REC_TYP
E,
    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 tables list information about the parameters in the Update Organization API. The tables include 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
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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: If this value is changed, then validated against AR lookup type OWN_RENT_IND.
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 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	Validation: Validated against AR lookup type MONTH
gsa_indicator_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
jgzz_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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	Validation: Validated against AR lookup type = value of local_activity_code_type.
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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.
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.
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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
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	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
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	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	
p_party_object_ version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing party record Comment: • Pass the current object_version_number of the record from hz_parties • Return new value after update
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. If orig_system is

passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key.

When Oracle Human Resources calls this API to replicate information in HR tables, if the HZ_PROTECT_HR_PERSON_INFO profile option is set to YES, then sensitive information, such as gender, marital_status, date_of_birth, and place_of_birth, is not propagated from HR to the HZ_PERSON_PROFILES table.

Classification code assignment records are also created in the HZ_CODE_ASS IGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.
- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

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                              p_init_msg_list,
    PersonRec                           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 tables list information about the parameters in the Create Person API. The tables include 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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
deceased_ind	IN	VARCHAR2	No	Validation: Validated against date_of_death. If deceased_ind is N, then date_of_death cannot have a value. Default: If date_of_death is null, then N. If date_of_death is not null, then Y.
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 OWN_RENT_IND.
last_known_gps	IN	VARCHAR2	No	
content_source_type	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. 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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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
actual_content_ source	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. Default - SST

Party_rec Record Type Attributes

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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 Validation: If orig_system is passed in, then orig_sysatem_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system.
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

Parameter Name	Type	Data Type	Required Validation, Default, Comments	
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	
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

Classification code assignment records are also created in the HZ_CODE_ASSIGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.
- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

When Oracle Human Resources (HR) calls this API to replicate information in HR tables, if the HZ_PROTECT_HR_PERSON_INFO profile option is set to YES, then sensitive information, such as gender, marital_status, date_of_birth, and place_of_birth, is not propagated from HR to the HZ_PERSON_PROFILES table.

If HR created the record, then only the attributes replicated from HR, not the entire person record, is protected against update by other users. If users attempt to change the value of an attribute replicated from HR, then they receive an error.

The attributes that you can replicate from HR are:

- person_first_name
- person_last_name
- person_middle_name
- person_name_suffix
- person_previous_last_name
- known_as
- person_title
- person_first_name_phonetic
- person_last_name_phonetic
- person_name_phonetic
- gender
- date_of_birth
- place_of_birth

- marital_status

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 tables list information about the parameters in the Update Person API. The tables include 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
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	
deceased_ind	IN	VARCHAR2	No	Default: If this parameter is null and date_of_death is not null, then this parameter changes to Y.
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: If the value is changed, then the value is validated against the AR lookup type OWN_RENT_IND.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
status	IN	VARCHAR2	No	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	
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, Comment
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 OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing party record. Comment: • Pass the current object_version _number of the record from HZ_PARTIES • Return new value after update
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. If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key.

Classification code assignment records are also created in the HZ_CODE_ASS IGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.

- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

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                              p_init_msg_list,
    GroupRec                            p_group_rec,
    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 tables list information about the parameters in the Create Group API. The tables include 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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_ reference	IN	VARCHAR2	No	Default: party_id Validation: If orig_system is passed in, then orig_sysatem_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
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	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

Classification code assignment records are also created in the HZ_CODE_ASSIGNMENTS table based on certain conditions.

- If CATEGORY_CODE is specified, then a code assignment is created with CUSTOMER_CATEGORY as the class category and CATEGORY_CODE as the class code.
- If LOCAL_ACTIVITY_CODE_TYPE is specified, then a code assignment is created with LOCAL_ACTIVITY_CODE_TYPE class category and LOCAL_ACTIVITY_CODE class code.
- If SIC_CODE_TYPE is specified, then a code assignment is created with SIC_CODE_TYPE class category and SIC_CODE class code.

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 tables list information about the parameters in the Update Group API. The tables include 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
application_id	IN	NUMBER	No	Validation: Non Updateable if value exists

party_rec Record Type Attributes

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
status	IN	VARCHAR2	No	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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute24	IN	VARCHAR2	No	
p_party_object_ version_number	IN/OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing party record Comment: • Pass the current object_version_number of the record from HZ_PARTIES • Return new value after update.

Party Information APIs

PL/SQL Package Name: HZ_PARTY_INFO_V2PUB

PL/SQL Record Structure for Party Info

```
TYPE credit_rating_rec_type IS RECORD(
credit_rating_id                NUMBER,
description                     VARCHAR2(2000),
party_id                        NUMBER,
rating                          VARCHAR2(60),
rated_as_of_date                DATE,
rating_organization             VARCHAR2(240),
comments                        VARCHAR2(240),
det_history_ind                 VARCHAR2(5),
fincl_embt_ind                  VARCHAR2(5),
criminal_proceeding_ind         VARCHAR2(5),
claims_ind                      VARCHAR2(5),
secured_flng_ind                VARCHAR2(5),
fincl_lgl_event_ind             VARCHAR2(5),
disaster_ind                    VARCHAR2(5),
oprg_spec_evnt_ind              VARCHAR2(5),
other_spec_evnt_ind             VARCHAR2(5),
status                          VARCHAR2(1),
avg_high_credit                 NUMBER,
credit_score                    VARCHAR2(30),
credit_score_age                NUMBER,
credit_score_class              NUMBER,
credit_score_commentary         VARCHAR2(30),
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),
credit_score_date               DATE,
credit_score_incd_default       NUMBER,
credit_score_natl_percentile    NUMBER,
```

```
failure_score                   VARCHAR2(30),
failure_score_age               NUMBER,
failure_score_class             NUMBER,
failure_score_commentary        VARCHAR2(30),
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),
failure_score_date              DATE,
failure_score_incd_default      NUMBER,
failure_score_natnl_percentile  NUMBER,
failure_score_override_code     VARCHAR2(30),
global_failure_score            VARCHAR2(30),
debarment_ind                   VARCHAR2(30),
debarments_count                NUMBER,
debarments_date                 DATE,
high_credit                     NUMBER,
maximum_credit_currency_code    VARCHAR2(240),
maximum_credit_rcmd             NUMBER,
paydex_norm                     VARCHAR2(3),
paydex_score                    VARCHAR2(3),
paydex_three_months_ago         VARCHAR2(3),
credit_score_override_code      VARCHAR2(30),
cr_scr_clas_expl                VARCHAR2(30),
low_rng_delq_scr                NUMBER,
high_rng_delq_scr               NUMBER,
delq_pmt_rng_prcnt              NUMBER,
delq_pmt_pctg_for_all_firms     NUMBER,
num_trade_experiences           NUMBER,
paydex_firm_days                VARCHAR2(15),
paydex_firm_comment             VARCHAR2(60),
paydex_industry_days            VARCHAR2(15),
paydex_industry_comment         VARCHAR2(50),
paydex_comment                  VARCHAR2(240),
suit_ind                        VARCHAR2(5),
lien_ind                        VARCHAR2(5),
judgement_ind                   VARCHAR2(5),
bankruptcy_ind                  VARCHAR2(5),
no_trade_ind                    VARCHAR2(5),
prnt_hq_bkcy_ind                VARCHAR2(5),
num_prnt_bkcy_filing            NUMBER,
prnt_bkcy_filg_type             VARCHAR2(20),
prnt_bkcy_filg_chapter          NUMBER,
prnt_bkcy_filg_date             DATE,
num_prnt_bkcy_convs             NUMBER,
prnt_bkcy_conv_date             DATE,
prnt_bkcy_chapter_conv          VARCHAR2(60),
slow_trade_expl                 VARCHAR2(100),
negv_pmt_expl                   VARCHAR2(150),
pub_rec_expl                    VARCHAR2(150),
business_discontinued           VARCHAR2(240),
spcl_event_comment              VARCHAR2(150),
num_spcl_event                  NUMBER,
```

```
spcl_event_update_date          DATE,
spcl_evnt_txt                   VARCHAR2(2000),
actual_content_source           VARCHAR2(30),
created_by_module               VARCHAR2(150)
);
```

Create Credit Ratings

PL/SQL Procedure

```
PROCEDURE create_credit_rating(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_credit_rating_rec        IN  CREDIT_RATING_REC_TYPE,
    x_credit_rating_id         OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Credit Ratings 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
credit_rating_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence hz_credit_ratings_s. Not updateable.
description	IN	VARCHAR2	No	
party_id	IN	NUMBER	Yes	Validation: Must a valid party from hz_parties. Non-updateable
rating	IN	VARCHAR2	No	
rated_as_of_date	IN	DATE	No	
rating_organization	IN	VARCHAR2	No	
comments	IN	VARCHAR2	No	
det_history_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_embt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
criminal_proceeding_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
claims_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
secured_flng_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
fincl_lgl_event_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
disaster_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
oprg_spec_evnt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
other_spec_evnt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
status	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS
avg_high_credit	IN	NUMBER	No	
credit_score	IN	VARCHAR2	No	
credit_score_age	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	
credit_score_ commentary	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary2	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary3	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary4	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary5	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary6	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary7	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary8	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary9	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary10	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_date	IN	DATE	No	
credit_score_incd_ default	IN	NUMBER	No	
credit_score_natl_ percentile	IN	NUMBER	No	
failure_score	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
failure_score_age	IN	NUMBER	No	
failure_score_class	IN	NUMBER	No	
failure_score_ commentary	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary2	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary3	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary4	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary5	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary6	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary7	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary8	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary9	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary10	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_date	IN	DATE	No	
failure_score_incd_ default	IN	NUMBER	No	
failure_score_natnl_ percentile	IN	NUMBER	No	
failure_score_ override_code	IN	VARCHAR2	No	
global_failure_score	IN	VARCHAR2	No	
debarment_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
debarments_count	IN	NUMBER	No	
debarments_date	IN	DATE	No	
high_credit	IN	NUMBER	No	
maximum_credit_ currency_code	IN	VARCHAR2	No	Validation: Foreign key to fnd_currencies.currency_ code
maximum_credit_ rcmd	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
paydex_norm	IN	VARCHAR2	No	
paydex_score	IN	VARCHAR2	No	
paydex_three_ months_ago	IN	VARCHAR2	No	
credit_score_ override_code	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE
cr_scr_clas_expl	IN	VARCHAR2	No	
low_rng_delq_scr	IN	NUMBER	No	
high_rng_delq_scr	IN	NUMBER	No	
delq_pmt_rng_prcnt	IN	NUMBER	No	
delq_pmt_pctg_for_ all_firms	IN	NUMBER	No	
num_trade_ experiences	IN	NUMBER	No	
paydex_firm_days	IN	VARCHAR2	No	
paydex_firm_ comment	IN	VARCHAR2	No	
paydex_industry_ days	IN	VARCHAR2	No	
paydex_industry_ comment	IN	VARCHAR2	No	
paydex_comment	IN	VARCHAR2	No	
suit_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
lien_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
judgement_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
bankruptcy_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
no_trade_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
prnt_hq_bkcy_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
num_prnt_bkcy_ filing	IN	NUMBER	No	
prnt_bkcy_filg_type	IN	VARCHAR2	No	
prnt_bkcy_filg_ chapter	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
prnt_bkcy_filg_date	IN	DATE	No	
num_prnt_bkcy_convs	IN	NUMBER	No	
prnt_bkcy_conv_date	IN	DATE	No	
prnt_bkcy_chapter_conv	IN	VARCHAR2	No	
slow_trade_expl	IN	VARCHAR2	No	
negv_pmt_expl	IN	VARCHAR2	No	
pub_rec_expl	IN	VARCHAR2	No	
business_discontinued	IN	VARCHAR2	No	
spcl_event_comment	IN	VARCHAR2	No	
num_spcl_event	IN	NUMBER	No	
spcl_event_update_date	IN	DATE	No	
spcl_evnt_txt	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	No	Validate: HZ_MIXNM_UTILITY ValidateContentSource
created_by_module	IN	VARCHAR2	Yes	

Other Validations

The following column combination must be unique: party_id, truncated_as_of_date, rating_organization, and actual_content_source.

Update Credit Ratings

PL/SQL Procedure

```
PROCEDURE update_credit_rating(
    p_init_msg_list             IN     VARCHAR2 := FND_API.G_FALSE
,
    p_credit_rating_rec         IN     CREDIT_RATING_REC_TYPE,
    p_object_version_number     IN  OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Credit Ratings 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
credit_rating_id	IN	NUMBER	Yes	Validation: Valid credit_rating_id from the HZ_CREDIT_RATINGS table.
description	IN	VARCHAR2	No	
party_id	IN	NUMBER	No	Validation: Must a valid party from hz_parties. Non-updateable
rating	IN	VARCHAR2	No	
rated_as_of_date	IN	DATE	No	
rating_organization	IN	VARCHAR2	No	
comments	IN	VARCHAR2	No	
det_history_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_embt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
criminal_ proceeding_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
claims_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
secured_flng_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_lgl_event_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
disaster_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
oprg_spec_evnt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
other_spec_evnt_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
status	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS
avg_high_credit	IN	NUMBER	No	
credit_score	IN	VARCHAR2	No	
credit_score_age	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	
credit_score_ commentary	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
credit_score_ commentary2	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary3	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary4	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary5	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary6	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary7	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary8	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary9	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_ commentary10	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_date	IN	DATE	No	
credit_score_incd_ default	IN	NUMBER	No	
credit_score_natl_ percentile	IN	NUMBER	No	
failure_score	IN	VARCHAR2	No	
failure_score_age	IN	NUMBER	No	
failure_score_class	IN	NUMBER	No	
failure_score_ commentary	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary2	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary3	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary4	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary5	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary6	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary7	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
failure_score_ commentary8	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary9	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_ commentary10	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_date	IN	DATE	No	
failure_score_incd_ default	IN	NUMBER	No	
failure_score_natnl_ percentile	IN	NUMBER	No	
failure_score_ override_code	IN	VARCHAR2	No	
global_failure_score	IN	VARCHAR2	No	
debarment_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
debarments_count	IN	NUMBER	No	
debarments_date	IN	DATE	No	
high_credit	IN	NUMBER	No	
maximum_credit_ currency_code	IN	VARCHAR2	No	Validation: Foreign key to fnd_currencies.currency_ code
maximum_credit_ rcmd	IN	NUMBER	No	
paydex_norm	IN	VARCHAR2	No	
paydex_score	IN	VARCHAR2	No	
paydex_three_ months_ago	IN	VARCHAR2	No	
credit_score_ override_code	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE
cr_scr_clas_expl	IN	VARCHAR2	No	
low_rng_delq_scr	IN	NUMBER	No	
high_rng_delq_scr	IN	NUMBER	No	
delq_pmt_rng_prcnt	IN	NUMBER	No	
delq_pmt_pctg_for_ all_firms	IN	NUMBER	No	
num_trade_ experiences	IN	NUMBER	No	
paydex_firm_days	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
paydex_firm_ comment	IN	VARCHAR2	No	
paydex_industry_ days	IN	VARCHAR2	No	
paydex_industry_ comment	IN	VARCHAR2	No	
paydex_comment	IN	VARCHAR2	No	
suit_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
lien_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
judgement_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
bankruptcy_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
no_trade_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
prnt_hq_bkcy_ind	IN	VARCHAR2	No	Validation: Must be a valid lookup value from lookup YES/NO
num_prnt_bkcy_ filing	IN	NUMBER	No	
prnt_bkcy_filg_type	IN	VARCHAR2	No	
prnt_bkcy_filg_ chapter	IN	NUMBER	No	
prnt_bkcy_filg_date	IN	DATE	No	
num_prnt_bkcy_ convs	IN	NUMBER	No	
prnt_bkcy_conv_ date	IN	DATE	No	
prnt_bkcy_chapter_ conv	IN	VARCHAR2	No	
slow_trade_expl	IN	VARCHAR2	No	
negv_pmt_expl	IN	VARCHAR2	No	
pub_rec_expl	IN	VARCHAR2	No	
business_ discontinued	IN	VARCHAR2	No	
spcl_event_ comment	IN	VARCHAR2	No	
num_spcl_event	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
spcl_event_update_date	IN	DATE	No	
spcl_evnt_txt	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	No	Validate: HZ_MIXNM_UTILITY ValidateContentSource
created_by_module	IN	VARCHAR2	No	Not updateable if value exists.

Other Validations

The following column combination must be unique: party_id, truncated_as_of_date, rating_organization, and actual_content_source.

4

Party Contact API Use

This chapter describes Party Contact APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Party Contact APIs

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),
    orig_system                         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),
    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.rela
tionship_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),
    orig_system                     VARCHAR2(30),
    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                   orig_system;
    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                       orig_system;
    public String                       role_level;
    public String                       primary_contact_per_role_t
ype;
    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.

If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_ REFERENCES table to store the mapping between the source system reference and the TCA primary key.

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 tables list information about the parameters in the Create Org Contact API. The tables include 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
orig_system_ reference	IN	VARCHAR2	No	Default: org_contact_id Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system.
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	

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.

party_rel_rec Record Type Attributes

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
relationship_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.
subject_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign key to HZ_PARTIES.PARTY_ID Comment: Pass the party_id of the contact person here.
subject_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to FND_OBJECT_INSTANCE_SETS.INSTANCESET_NAME Comment: Pass the party_type of the subject person, which is PERSON here.
subject_table_name	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to FND_OBJECTS.OBJ.NAME Comment: Pass HZ_PARTIES for the table name of the source of the subject.
object_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign key to HZ_PARTIES.PARTY_ID Comment: Pass the party_id of the organization or person for which you are creating the contact.
object_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to FND_OBJECT_INSTANCE_SETS.INSTANCE_SET_NAME Comment: Pass ORGANIZATION or PERSON depending on whether you are creating contact for an organization or for a person.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
object_table_name	IN	VARCHAR2	No	Validation: • Mandatory attribute • Foreign key to FND_OBJECTS.OBJ_NAME Comment: Pass HZ_PARTIES as the table name that is the source of the object.
relationship_code	IN	VARCHAR2	No	Validation: • Mandatory attribute • Validated against AR lookup type PARTY_ RELATIONS_TYPE • Required to be a valid forward_rel_code for the particular relationship type requested.
relationship_type	IN	VARCHAR2	No	Validation: • Mandatory attribute • Must be a valid relationship_type from the HZ_RELATIONSHIP_TYPE table for the combination of subject_type, object_type, and relationship_code passed.
comments	IN	VARCHAR2	No	
start_date	IN	DATE	No	Validation: Mandatory attribute
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: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of *Y*. 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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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.
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
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	

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	
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

party_rec Record Type Attributes

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	Yes	Validation: Unique if passed in, otherwise generated from sequence

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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
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	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

PL/SQL Procedure

```
PROCEDURE update_org_contact (
    p_init_msg_list                IN       VARCHAR2:= FND_API.G_F
ALSE,
    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 tables list information about the parameters in the Update Org Contact API. The tables include 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	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_ system.
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	
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: • Pass the relationship_id of the relationship record for this org contact. • Pass if you want to update the relationship record.
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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	No	Validation: Not updateable if value exists
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

party_rec Record Type Attributes

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	Yes/No	Validation: Valid party id in HZ_PARTIES Comment: • Pass party_id of the relationship's party record • Pass if you want to update the party record
party_number	IN	VARCHAR2	No	Validation: Not updateable
validated_flag	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: • Validate 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	
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	
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: • Mandatory attribute • Validated against the value in the database for the existing org contact record Comment: • Pass the current object_version_number of the record from HZ_ORG_CONTACTS • Return a new value after update

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_rel_object_version_number	IN OUT	NUMBER	Yes/No	Validation: • Mandatory attribute if relationship to be updated • Validated against the value in the database for the existing relationship record Comment: • Pass the current object_version_number of the record from HZ_RELATIONSHIPS • Return a new value after update
p_party_object_version_number	IN OUT	NUMBER	Yes/No	Validation: • Mandatory attribute if party to be updated • Validated against the value in the database for the existing party record Comment: • Pass the current object_version_number of the record from HZ_PARTIES • Return a new value after update

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.

If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key.

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_RO
LE_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: • Mandatory attribute • Validated against AR lookup type CONTACT_ROLE_TYPE
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
org_contact_id	IN	NUMBER	Yes	Validation: Mandatory attribute Foreign key to HZ_ORG_CONTACTS.ORG_CONTACT_ID
orig_system_reference	IN	VARCHAR2	No	Default: org_contact_role_id Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system.
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
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

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_RO
LE_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                              p_init_msg_list,
    OrgContactRoleRec                   p_org_contact_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 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: • Cannot be set to null during update • Validated against AR lookup type CONTACT_ROLE_TYPE
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: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_ SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_ SYSTEMS.orig_system.
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 • Cannot be set to null during update
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: • Mandatory attribute • Validated against value in the database for the existing org contact role record Comment: • Pass the current object_version_number of the record from hz_org_contact_roles. • Return new value after update.

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.

5

Person and Organization Information, Location, and Party Site API Use

This chapter describes Person and Organization Information, Location, and Party Site APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Person Information APIs
- Organization Information APIs
- Location APIs
- Party Site APIs

Person Information 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(30),
    speaks_level                         VARCHAR2(30),
    writes_level                         VARCHAR2(30),
    spoken_comprehension_level           VARCHAR2(30),
    status                               VARCHAR2(1),
    created_by_module                    VARCHAR2(150),
    application_id                       NUMBER
)
```

PL/SQL Record Structure for Citizenship

```
TYPE citizenship_rec_type IS RECORD(
    citizenship_id                  NUMBER,
    party_id                        NUMBER,
    birth_or_selected               VARCHAR2(30),
    country_code                    VARCHAR2(2),
    date_recognized                 DATE,
    date_disowned                   DATE,
    end_date                        DATE,
    document_type                   VARCHAR2(30),
    document_reference              VARCHAR2(60),
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150)
);
```

PL/SQL Record Structure for Education

```
TYPE education_rec_type IS RECORD(
    education_id                    NUMBER,
    party_id                        NUMBER,
    course_major                    VARCHAR2(60),
    degree_received                 VARCHAR2(60),
    start_date_attended             DATE,
    last_date_attended              DATE,
    school_attended_name            VARCHAR2(60),
    school_party_id                 NUMBER
    school_party_id                 NUMBER,
    type_of_school                  VARCHAR2(60),
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150)
);
```

PL/SQL Record Structure for Employment History

```
TYPE employment_history_rec_type IS RECORD(
    employment_history_id           NUMBER,
    party_id                        NUMBER,
    begin_date                      DATE,
    end_date                        DATE,
    employment_type_code            VARCHAR2(30),
    employed_as_title_code          VARCHAR2(30),
    employed_as_title               VARCHAR2(60),
    employed_by_name_company        VARCHAR2(60),
    employed_by_party_id            NUMBER,
    employed_by_division_name       VARCHAR2(60),
    supervisor_name                 VARCHAR2(60),
    branch                          VARCHAR2(80),
    military_rank                   VARCHAR2(240),
    served                          VARCHAR2(240),
    station                         VARCHAR2(240),
    responsibility                  VARCHAR2(240),
    weekly_work_hours               NUMBER,
    reason_for_leaving              VARCHAR2(240),
    faculty_position_ind            VARCHAR2(30),
    tenure_code                     VARCHAR2(30),
    fraction_of_tenure              NUMBER,
    comments                        VARCHAR2(2000),
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150),
);
```

PL/SQL Record Structure for Work Classes

```
TYPE work_class_rec_type IS RECORD(
    work_class_id                   NUMBER,
    level_of_experience             VARCHAR2(60),
    work_class_name                 VARCHAR2(240),
    employment_history_id           NUMBER,
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150)
);
```

PL/SQL Record Structure for Person Interest

```
TYPE person_interest_rec_type IS RECORD(
    person_interest_id              NUMBER,
    level_of_interest               VARCHAR2(30),
    party_id                        NUMBER,
    level_of_participation          VARCHAR2(30),
    interest_type_code              VARCHAR2(30),
    comments                        VARCHAR2(240),
    sport_indicator                 VARCHAR2(1),
    sub_interest_type_code          VARCHAR2(30),
    interest_name                   VARCHAR2(240),
    team                            VARCHAR2(240),
    since                           DATE,
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150)
);
```

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;
    publicPersonLanguageRec();
 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
language_name	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to fnd_languages.language_code
party_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign key to hz_parties.party_id
native_language	IN	VARCHAR2	No	
primary_language_ indicator	IN	VARCHAR2	No	
reads_level	IN	VARCHAR2	No	
speaks_level	IN	VARCHAR2	No	
writes_level	IN	VARCHAR2	No	
spoken_ comprehension_level				Validation: Validated against the HZ_LANGUAGE_ PROFICIENCY lookup type.
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, which should be an active 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
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: • Validated against AR lookup type REGISTRY_STATUS • Cannot be set to null during update
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: • Mandatory attribute • Validated against value in the database for the existing person language record Comment: • Pass the current object_version_number of the record from HZ_PERSON_LANGUAGE. • Return new value after update.

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, which should be an active language.

Create Citizenship API

PL/SQL Procedure

```
PROCEDURE create_citizenship(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_citizenship_rec          IN  CITIZENSHIP_REC_TYPE,
    x_citizenship_id           OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Citizenship 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
citizenship_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence. Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES. Party_id must represent a Person party.
birth_or_selected_ind	IN	VARCHAR2	No	Validation: Validated against the HZ_CITIZENSHIP_ACQUISITION lookup type.
country_code	IN	VARCHAR2	Yes	Validation: Must exist in FND_TERRITORIES
date_recognized	IN	DATE	No	
date_disowned	IN	DATE	No	
end_date	IN	DATE	No	
document_type	IN	VARCHAR2	No	
document_reference	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	Validation: Not updateable if value exists

Update Citizenship API

PL/SQL Procedure

```
PROCEDURE update_citizenship(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_citizenship_rec        IN     CITIZENSHIP_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Citizenship 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
citizenship_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence.Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES. Party_id must represent a Person party.
p_object_version_number	IN OUT	NUMBER	Yes	
birth_or_selected_ind	IN	VARCHAR2	No	Validation: validated against AR lookup type HZ_CITIZENSHIP_ACQUISITION
country_code	IN	VARCHAR2	Yes	Validation: Must exist in FND_TERRITORIES
date_recognized	IN	DATE	No	
date_disowned	IN	DATE	No	
end_date	IN	DATE	No	
document_type	IN	VARCHAR2	No	
document_reference	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Create Education API

PL/SQL Procedure

```
PROCEDURE create_education(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_education_rec            IN  EDUCATION_REC_TYPE,
    x_education_id             OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Education 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
education_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES. Party_id must represent a Person party.
course_major	IN	VARCHAR2	No	
degree_received	IN	VARCHAR2	No	
start_date_attended	IN	DATE	No	
last_date_attended	IN	DATE	No	Validation: If both start_date_attended and last_ date_attended are passed, then last_date_attended must be greater than or equal to start_date_attended.
school_attended_ name	IN	VARCHAR2	No	Validation: If school_party_id is passed, then school_attended_name should not be passed. Comments: This field captures the school name in situations where there is no Party that represents the school. If the school_party_id is known, then that party name will be denormalized in the school_attended_name field.
school_party_id	IN	VARCHAR2	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	IN	VARCHAR2	No	Validation: If the value is modified, then the value is validated against the HZ_TYPE_OF_SCHOOL lookup type.
Status	IN	VARCHAR2	No	Validation: Validated against the REGISTRY_ STATUS lookup type.
created_by_module	IN	VARCHAR2	Yes	

Update Education API

PL/SQL Procedure

```
PROCEDURE update_education(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_education_rec          IN     EDUCATION_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Education 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
education_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence. Not updateable.
party_id	IN	NUMBER	No	Validation: If passed it must exist in HZ_PARTIES and represent a Person party.
course_major	IN	VARCHAR2	No	
degree_received	IN	VARCHAR2	No	
start_date_attended	IN	DATE	No	
last_date_attended	IN	DATE	No	Validation: If both start_date_attended and last_ date_attended are passed, then last_date_attended must be greater than or equal to start_date_attended.
school_attended_ name	IN	VARCHAR2	No	Validation: If school_party_id is passed, then school_attended_name should not be passed. Comments: This field captures the school name in situations where there is no Party that represents the school. If the school_party_id is known, then that party name will be denormalized in the school_attended_name field.
school_party_id	IN	VARCHAR2	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Create Employment History API

PL/SQL Procedure

```
PROCEDURE create_employment_history(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_employment_history_rec   IN  EMPLOYMENT_HISTORY_REC_TYPE,
    x_employment_history_id    OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Employment History 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
employment_ history_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES. Party_id must represent a Person party.
begin_date	IN	DATE	No	
end_date	IN	DATE	No	Validation: If both begin_date and end_date are passed, then end_date must be greater than or equal to begin_date
employment_type_ code	IN	VARCHAR2	No	Validation: Validated against the HZ_EM PLOYMENT_TYPE lookup type
employed_as_title_ code	IN	VARCHAR2	No	Validation: Validated against the RESPONSIBILITY lookup type .
employed_as_title	IN	VARCHAR2	No	Validation: if employed_as_title_code is supplied, then employed_as_title must be null.
employed_by_ name_company	IN	VARCHAR2	No	Validation: If employed_by_party_id is passed, then employed_by_name_company should not be passed. Comments: This field captures the employer name in situations where there is no Party that represents the employer. If the employed_by_party_id is known, then that party name will be denormalized in the employed_by_name_company field.
employed_by_ party_id	IN	NUMBER	No	Validation: Must exist in the HZ_PARTIES table.
employed_by_ division_name	IN	VARCHAR2	No	
supervisor_name	IN	VARCHAR2	No	
Branch	IN	VARCHAR2	No	
military_rank	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
Served	IN	VARCHAR2	No	
Station	IN	VARCHAR2	No	
weekly_work_hours	IN	NUMBER	No	Validation: If passed, then must be greater than zero and less than or equal to 168.
reason_for_leaving	IN	VARCHAR2	No	
faculty_position_ind	IN	VARCHAR2	Yes	Default: N Validation: validated against AR lookup type YES/NO.
tenure_code	IN	VARCHAR2	No	Validation: Must only be passed if FACULTY_POS ITION_IND = 'Y', must be null otherwise. Validated against AR lookup type HZ_TENURE_CODE
fraction_of_tenure	IN	NUMBER	No	Validation: Must only be passed if FACULTY_ POSITION_IND = 'Y', must be null otherwise. If passed, must be between 0 and 100 inclusive.
comments	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Update Employment History API

PL/SQL Procedure

```
PROCEDURE update_employment_history(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_employment_history_rec IN     EMPLOYMENT_HISTORY_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Employment History 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
employment_ history_id	IN	NUMBER	Yes	Validation: From hz_employment_history. Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES. Party_id must represent a Person party.
begin_date	IN	DATE	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
end_date	IN	DATE	No	Validation: If both begin_date and end_date are passed, then end_date must be greater than or equal to begin_date
employment_type_code	IN	VARCHAR2	No	Validation: validated against AR lookup type HZ_EMPLOYMENT_TYPE
employed_as_title_code	IN	VARCHAR2	No	Validation: validated against AR lookup type RESPONSIBILITY.
employed_as_title	IN	VARCHAR2	No	Validation: if employed_as_title_code is supplied, then employed_as_title must be null.
employed_by_name_company	IN	VARCHAR2	No	Validation: If employed_by_party_id is passed, then employed_by_name_company should not be passed. Comments: This field captures the employer name in situations where there is no Party that represents the employer. If the employed_by_party_id is known, then that party name will be denormalized in the employed_by_name_company field.
employed_by_party_id	IN	NUMBER	No	Validation: Must exist in HZ_PARTIES.
employed_by_division_name	IN	VARCHAR2	No	
supervisor_name	IN	VARCHAR2	No	
Branch	IN	VARCHAR2	No	
military_rank	IN	VARCHAR2	No	
Served	IN	VARCHAR2	No	
Station	IN	VARCHAR2	No	
weekly_work_hours	IN	NUMBER	No	Validation: If passed, then must be greater than zero and less than or equal to 168.
reason_for_leaving	IN	VARCHAR2	No	
faculty_position_ind	IN	VARCHAR2	Yes	Default: N\ Validation: validated against AR lookup type YES/NO.
tenure_code	IN	VARCHAR2	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, must be null otherwise. Validated against AR lookup type HZ_TENURE_CODE
fraction_of_tenure	IN	NUMBER	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, must be null otherwise. If passed, must be between 0 and 100 inclusive.
Comments	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
Status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Create Work Classes API

PL/SQL Procedure

```
PROCEDURE create_work_class(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_work_class_rec           IN  WORK_CLASS_REC_TYPE,
    x_work_class_id            OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Work Classes 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
work_class_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence. Not updateable.
level_of_experience	IN	VARCHAR2	No	
work_class_name	IN	VARCHAR2	Yes	
employment_history_id	IN	NUMBER	Yes	Validation: Must exist in the HZ_EMPLOYMENT_HISTORY table. Not updateable.
reason_for_leaving	IN	VARCHAR2	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Update Work Classes API

PL/SQL Procedure

```
PROCEDURE update_work_class(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_work_class_rec         IN     WORK_CLASS_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Work Classes 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
work_class_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence.Not updateable.
level_of_experience	IN	VARCHAR2	No	
work_class_name	IN	VARCHAR2	Yes	
employment_ history_id	IN	NUMBER	Yes	Validation: Must exist in HZ_EMPLOYMENT_H ISTORY.Not updateable.
reason_for_leaving	IN	VARCHAR2	No	
Status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	

Create Person Interest API

PL/SQL Procedure

```
PROCEDURE create_person_interest(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_person_interest_rec      IN  PERSON_INTEREST_REC_TYPE,
    x_person_interest_id       OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Update Person Interest API

PL/SQL Procedure

```
PROCEDURE update_person_interest(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_person_interest_rec    IN     PERSON_INTEREST_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Organization Information APIs

PL/SQL Package Name: HZ_ORGANIZATION_INFO_V2PUB

PL/SQL Record Structure for Financial Report

```
TYPE financial_report_rec_type IS RECORD(
    financial_report_id             NUMBER,
    party_id                        NUMBER,
    type_of_financial_report        VARCHAR2(60),
    document_reference              VARCHAR2(150),
    date_report_issued              DATE,
    issued_period                   VARCHAR2(60),
    report_start_date               DATE,
    report_end_date                 DATE,
    actual_content_source           VARCHAR2(30),
    requiring_authority             VARCHAR2(60),
    audit_ind                       VARCHAR2(30),
    consolidated_ind                VARCHAR2(30),
    estimated_ind                   VARCHAR2(30),
    fiscal_ind                      VARCHAR2(30),
    forecast_ind                    VARCHAR2(30),
    opening_ind                     VARCHAR2(30),
    proforma_ind                    VARCHAR2(30),
    qualified_ind                   VARCHAR2(30),
    restated_ind                    VARCHAR2(30),
    signed_by_principals_ind        VARCHAR2(30),
    trial_balance_ind               VARCHAR2(30),
    unbalanced_ind                  VARCHAR2(30),
    status                          VARCHAR2(30),
    created_by_module               VARCHAR2(150)

);
```

PL/SQL Record Structure for Financial Number

```
TYPE financial_number_rec_type IS RECORD(
    financial_number_id             NUMBER,
    financial_report_id             NUMBER,
    financial_number                NUMBER,
    financial_number_name           VARCHAR2(60),
    financial_units_applied         NUMBER,
    financial_number_currency       VARCHAR2(240),
    projected_actual_flag           VARCHAR2(1),
    status                          VARCHAR2(1),
    created_by_module               VARCHAR2(150)
);
```

Create Financial Report API

PL/SQL Procedure

```
PROCEDURE create_financial_report(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_financial_report_rec     IN  FINANCIAL_REPORT_REC_TYPE,
    x_financial_report_id      OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Financial Report 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
financial_report_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence. Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in the HZ_PARTIES table. Party_id must represent an Organization party.
type_of_financial_report	IN	VARCHAR2	No	
document_reference	IN	VARCHAR2	No	
issued_period	IN	VARCHAR2	No	Validation: Either issued_period or report_start_date must be provided, but not both.
report_start_date	IN	DATE	No	Validation: Either issued_period or report_start_date must be provided, but not both. If report_start_date is provided, then it must be less than or equal to report_end_date.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
report_end_date	IN	DATE	No	Validation: Must be provided if report_start_date is provided, otherwise must be null. If provided, then it must be greater than or equal to report_start_date.
requiring_authority	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of *Y*.
audit_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
estimated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
fiscal_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
forecast_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
opening_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
proforma_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
qualified_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
restated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
signed_by_principals_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
trial_balance_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
unbalanced_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	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	

Other Validation

The following combination of columns must be unique: party_id, financial_report_type, document_reference, date_report_issued, (issued_period or report_start_date and report_end_date) and actual_content_source. Note that issued_period and report_start_date + report_end_date are mutually exclusive.

Update Financial Report API

PL/SQL Procedure

```
PROCEDURE update_financial_report(
    p_init_msg_list          IN     VARCHAR2 := FND_API.G_FALSE,
    p_financial_report_rec   IN     FINANCIAL_REPORT_REC_TYPE,
    p_object_version_number  IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Financial Report 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
financial_report_id	IN	NUMBER	Yes	Validation: Not updateable.
party_id	IN	NUMBER	Yes	Validation: Must exist in the HZ_PARTIES table. Party_id must represent an Organization party.
type_of_financial_report	IN	VARCHAR2	No	
document_reference	IN	VARCHAR2	No	
issued_period	IN	VARCHAR2	No	Validation: Either issued_period or report_start_date must be provided (not both).
report_start_date	IN	DATE	No	Validation: Either issued_period or report_start_date must be provided, but not both. If report_start_date is provided, then it must be less than or equal to report_end_date.
report_end_date	IN	DATE	No	Validation: Must be provided if report_start_date is provided, otherwise must be null. If provided, then it must be greater than or equal to report_start_date.
requiring_authority	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*.
audit_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
estimated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
fiscal_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
forecast_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
opening_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
proforma_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
qualified_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
restated_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
signed_by_ principals_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
trial_balance_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
unbalanced_ind	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	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	No	

Other Validation

The following combination of columns must be unique: party_id, financial_report_ type, document_reference, date_report_issued, (issued_period or report_start_date and report_end_date) and actual_content_source. Note that issued_period and report_start_date + report_end_date are mutually exclusive.

Create Financial Number API

```
PROCEDURE create_financial_number(
    p_init_msg_list            IN  VARCHAR2 := FND_API.G_FALSE,
    p_financial_number_rec     IN  FINANCIAL_NUMBER_REC_TYPE,
    x_financial_number_id      OUT NOCOPY NUMBER,
    x_return_status            OUT NOCOPY VARCHAR2,
    x_msg_count                OUT NOCOPY NUMBER,
    x_msg_data                 OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Financial Number 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
financial_number_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.Not updateable.
financial_report_id	IN	NUMBER	Yes	Validation: Must exist in HZ_FINANCIAL_REPORTS. Non-updateable.
financial_number	IN	VARCHAR2	No	
financial_number_name	IN	VARCHAR2	No	Validation: Must be a valid lookup under FIN_NUM_NAME lookup type.
financial_units_applied	IN	NUMBER	No	
financial_number_currency	IN	VARCHAR2	No	
projected_actual_flag	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: validated against AR lookup type REGISTRY_STATUS
created_by_module	IN	VARCHAR2	Yes	

Other Validation

The following combination of columns must be unique: financial_report_id and financial_number_name.

Update Financial Number API

```
PROCEDURE update_financial_number(
    p_init_msg_list             IN     VARCHAR2 := FND_API.G_FALSE
,
    p_financial_number_rec      IN     FINANCIAL_NUMBER_REC_TYPE,
    p_object_version_number     IN OUT NOCOPY NUMBER,
    x_return_status             OUT NOCOPY VARCHAR2,
    x_msg_count                 OUT NOCOPY NUMBER,
    x_msg_data                  OUT NOCOPY VARCHAR2
);
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Financial Number 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
financial_number_id	IN	NUMBER	Yes	Validation: Not updateable.
financial_report_id	IN	NUMBER	Yes	Validation: Must exist in the HZ_FINANCIAL_REPORTS table. Non-updateable.
financial_number	IN	VARCHAR2	No	
financial_number_name	IN	VARCHAR2	No	Validation: Must be a valid lookup under FIN_NUM_NAME lookup type.
financial_units_applied	IN	NUMBER	No	
financial_number_currency	IN	VARCHAR2	No	
projected_actual_flag	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: validated against AR lookup type REGISTRY_STATUS
created_by_module	IN	VARCHAR2	Yes	

Other Validation

The following combination of columns must be unique: financial_report_id and financial_number_name.

Location APIs

PL/SQL Package Name: HZ_LOCATION_V2PUB

Java Class Name: HzLocationV2Pub

PL/SQL Constant:

```
G_MISS_CONTENT_COURCE_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),
    orig_system                     VARCHAR2(30),
    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),
    delivery_point_code                  VARCHAR2(50),
    location_directions                  VARCHAR2(640),
    address_effective_date               DATE,
    address_expiration_date              DATE,
    clli_code                            VARCHAR2(60),
    language                             VARCHAR2(4),
    short_description                    VARCHAR2(240),
    description                          VARCHAR2(2000),
    geometry                            MDSYS.SDO_GEOMETRY:= hz_geo
metry_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_CONTEN
T_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                   orig_system;
    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                   delivery_point_code;
    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

Description

This routine is used to create an Address Location. The API creates a record in the HZ_LOCATIONS table. The API also creates a record in the HZ_LOCATIONS_PROFILES table. That record stores address-specific information about the location. The location created by this API is just a physical location and can be used to create party site or customer account site. If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key. If timezone_id is not passed in, the API generates a time zone value based on the address components and time zone setup. However, if the user passes in the time zone the API keeps the time zone value that the user chose.

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 Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
country	IN	VARCHAR2	Yes	Validation: • Mandatory Attribute • Foreign Key to fnd_territories.territory_code
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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
suite	IN	VARCHAR2	No	
postal_plus4_code	IN	VARCHAR2	No	
position	IN	VARCHAR2	No	
delivery_point_code	IN	VARCHAR2	No	
location_directions	IN	VARCHAR2	No	
address_effective_ date	IN	DATE	No	
address_expiration_ date	IN	DATE	No	
clli_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: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. 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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	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 : Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. 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. The API also creates or updates a record in the HZ_LOCATIONS_PROFILES table. Whether to create or update a location profile record depends on the value of the HZ:Maintain Location History and HZ: Allow to Update Standardized Address profile options.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique. If timezone_id is not passed in, the API generates a time zone value based on the changes of the address components and time zone setup even if a time zone already exists in the database. However, if the user passes in the time zone the API keeps the time zone value that the user chose.

The loc_assignment records for this location are also updated.

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	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
floor	IN	VARCHAR2	No	
suite	IN	VARCHAR2	No	
postal_plus4_code	IN	VARCHAR2	No	
position	IN	VARCHAR2	No	
delivery_point_code	IN	VARCHAR2	No	
location_directions	IN	VARCHAR2	No	
address_effective_ date	IN	DATE	No	
address_expiration_ date	IN	DATE	No	
clli_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: 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	
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
actual_content_ source	IN	VARCHAR2	No	Validation: Cannot be updated.
p_object_version_ number	IN/OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing person location record Comment: • Pass the current object_version_number of the location record • Return new value after update.

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),
    orig_system                    VARCHAR2(30),
    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
    global_location_number         VARCHAR2(40)
)
```

PL/SQL Record Structure for Party Site Use

```
TYPE party_site_use_rec_type           IS RECORD
    party_site_use_id                  NUMBER,
    comments                           VARCHAR2(240),
    site_use_type                      VARCHAR2(30),
    party_site_id                      NUMBER,
    primary_per_type                   VARCHAR2(1),
    status                             VARCHAR2(1),
    created_by_module                  VARCHAR2(150),
    application_id                     NUMBER
)
```

Java Inner Class for Party Site

```
public static class PartySiteRec {
    public BigDecimal                   party_site_id;
    public BigDecimal                   party_id;
    public BigDecimal                   location_id;
    public String                       party_site_number;
    public String                       orig_system_reference;
    public String                       orig_system;
    public String                       mailstop;
    public String                       identifying_address_flag;
    public String                       status;
    public String                       party_site_name;
    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                       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. If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key.

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: • Mandatory attribute • Foreign Key to hz_parties.party_id

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
location_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign Key to hz_locations.location_id
party_site_number	IN	VARCHAR2	Yes/No	Validation: • Required when profile HZ_GENERATE_PARTY_SITE_NUMBER = N, else generated from sequence. • Unique when passed in
orig_system_reference	IN	VARCHAR2	No	Default: party_site_id Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
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	

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: 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.
global_location_number	IN	VARCHAR2	No	Validation: • Cannot have nonnumeric characters. • Must be 13 digits long. • Must satisfy the check-digit algorithm. The check-digit algorithm is: 1. Add the digits in even position. Multiply this sum by three. 2. Add the digits in odd position, excluding the thirteenth digit. 3. Add the values obtained in the previous two steps. The thirteenth digit should be the smallest number that should be added to make the sum obtained in step 3 a multiple of 10.
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 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 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 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 party site from the existing party sites, and then makes that one as identifying address. If the API cannot find any active 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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_ REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

If you update the status of a party site from Active to Inactive, then the party site uses and customer account sites associated with this party site are inactivated. The customer account site uses that belong to the associated customer account sites are also inactivated.

If you update the status of a party site from Inactive to Active, then the associated customer account sites to this party site are also updated and made active.

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
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: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*.
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 • Cannot be set to null during update
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	

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	
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
global_location_number	IN	VARCHAR2	No	Validation: • Cannot have nonnumeric characters. • Must be 13 digits long. • Must satisfy the check-digit algorithm. The check-digit algorithm is: 1. Add the digits in even position. Multiply this sum by three. 2. Add the digits in odd position, excluding the thirteenth digit. 3. Add the values obtained in the previous two steps. The thirteenth digit should be the smallest number that should be added to make the sum obtained in step 3 a multiple of 10.
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the record from HZ_PARTY_SITES • Return new value after update.

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 a party site use for a party site that is already present in the HZ_PARTY_SITES table.

The first active party site use that you create is identified as the primary party site use. The primary party site use cannot be inactive.

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: • Mandatory attribute • Validated against AR lookup type PARTY_SITE_USE_CODE
party_site_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign key to hz_party_sites.party_site_id
primary_per_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO. If the status is set to Inactive (I), then this value cannot be set to Yes (Y).
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.

If the primary site use for a particular site use type (primary_per_type value is *Y*) is set as inactive, then the earliest created active site use for the same site use type is set as primary. If a site use is activated and this becomes the only active site use for that site use type, then it is set as primary.

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: • Validated against AR lookup type REGISTRY_STATUS • Cannot be updated to null
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the record from hz_party_site_uses • Return new value after update

6

Contact Point API Use

This chapter describes Contact Point APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Contact Point APIs

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),
    orig_system                         VARCHAR2(30),
    content_source_type                VARCHAR2(30):= G_MISS_CONTEN
T_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_numberVARCHAR2(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                       orig_system;
    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.

If orig_system is passed in, the API also creates a record in the HZ_ORIG_SYS_ REFERENCES table to store the mapping between the source system reference and the TCA primary key. If timezone_id is not passed in, the API generates a time zone value based on the phone components and time zone setup. However, if the user passes in the time zone the API keeps the time zone value that the user chose.

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

Java Method for EMAIL Contact Points

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

Java Method for PHONE Contact Points

```
public static void createPhoneContactPoint(
    OracleConnection_connection,
    String                           p_init_msg_list,
    ContactPointRec                  p_contact_point_rec,
    PhoneRec                         p_phone_rec,
    BigDecimal [ ]                   x_contact_point_id,
    String [ ]                       x_return_status,
    BigDecimal [ ]                   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 tables list information about the parameters in the Create Contact Point API. The tables include 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: • Mandatory attribute • Validated against AR lookup type COMMUN ICATION_TYPE • EDI and EFT contact points must be Organization-type parties.
status	IN	VARCHAR2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
owner_table_name	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Validated against AR lookup type OWNER_ TABLE_NAME 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
owner_table_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Foreign Key hz_parties.party_id when owner_table_name = HZ_PARTIES. • Foreign Key to hz_party_sites.party_site_id when owner_table_name = HZ_PARTY_SITES
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 Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_ system.
content_source_type	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_ system with sst_flag value of *Y*. Default: USER_ENTERED Comment: This parameter is no longer used. Use actual_content_source.

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	
contact_point_ purpose	IN	VARCHAR2	No	Validation: • Validated against AR lookup type CONTACT_ POINT_PURPOSE when contact_point_type is not WEB • Validated against AR lookup type CONTACT_ POINT_PURPOSE_WEB when contact_point_ type=WEB
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.

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.
actual_content_source	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of *Y*. Default: USER_ENTERED

If you use the generic validation method, validations only apply when contact_point type = EDI.

p_edi_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
edi_transaction_handling	IN	VARCHAR2	No	
edi_id_number	IN	VARCHAR2	Yes	
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	

If you use the generic validation method, validations only apply when contact_point_type=EMAIL.

p_email_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
email_format	IN	VARCHAR2	No	Validation: Validated against AR lookup type EMAIL_FORMAT Default: MAILHTML
email_address	IN	VARCHAR2	Yes	Validation: Mandatory attribute

If you use the generic validation method, validations only apply when contact_point_type = PHONE or PAGER.

p_phone_rec record type

Parameter	Type	Data Type	Required	Validation, Default Comment
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	
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. If raw_phone_number is NULL, then you cannot update phone_number to NULL.
phone_extension	IN	VARCHAR2	No	
phone_line_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Validated against AR lookup type PHONE_L INE_TYPE
raw_phone_number	IN	VARCHAR2	Yes/No	Validation: Mandatory if phone_number is not passed in

If you use the generic validation method, validations only apply when contact_point_type = TELEX.

p_telex_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
telex_number	IN	VARCHAR2	Yes	Validation: Mandatory attribute

If you use the generic validation method, validations only apply when contact_point_type = WEB.

p_web_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
web_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute
url	IN	VARCHAR2	Yes	Validation: Mandatory attribute

If you use the generic validation method, validations only apply when contact_point_type = EFT.

p_eft_rec_record_type

Parameter	Type	Data Type	Required	Validation, Default, Comment
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	
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 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 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 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, email, phone contact point id, phone purpose, phone line type, phone country code, phone area code, phone number, and phone extension 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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_ REFERENCES table based on orig_system and orig_system_reference if they are not null and unique. If timezone_id is not passed in, the API generates a time zone value based on the changes of the phone components and time zone setup even if a time zone already exists in the database. However, if the user passes in the time zone, the API keeps the time zone value that the user chose.

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_R
EC,
    p_phone_rec             IN     PHONE_REC_TYPE:= G_MISS_PHONE_R
EC,
    p_telex_rec             IN     TELEX_REC_TYPE:= G_MISS_TELEX_R
EC,
    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 tables list information about the parameters in the Update Contact Point API. The tables include 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	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: • Validated against AR lookup type REGISTRY_ STATUS • Cannot be set to null during update
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
orig_system_ reference	IN	VARCHAR2	No	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system.
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	
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
contact_point_purpose	IN	VARCHAR2	No	Validation: • Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB • Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type=WEB
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.

If you use the generic validation method, validations only apply when contact_point_type = EDI.

p_edi_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
edi_transaction_handling	IN	VARCHAR2	No	
edi_id_number	IN	VARCHAR2	No	
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	

If you use the generic validation method, validations only apply when contact_point_type = EMAIL.

p_email_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
email_format	IN	VARCHAR2	No	Validation: • Validated against AR lookup type EMAIL_ FORMAT. • Cannot set to null during update
email_address	IN	VARCHAR2	No	Validation: Cannot be set to null during update

If you use the generic validation method, validations only apply when contact_point_type = PHONE or PAGER.

p_phone_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
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	
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: • Validated against AR lookup type PHONE_L INE_TYPE. • Cannot be updated to null
raw_phone_number	IN	VARCHAR2	No	Validation: Mandatory if phone_number is not passed in

If you use generic validations, validations only apply when contact_point_type = TELEX.

p_telex_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
telex_number	IN	VARCHAR2	No	Validation: Mandatory attribute

If you use the generic validation method, validations only apply when contact_point_type = WEB.

p_web_rec record type

Parameter	Type	Data Type	Required	Validation, Default, Comment
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

Parameter	Type	Data Type	Required	Validation, Default, Comment
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	
p_object_version_ number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record. Comment: • Pass the current object_version_number of the record from hz_contact_points • Return new value after update

Other Validations

- The Primary_flag is a lookup code of lookup type YES/NO. The API automatically marks the first activecontact 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 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 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, email, phone contact point id, phone purpose, phone line type, phone country code, phone area code, phone number, and phone extension 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.

7

Relationship and Hierarchy API Use

This chapter describes Relationship Type, Relationship, and Hierarchy Retrieval APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Relationship Type APIs
- Relationship APIs
- Hierarchy Retrieval APIs

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_R
EC_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                          p_init_msg_list,
    RelationshipTypeRec             p_relationship_type_rec,
    BigDecimal [ ]                  x_relationship_type_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 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: • Mandatory attribute • Validated against AR lookup type PARTY_ RELATIONS_TYPE
backward_rel_code	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Validated against AR lookup type PARTY_ RELATIONS_TYPE
direction_code	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Validated against AR lookup type DIRECTION_ CODE

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
hierarchical_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO. • If hierarchical_flag = *Y*, then allow_circular_ relationships must be *N*. • If hierarchical_flag = *Y*, then direction_code must be *P* or *C* • If hierarchical_flag = *N*, then multiple_parent_ allowed must be *Y*. • All relationship type records with same relationship_type value must have same value for hierarchical_flag. Default : *N* Comment : Indicates whether the relationship type is hierarchical.
create_party_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Default: N Comment: Indicates whether a denormalized party will be created for a relationship having this relationship type
allow_relate_to_self_ flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO Default: N
allow_circular_ relationships	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO • If hierarchical_flag = *Y*, then allow_circular_ relationships must be *N*. • If direction_code = *N*, the allow_circular_ relationships must be *Y*. • All relationship type records with same relationship_type value must have same value for allow_circular_relationships. Default: Y 7
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

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 module from which creation of record is initiated
multiple_parent_ allowed	IN	VARCHAR2	No	Validation : Validated against AR lookup type YES/NO • If hierarchical_flag = *N*, then multiple_parent_ allowed must be *Y*. • All relationship type records with same relationship_type value must have same value for multiple_parent_allowed. Comment : Indicates whether a child can have multiple parents. This is applicable when hierarchical_flag is *Y*. Default : *Y*
incl_unrelated_ entities	IN	VARCHAR2	No	Validation :Validated against AR lookup type YES/NO Comment : Indicates whether entities not having a relationship will be included in a hierarchy or not. This is applicable when hierarchical_flag is *Y*. Default : *N*
forward_role	IN	VARCHAR2	No	Validation: • Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type. • Forward_role should be unique. Default : Defaulted to 'USER_ROLE_'||to_ char(relationship_type_id) Comment : This describes the role a subject party plays in a relationship.
backward_role	IN	VARCHAR2	No	Validation: • Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type. • Backward_role should be unique. 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 a relationship type, relationship phrase (code), subject_type, and object_type results in a unique relationship phrase (code) in the reverse direction.
    - The combination of a forward relationship code, subject type, and object type identify a unique backward relationship code. Therefore, another record with same combination of forward relationship code, subject type, and object type has the same backward relationship code.

      For example, if A and B have the same relationship type, the forward relationship code of A is the same as backward relationship code of B, the subject type of A is the same as the object type of B and the object type of A is the same as the subject type of B, then the backward relationship code of A must be the same as the forward relationship code of B.
    - The combination of a backward relationship code, subject type, and object type in a relationship_type similarly identifies a unique forward relationship code.

      For example, if A and B have the same relationship type, the backward relationship code of A is the same as forward relationship code of B, and the subject type of A is the same as the object type of B and the object type of A is the same as the subject type of B, then the forward relationship code of A must be the same as that of the backward relationship code of B.
- 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.
- If forward_rel_code and backward_rel_code are the same, then the subject_type and object_type should also be the 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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
create_party_flag	IN	VARCHAR2	No	Validation: • Can be updated. No relationship record, created with the current setup of create_party_flag, can exist with this relationship type. • Validate against AR lookup type YES/NO.
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
status	IN	VARCHAR2	No	Validation: • Validated against AR lookup type CODE_STATUS • Cannot set to null during update
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the record from hz_relationship_types • Return new value after update

Other Validations

- The combination of relationship type, forward relationship code, backward relationship code, subject type, and object_type should be unique.
- The combination of a relationship type, relationship phrase (code), subject_type and object_type results in a unique relationship phrase (code) in the reverse direction.
- 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_CONTEN
T_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                          party_rec;
    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.

There are two signatures for this API. One accepts p_create_org_contact as a parameter, the other does not. In the second case, p_create_org_contact is defaulted to *Y*, and the other overloaded procedure is called.

> **Important:** Do not use the Create Relationship API to create D&B hierarchy relationships. See: D&B Hierarchy, *Oracle Trading Community Architecture User Guide*.

PL/SQL Procedure

The p_create_org_contact parameter is not in the overloaded 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
    p_create_org_contact    IN     VARCHAR)
```

Java Method

The p_create_org_contact parameter is not in the overloaded procedure.

```
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
    String                          p_create_org_contact
) throws SQLException;
```

Parameter Description and Validation

The following tables list information about the parameters in the Create Relationship API. The tables include 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: • Mandatory attribute. • Validated against Primary Key in fnd_objects.obj_name where fnd_objects.object_id = fnd_object_instance_sets.object_id and fnd_object_instance_sets.instance_set_name= subject_type
subject_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to fnd_object_instance_sets. instance_set_name
subject_table_name	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to fnd_objects.obj_name.
object_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Validated against Primary Key in fnd_objects. obj_name where fnd_objects.object_id=fnd_ object_instance_sets.object_id and fnd_object_ instance_sets.instance_set_name=subject_type
object_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to fnd_object_instance_sets. instance_set_name

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
object_table_name	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign Key to fnd_objects.obj_name
relationship_code	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Validated against AR lookup type PARTY_ RELATIONS_TYPE • Required to be a valid relationship code for the particular relationship type requested.
relationship_type	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to hz_relationship_types. relationship_type
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: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. 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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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

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
actual_content_source	IN	VARCHAR2	No	Validation : Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of *Y*. Default : USER_ENTERED

Party_rec Record Type Attributes

Parameter	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated by sequence
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
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
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	

Parameter	Type	Data Type	Required	Validation, Default, Comment
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_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. This parameter is not in the overloaded procedure.

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 2nd relationship code from the HZ_RELATIONSHIP_TYPES table, and use it to create a second relationship. The 2nd 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_R
EC_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 tables list information about the parameters in the Update Relationship API. The tables include 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: • Validated 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: Cannot be updated.
attribute_category	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required Validation, Default, Comment
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
additional_ information6	IN	VARCHAR2	No
additional_ information7	IN	VARCHAR2	No

Parameter Name	Type	Data Type	Required Validation, Default, Comment
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
additional_ information25	IN	VARCHAR2	No
additional_ information26	IN	VARCHAR2	No
additional_ information27	IN	VARCHAR2	No

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: Value will not be updated in the database.

Party_rec Record Type Attributes

Parameter	Type	Data Type	Required	Validation, Default, Comment
party_id	IN	NUMBER	No	Validation: Valid party_id should be passed in to update party sequence Comment: • Pass the party_id from the hz_parties record • Pass only if you want to update the party record
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
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system.
status	IN	VARCHAR2	No	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	
attribute3	IN	VARCHAR2	No	

Parameter	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
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	Type	Data Type	Required	Validation, Default, Comment
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute for relationship record • Validated against value in the database for the existing record Comment: • Pass the current object_version_number from hz_relationships table • Return new value after update
p_party_object_version_number	IN OUT	NUMBER	Yes/No	Validation: • Mandatory attribute if party record for the relationship needs to be updated • Validated against value in the database for the existing record Comment: • Pass the current object_version_number from hz_parties if you want to update party record • Return new value after update

Other Validations

- 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 2nd relationship code from the HZ_RELATIONSHIP_TYPES table, and use it to create a second relationship. The 2nd 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.

Hierarchy Retrieval APIs

PL/SQL Package Name: HZ_HIERARCHY_V2PUB

PL/SQL Record Structure

```
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 IN
DEX BY BINARY_INTEGER;
```

Java Inner Class

```
public static class RelatedNodesListRec {
    public BigDecimal related_node_id;
    public String related_node_table_name;
    public String related_node_object_type;
    public BigDecimal level_number;
    public String top_parent_flag;
    public String leaf_child_flag;
    public java.sql.Timestamp effective_start_date;
    public java.sql.Timestamp effective_end_date;
    public BigDecimal relationship_id;
    public RelatedNodesListRec() {};
    public RelatedNodesListRec(boolean __RosettaUseGMISSValues)
)
```

Parameter Description and Validation

The following tables list information about the parameters in the Hierarchy Retrieval API. The tables include 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'
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

Parameter	Type	Data Type	Required	Validation, Default, Comment
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
)
```

Java Method

```
public static void isTopParent(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_hierarchy_type,
    BigDecimal p_parent_id,
    String p_parent_table_name,
    String p_parent_object_type,
    java.sql.Timestamp p_effective_date,
    String  [] x_result,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data  ) throws SQLException;
```

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
)
```

Java Method

```
public static void checkParentChild(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_hierarchy_type,
    BigDecimal p_parent_id,
    String p_parent_table_name,
    String p_parent_object_type,
    BigDecimal p_child_id,
    String p_child_table_name,
    String p_child_object_type,
    java.sql.Timestamp p_effective_date,
    String  [] x_result,
    BigDecimal  [] x_level_number,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data  ) throws SQLException;
```

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
)
```

Java Method

```
public static void getParentNodes(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_hierarchy_type,
    BigDecimal p_child_id,
    String p_child_table_name,
    String p_child_object_type,
    String p_parent_table_name,
    String p_parent_object_type,
    String p_include_node,
    java.sql.Timestamp p_effective_date,
    BigDecimal p_no_of_records,
    RelatedNodesListRec [][] x_related_nodes_list,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data  ) throws SQLException;
```

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
)
```

Java Method

```
public static void getChildNodes(
        OracleConnection _connection,
        String p_init_msg_list,
        String p_hierarchy_type,
        BigDecimal p_parent_id,
        String p_parent_table_name,
        String p_parent_object_type,
        String p_child_table_name,
        String p_child_object_type,
        String p_include_node,
        java.sql.Timestamp p_effective_date,
        BigDecimal p_no_of_records,
        RelatedNodesListRec [][] x_related_nodes_list,
        String [] x_return_status,
        BigDecimal [] x_msg_count,
        String [] x_msg_data
 ) throws SQLException;
```

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
)
```

Java Method

```
public static void getTopParentNodes(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_hierarchy_type,
    String p_parent_table_name,
    String p_parent_object_type,
    java.sql.Timestamp p_effective_date,
    BigDecimal p_no_of_records,
    RelatedNodesListRec [][] x_top_parent_list,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data  ) throws SQLException;
```

8

Classification API Use

This chapter describes Classification APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Classification APIs

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
    delimiter VARCHAR2(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,
    owner_table_key_1                    VARCHAR2(255)
    owner_table_key_2                    VARCHAR2(255)
    owner_table_key_3                    VARCHAR2(255)
    owner_table_key_4                    VARCHAR2(255)
    owner_table_key_5                    VARCHAR2(255)
    class_category                       VARCHAR2(30),
    class_code                           VARCHAR2(30),
    primary_flag                         VARCHAR2(1),
    content_source_type                  VARCHAR2(30):= HZ_PARTY_V
2PUB.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: • Unique • A Class Category correspond to a LOOKUP_TYPE
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
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                              p_init_msg_list,
    ClassCategoryRec                    p_class_category_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 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: • Unique • A Class Category correspond to a LOOKUP_ TYPE
allow_multi_parent_ flag	IN	VARCHAR2	No	Validation: • Validated against FND lookup type YES/NO • This flag cannot be updated from Y to N, if there are any class codes related to more than one parent code
allow_multi_assign_ flag	IN	VARCHAR2	No	Validation: • Validated against FND lookup type YES/NO • 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.
allow_leaf_node_ only_flag	IN	VARCHAR2	No	Validation: • Validated against FND lookup type YES/NO • 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.
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number from HZ_CLASS_CATEGORIES • Return new value after update

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_RELATIO
N_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
class_code	IN	VARCHAR2	Yes	Validation: • Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column • sub_class_code cannot be an ancestor code of class_code. This validation avoids recursive relationship
sub_class_code	IN	VARCHAR2	Yes	Validation: • Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column • class_code cannot be a descendant of sub_class_code. This validation avoid recursive relationship • 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.
start_date_active	IN	DATE	No	
end_date_active	IN	DATE	No	Validation: • A Class Code Relation must be unique for a period of time range from start_date_active to end_date_active. • The end_date_active must be null or greater then the start_date_active.
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 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: • 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. • End_date_Active must be null or greater than Start_Date_Active.
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the record from hz_class_code_relations • Returns new value after update

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.

The primary code assignment of type CUSTOMER_CATEGORY is denormalized into the HZ_PARTIES table. The primary SIC code assignment is denormalized into HZ_PARTIES and HZ_ORGANIZATION_PROFILES.

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
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: One or the other of owner_table_id or owner_table_ key_1 must be supplied. Both cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified.
owner_table_key_1	IN	VARCHAR2	No	Validation: One or the other of owner_table_id or owner_table_ key_1 must be supplied. Both cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified. Not updateable.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
owner_table_key_2	IN	VARCHAR2	No	Validation: If owner_table_key_1 is not supplied, then this parameter cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified. Not updateable.
owner_table_key_3	IN	VARCHAR2	No	Validation: If owner_table_key_2 is not supplied, then this parameter cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified. Not updateable.
owner_table_key_4	IN	VARCHAR2	No	Validation: If owner_table_key_3 is not supplied, then this parameter cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified. Not updateable.
owner_table_key_5	IN	VARCHAR2	No	Validation: If owner_table_key_4 is not supplied, then this parameter cannot be supplied. If supplied the value must correspond to the primary key value of the entity which is being classified. Not updateable.
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: • Validated against FND lookup values where lookup type YES/NO • An owner_table_id can only have one primary (PRIMARY_FLAG = Y) assignment to a class_code of one class_category for one actual_content_source at one time.
content_source_type	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of *Y*. Default: USER_ENTERED
start_date_active	IN	DATE	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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
actual_content_source	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of *Y*.
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.

The primary code assignment of type CUSTOMER_CATEGORY is denormalized into the HZ_PARTIES table. The primary SIC code assignment is denormalized into HZ_PARTIES and HZ_ORGANIZATION_PROFILES.

PL/SQL Procedure

```
PROCEDURE update_code_assignment(
    p_init_msg_list                     IN          VARCHAR2:= FND
_API.G_FALSE,
    p_code_assignment_rec               IN          CODE_ASSIGNMEN
T_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                              p_init_msg_list,
    CodeAssignmentRec                   p_code_assignment_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 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_ASS IGNMENTS.CODE_ASSIGNMENT_ID
owner_table_name	IN	VARCHAR2	No	
owner_table_id	IN	NUMBER	No	
class_category	IN	VARCHAR2	Yes	
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	
end_date_active	IN	DATE	No	Comment: Cannot create classification with time period that overlaps an existing classification. Validation: Done in HZ_CLASS_VALIDATE_V2. validate_code_assignment
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: • Mandatory attribute • Validated against value in the database for the existing records Comment: • Pass the current object_version_number of the record from hz_code_assignments • Returns new value after update
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: • Validated against FND lookup values where Lookup type = CODE_ASSIGN_OWNER_ TABLE • The combination (class_category, owner_table) must be unique.
column_name	IN	VARCHAR2	No	
additional_where_ clause	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

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: • Validated against FND lookup type 'CODE_ASSIGN_OWNER_TABLE' • The combination (class_category, owner_table) must be unique.
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: • Mandatory attribute • Validated against value in the database for the existing records Comment: • Pass the current object_version_number of the record from hz_class_category_uses • Returns new value after update

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,
    p_key_1                        IN         NUMBER,
    p_key_2                        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	Default: NULL
p_key_1	IN	VARCHAR2		Default: NULL
p_key_2	IN	VARCHAR2		Default: NULL

9

Contact Preference API Use

This chapter describes Contact Preference APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Contact Preference APIs

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: • Mandatory attribute • Non updateable • Validated against AR lookup type SUBJECT_TABLE
contact_level_table_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • Not updateable • Is the foreign key of contact_level_table
contact_type	IN	VARCHAR2	Yes	Validation: • Mandatory Attribute. • Non-updateable • contact_type is lookup code in lookup type CONTACT_TYPE
preference_code	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Should be validated against the PREFERENCE_CODE lookup type

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_topic_type	IN	VARCHAR2	N	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: TABLES: • AMS_SOURCE_CODES, • AS_INTEREST_TYPES_B, • AS_INTEREST_CODES_B • LOOKUP_TYPE: CONTACT_USAGE
preference_topic_type_id	IN	NUMBER	N	Validation : Valid if the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values: • AMS_SOURCE_CODES • AS_INTEREST_TYPES_B • AS_INTEREST_CODES_B The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected PREFERENCE_TOPIC_TYPE.
preference_topic_type_code	IN	VARCHAR2	N	Validation: If PREFERENCE_TOPIC_TYPE = CONTACT_USAGE, then PREFERENCE_TOPIC_TYPE_CODE should be a lookup code of lookup type CONTACT_USAGE. Validation exists to ensure that this lookup code exists. If PREFERENCE_TOPIC_TYPE is FND_BUSINESS_PURPOSES_B, then PREFERENCE_TOPIC_TYPE_CODE is the name of a column in FND_BUSINESS_PURPOSES_B. No validation exists to ensure that the value passed is a proper column name.
preference_start_date	IN	DATE	Y	Validation: Mandatory attribute.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_end_date	IN	DATE	N	Validation: • Cannot be updated to a day before the sysdate • PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE • 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.
preference_start_time_hr	IN	NUMBER	N	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	N	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_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: • Mandatory attribute • Should be validated against the REQUESTED_BY lookup type

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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*.
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

CONTACT_TYPE allowed	when CONTACT_POINT_TYPE is	and CONTACT_LEVEL_TABLE is
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
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:= FN
D_API.G_FALSE,
    p_contact_preference_rec              IN         CONTACT_PREFER
ENCE_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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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 TABLES • AMS_SOURCE_CODES • AS_INTEREST_TYPES_B • AS_INTEREST_CODES_B • LOOKUP_TYPE
preference_topic_type_id	IN	NUMBER	No	Validation: If the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values: • AMS_SOURCE_CODES • AS_INTEREST_TYPES_B • AS_INTEREST_CODES_B The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected in the attribute PREFERENCE_TOPIC_TYPE.
preference_topic_type_code	IN	VARCHAR2	No	Validation: If PREFERENCE_TOPIC_TYPE = CONTACT_USAGE, then PREFERENCE_TOPIC_TYPE_CODE should be a lookup code of lookup type CONTACT_USAGE. Validation exists to ensure that this lookup code exists. If PREFERENCE_TOPIC_TYPE is FND_BUSINESS_PURPOSES_B, then PREFERENCE_TOPIC_TYPE_CODE is the name of a column in FND_BUSINESS_PURPOSES_B. No validation exists to ensure that the value passed is a proper column name.
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: • PREFERENCE_END_DATE can not update to a day before sysdate • PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE,

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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
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
status	IN	VARCHAR2	No	Validation: • Should be validated against the CODE_STATUS lookup type • The PREFERENCE_END_DATE attribute should be set to sysdate when the STATUS column has a value other than *A* • Can not be set to NULL during update
created_by_module	IN	VARCHAR2	No	Validation: Not updateable if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the contact preference record • Return new value after update

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

CONTACT_TYPE allowed	when CONTACT_POINT_TYPE is	and CONTACT_LEVEL_TABLE is
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
EDI	EDI	HZ_CONTACT_POINTS
	NONE	HZ_PARTY_SITES
	NONE	HZ_PARTIES

10

Customer Account API Use

This chapter describes Customer Account APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Customer Account APIs

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),
orig_system                      VARCHAR2(30),
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                   orig_system;
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                   attribute13;
    public String                   attribute14;
    public String                   attribute15;
    public String                   bill_to_flag;
    public String                   ship_to_flag;
    public String                   created_by_module;
    public BigDecimal               application_id;

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

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. If an orig_system_reference is passed in, the API creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key. If orig_system_reference is not passed in, the default is UNKNOWN.

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_TYP
E,
    p_customer_profile_rec IN        HZ_CUSTOMER_PROFILE_V2PUB.CUS
TOMER_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_TY
PE,
    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_profi
le_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_prof
ile_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: • Mandatory Attribute. If GENERATE_CUSTOMER_NUMBER of AR_SYSTEM_PARAMETERS is on, if user has passed in an account_number, error out • account_number will be generated from sequence. If autonumbering is off, if user has not passed in value, error out
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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
global_attribute20	IN	VARCHAR2	N	
orig_system_ reference	IN	VARCHAR2	N	Validation: unique if passed in Default: cust_account_id
orig_system	IN	VARCHAR2	N	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system. Default: UNKNOWN if an orig_system_reference is passed in.
status	IN	VARCHAR2	N	Validation: • status is lookup code in lookup type CODE_STATUS • status cannot be set to null during update. It is defaulted to 'A' if user does not pass any value
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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
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	
account_termination_ date	IN	DATE	N	Validation: • account_termination_date should be greater than account_established_date • account_termination_date should be greater than account_activation_date
account_activation_ date	IN	DATE	N	Validation: account_activation_date should be greater than account_established_date

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
ship_sets_include_lines_flag	IN	VARCHAR2	N	Validation: • ship_sets_include_lines_flag is lookup code in lookup type YES/NO • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N. Default: N
arrivalsets_include_lines_flag	IN	VARCHAR2	N	Validation: • arrivalsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_party_number	OUT	NUMBER	N	Comment: Returns party_number of the organization or person created
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_ REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

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 if automatic numbering is on. If not, then account_number can be updated.
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	
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	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
orig_system_ reference	IN	VARCHAR2	No	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	N	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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.
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
account_ termination_date	IN	DATE	No	Validation: • account_termination_date should be greater than account_establish • account_termination_date should be greater than account_activation_date
account_activation_ date	IN	DATE	No	Validation: account_activation_date should be greater than account_established_date
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: • shipsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.
arrivalsets_include_ lines_flag	IN	VARCHAR2	No	Validation: • arrivalsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.
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
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: • Mandatory attribute • Validated against value in the database for the existing record. Comment • Pass the current object_version_number of the customer account record • Return new value after update.

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_RELA
TE_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: • Mandatory attribute • elated_cust_account _id is foreign key of hz_cust_accounts
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: 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 for all active customer accounts in an organization.

Update Customer Account Relationship API

Description

This routine is used to update a Customer Account Relationship. The API updates accounts in the HZ_CUST_ACCT_RELATE table.

There are two overloaded procedures for this API. One updates active customer account relationship records only. The other with rowid in the signature updates customer account relationship records based on the rowid.

PL/SQL Procedure

The p_rowid parameter is only in the overloaded procedure, and is used to identify the exact customer account relationship record to update.

```
PROCEDURE update_cust_acct_relate (
    p_init_msg_list                     IN          VARCHAR2:= FND
_API.G_FALSE,
    p_cust_acct_relate_rec              IN          CUST_ACCT_RELA
TE_REC_TYPE,
    p_rowid                             IN          ROWID,
    p_object_version_number             IN OUT      NUMBER,
    x_return_status                     OUT         VARCHAR2,
    x_msg_count                         OUT         NUMBER,
    x_msg_data                          OUT         VARCHAR2
)
```

Java Method

The p_rowid parameter is only in the overloaded procedure, and is used to identify the exact customer account relationship record to update.

```
public static void updateCustAcctRelate(
    OracleConnection_connection,
    String                              p_init_msg_list,
    CustAcctRelateRec                   p_cust_acct_relate_rec,
    String                              p_rowid,
    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: • Status cannot be set to null during update • Status is lookup code in lookup type CODE_STATUS
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	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_rowid	IN	ROWID	No	Comment: This is only in the overloaded procedure of this API.
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the cust account relate record • Return new value after update

Other Validations

The combination of cust_account_id and related_cust_account_id should be unique for all active customer accounts in an organization.

11

Customer Account Site API Use

This chapter describes Customer Account Site APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Customer Account Site APIs

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),
    orig_system                    VARCHAR2(30),
    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),
    orig_system                    VARCHAR2(30),
    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 BigDecimal                   party_site_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                       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                       orig_system;
    public String                       status;
    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                   orig_system;
    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_unearned;
    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. If an orig_system_reference is passed in, the API creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key. If orig_system_reference is not passed in, the default is UNKNOWN. When you use this API to create a new Customer Account Site, the status is inherited from the corresponding 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: • Mandatory attribute • cust_account_id is foreign key of hz_cust_accounts
party_site_id	IN	NUMBER	Yes	Validation: • Mandatory attribute • party_site_id is foreign key of hz_party_sites[2] • party_site_id must link to a location of content source type 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	

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	
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	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
orig_system_ reference	IN	VARCHAR2	No	Validation: unique if passed in Default: cust_acct_site_id
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system. Default: UNKNOWN if an orig_system_reference is passed in.
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
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

Use this routine to update a Customer Account Site. The API updates records in the HZ_CUST_ACCT_SITES table.

If the primary key is not passed in, then get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference, which must be unique and cannot be null.

When you change the status of an existing Customer Account Site to Inactive, you also cause the status of the following to change to Inactive;

- the corresponding Party Site
- all Customer Account Sites associated with that Party Site
- all Customer Account Site Uses for those Customer Account Sites

When you change the status of an existing Customer Account Site to Active, you also cause the status of the following to change to Active;

- the corresponding Party Site
- all Customer Account Sites associated with that Party Site
- all Customer Account Site Uses for those Customer Account Sites

When you update a Customer Account Site, you also update the corresponding loc_assignment record in the HZ_LOC_ASSIGNMENTS 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                               p_init_msg_list,
    CustAcctSiteRec                      p_cust_acct_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 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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
orig_system_ reference	IN	VARCHAR2	No	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Foreign key to HZ_ORIG_SYSTEMS.orig_system.
status	IN	VARCHAR2	No	Validation: • Status cannot be set to null during update. • Status is lookup code in lookup type CODE_STATUS.
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass current object_version_number of the record from hz_cust_acct_sites • Return new value after update

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. If an orig_system_reference is passed in, the API creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key. If orig_system_reference is not passed in, the default is UNKNOWN.

The first active Customer Account Site Use that you create becomes the primary site use for every customer and organization combination. When you create a new active, primary Customer Account Site Use, the new Customer Account Site Use becomes the new primary site use.

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.CUS
TOMER_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_profil
e_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: • Mandatory attribute • cust_account_id is foreign key to hz_cust_acct_sites
site_use_code	IN	VARCHAR2	No	Validation: • Mandatory attribute • site_use_code is lookup type YES/NO
primary_flag	IN	VARCHAR2	No	Validation: Primary_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
location	IN	VARCHAR2	No	Validation: • Mandatory attribute • It will be generated from sequence if user does not pass in and AUTO_SITE_NUMBERING in AR_SYSTEM_PARAMETERS is on • location must be unique within a customer account/ site_use_type
bill_to_site_use_id	IN	NUMBER	No	
orig_system_reference	IN	VARCHAR2	No	Validation: site_use_id
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system . Default: UNKNOWN if an orig_system_reference is passed in.
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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_L ISTS.
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	
attribute19	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_ category	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
ship_sets_include_lines_flag	IN	VARCHAR2	No	Validation: • shipsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N. Default: N
arrivalsets_include_lines_flag	IN	VARCHAR2	No	Validation: • arrivalsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N. 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: 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 creating a customer 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, which is active, 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.

If the primary key is not passed in, then get the primary key from the HZ_ORIG_SYS_REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

If the primary flag is reset to Y for a new or existing Customer Account Site Use, then the existing primary Customer Account Site Use is reset to N.

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
primary_flag	IN	VARCHAR2	No	Validation: • primary_flag cannot be set to null during update • primary_flag is lookup code in lookup type YES/NO
status	IN	VARCHAR2	No	Validation: • status cannot be set to null during update • status is lookup code in lookup type CODE_STATUS
location	IN	VARCHAR2	No	Validation: Can only be updated if the AUTO_SITE_NUMBERING profile option in AR_SYSTEMS_PARAMETERS is unchecked.
bill_to_site_use_id	IN	NUMBER	No	
orig_system_ reference	IN	VARCHAR2	No	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system.
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
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_L ISTS.
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	

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	
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	
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
ship_sets_include_ lines_flag	IN	VARCHAR2	No	Validation: • shipsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.
arrivalsets_include_ lines_flag	IN	VARCHAR2	No	Validation: • arrivalsets_include_lines_flag is lookup code in lookup type YES/NO. • If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N. • If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.
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.
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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
p_object_version_number	IN/OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record. Comment: • Pass current object_version_number of the record from hz_cust_site_uses • Return new value after update

Other Validations

cust_acct_site_id and site_use_code together should be unique.

12

Customer Profile and Customer Account Role API Use

This chapter describes Customer Profile and Customer Account Role APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Customer Profile APIs
- Customer Account Role APIs

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_fla
g;
    public String                    tax_printing_option;
    public String                    charge_on_finance_charge_f
lag;
    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.

If the credit_hold parameter value is *Y*, then records are inserted into OE_HOLD_SOURCES and OE_ORDER_HOLDS to keep these tables synchronized.

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	Validation:
				• Foreign key to hz_cust_accounts.cust_account_id
				• One of the following 2 columns need to be not null : cust_account_id and party_id
				• If cust_account_id is not null and party_id is null then:
				  • The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.
				  • The party_id in the hz_parties table associated with this cust_account_id must be active.
				• If cust_account_id is null and party_id is not null then:
				  • The party_id is a foreign key to an active hz_parties.party_id.
				  • The cust_account_id will be defaulted to (-1).
				• If cust_account_id is not null and party_id is not null then:
				  • The party_id is a foreign key to an active hz_parties.party_id.
				  • The party_id and the cust_account_id must be associated in hz_cust_accounts.
				• 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.
				• 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.

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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: • It is lookup code in lookup YES/NO. • If discount_terms = Y, then discount_grace_days should be greater than or equal to 0. • If discount_terms = N, then discount_grace_days should be null. Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)
dunning_letters	IN	VARCHAR2	No	Validation: • Dunning letters is lookup code in lookup type YES/NO. • If dunning_letters = Y, then dunning_letters_ set_id must have a value. • If dunning_letters = N, then dunning_letter_set_ id must be null. Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)
interest_charges	IN	VARCHAR2	No	Validation: • It is a lookup code in lookup type YES/NO. • 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. • If interest_charges = N, then charge_on_ finance_charge_flag must be null and interest_period_days must be null. Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
send_statements	IN	VARCHAR2	No	Validation: • It is a lookup code in lookup type YES/NO. • If send_statements = Y, then statement_cycle_id must have a value and credit_balance_ statements must have value. • If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N. Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)
credit_balance_ statements	IN	VARCHAR2	No	Validation: • It is a lookup code in lookup type YES/NO. • Can be populated if send_statements is Y. Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)
credit_hold	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)
profile_class_id	IN	NUMBER	No	Validation: • If passed in, profile_class_id should be positive. • Foreign key to hz_cust_profile_classes and the corresponding profile class should be active Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID= 0)
site_use_id	IN	NUMBER	No	Validation: • Foreign key to hz_cust_site_uses • One site use can only have one profile. • 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.
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: • If dunning_letters = Y, then dunning_letter_set_id must have a value. • If dunning_letters = N, then dunning_letter_set_id must be null. • Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.
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: • Must be a valid statment_cycle_id from AR_STATEMENT_CYCLES. • Can be populated if send_statements is Y.
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	
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, Comment
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	
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	
jgzz_attribute9	IN	VARCHAR2	No	
jgzz_attribute10	IN	VARCHAR2	No	
jgzz_attribute11	IN	VARCHAR2	No	
jgzz_attribute12	IN	VARCHAR2	No	
jgzz_attribute13	IN	VARCHAR2	No	
jgzz_attribute14	IN	VARCHAR2	No	
jgzz_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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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	Validation:
				• One of the following columns cannot be null: cust _account_id and party_id.
				• If cust_account_id is not null and party_id is null, then this is mandatory field.
				  • The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.
				  • The party_id in the hz_parties table associated with this cust_account_id must be active.
				• If cust_account_id is null and party_id is not null then
				  • The party_id is a foreign key to an active hz_parties.party_id.
				The cust_account_id will be defaulted to (-1).
				• If cust_account_id is not null and party_id is not null:
				  • The party_id is a foreign key to an active hz_parties.party_id.
				  • Party_id and cust_account_id must be associated in hz_cust_accounts.
				• You can only have one customer profile for one account. You can find the party_id in the hz_customer _profiles.party_id column.
				• You can only have one customer profile at party level for a party. The cust_account_id defaults to (-1).

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.

If the credit_hold parameter value is *Y*, then records are inserted into OE_HOLD_SOURCES and OE_ORDER_HOLDS to keep these tables synchronized.

PL/SQL Procedure

```
PROCEDURE update_customer_profile (
  p_init_msg_list                         IN         VARCHAR2:= FND
_API.G_FALSE,
  p_customer_profile_rec                  IN         CUSTOMER_PROFI
LE_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

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
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	
tolerance	IN	NUMBER	No	Validation: • Tolerance cannot be updated to null. • Must be between -100 and 100.
discount_terms	IN	VARCHAR2	No	Validation: • It is lookup code in lookup YES/NO. • If discount_terms = Y, then discount_grace_days should be greater than or equal to 0. • If discount_terms = N, then discount_grace_days should be null.
dunning_letters	IN	VARCHAR2	No	Validation: • It is lookup code in lookup YES/NO. • If dunning_letters = Y, then dunning_letters_set_id must have a value. • If dunning_letters = N, then dunning_letter_set_id must be null.
interest_charges	IN	VARCHAR2	No	Validation: • It is lookup code in lookup type YES/NO. • 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. • If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.
send_statements	IN	VARCHAR2	No	Validation: • It is lookup code in lookup type YES/NO. • If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have a value. • If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
credit_balance_ statements	IN	VARCHAR2	No	Validation: • It is a lookup code in lookup type YES/NO. • Can be populated if send_statments is Y.
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.
dunning_letter_set_ id	IN	NUMBER	No	Validation: • If dunning_letters = Y, then dunning_letters_ set_id must have a value. • If dunning_letters = N, then dunning_letter_set_ id must be null. • Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.
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: • Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES. • Can be populated if send_statements is Y.
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
auto_rec_incl_disputed_flag	IN	VARCHAR2	No	Validation: • Cannot be set to null • It is a lookup code in lookup type YES/NO
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	
jgzz_attribute5	IN	VARCHAR2	No	
jgzz_attribute6	IN	VARCHAR2	No	
jgzz_attribute7	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
jgzz_attribute8	IN	VARCHAR2	No	
jgzz_attribute9	IN	VARCHAR2	No	
jgzz_attribute10	IN	VARCHAR2	No	
jgzz_attribute11	IN	VARCHAR2	No	
jgzz_attribute12	IN	VARCHAR2	No	
jgzz_attribute13	IN	VARCHAR2	No	
jgzz_attribute14	IN	VARCHAR2	No	
jgzz_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	
global_attribute_ category	IN	VARCHAR2	No	
cons_inv_flag	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass the current object_version_number of the record • Return new value after update
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_A
MT_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: • Mandatory attribute • When p_check_foreign_key is FND_API.G_TRUE, cust_account_profile_id must be a foreign key to hz_customer_profiles
currency_code	IN	VARCHAR2	Yes	Validation: • Mandatory attribute • Foreign key to fnd_currencies • For a given cust_account_profile_id and currency_code, only one record of the profile amount is allowed.
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
site_use_id	IN	NUMBER	No	Validation: • When p_check_foreign_key is FND_API.G_TRUE, site_use_id must be a foreign key to hz_cust_site_uses • 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.
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	
jgzz_attribute9	IN	VARCHAR2	No	
jgzz_attribute10	IN	VARCHAR2	No	
jgzz_attribute11	IN	VARCHAR2	No	
jgzz_attribute12	IN	VARCHAR2	No	
jgzz_attribute13	IN	VARCHAR2	No	
jgzz_attribute14	IN	VARCHAR2	No	
jgzz_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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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 the Customer 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_R
EC_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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
jgzz_attribute6	IN	VARCHAR2	No	
jgzz_attribute7	IN	VARCHAR2	No	
jgzz_attribute8	IN	VARCHAR2	No	
jgzz_attribute9	IN	VARCHAR2	No	
jgzz_attribute10	IN	VARCHAR2	No	
jgzz_attribute11	IN	VARCHAR2	No	
jgzz_attribute12	IN	VARCHAR2	No	
jgzz_attribute13	IN	VARCHAR2	No	
jgzz_attribute14	IN	VARCHAR2	No	
jgzz_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	
global_attribute_ category	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass current object_version_number of the record from hz_cust_profile_amts table • Return new value after update

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),
    orig_system                          VARCHAR2(30),
    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                       orig_system;
    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                   orig_system;
    public String                   created_by_module;
    public BigDecimal               application_id;

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

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. If an orig_system_reference is passed in, the API creates a record in the HZ_ORIG_SYS_REFERENCES table to store the mapping between the source system reference and the TCA primary key. If orig_system_reference is not passed in, the default is UNKNOWN.

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_R
EC_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: • Mandatory attribute • Foreign key to hz_parties
cust_account_id	IN	NUMBER	Yes	Validation: • Mandatory field • Foreign key to hz_cust_accounts
cust_acct_site_id	IN	NUMBER	No	Validation: • Foreign key to hz_cust_acct_sites • 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
primary_flag	IN	VARCHAR2	No	Validation: • Primary flag is lookup code in lookup type YES/NO • It is unique per cust_account_id or cust_acct_site_id Default: N

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
role_type	IN	VARCHAR2	Y	Validation: • Mandatory attribute • It is a lookup code in lookup type ACCT_ROLE_TYPE[2] • The combination of CUST_ACCOUNT_ID, PARTY_ID, ROLE_TYPE should be unique.I. • Or the combination of CUST_ACCT_SITE_ID, PARTY_ID, ROLE_TYPE should be unique
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute24	IN	VARCHAR2	No	
orig_system_ reference	IN	VARCHAR2	Yes	Validation: Non updateable, but if a primary key is not passed in, you can pass in any ORIG_SYSTEM and ORIG_SYSTEM_REFERENCE that exists in the HZ_ORIG_SYS_REFERENCE table and unique validation is bypassed. The ORIG_SYSTEM_REFERENCE does not change, but persists in the table. Default: cust_account_role_id
orig_system	IN	VARCHAR2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_ B.orig_system. Default: UNKNOWN if an orig_system_reference is passed in.
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.

If the primary key is not passed in, get the primary key from the HZ_ORIG_SYS_ REFERENCES table based on orig_system and orig_system_reference if they are not null and unique.

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_R
EC_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
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	

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	
orig_system_ reference	IN	VARCHAR2	Yes	Validation: Not updateable
orig_system	IN	VARCHAR2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system.
attribute25	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: • Status cannot be set to null during update • It is a lookup code in lookup type REGISTRY_ STATUS
created_by_module	IN	VARCHAR2	Yes	Validation: Cannot be updated if value exists

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against the value in the database for the existing record. Comment: • Pass current object_version_number of the record from hz_custaccount_roles • Returns new value after update

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: • Mandatory field • Foreign key to hz_cust_account_roles
responsibility_type	IN	VARCHAR2	Yes	Validation: • Mandatory field • It is a lookup code in lookup type SITE_USE_CODE
primary_flag	IN	VARCHAR2	No	Validation: • Primary Flag is lookup code in lookup type YES/NO. • It is unique per cust_account_role_id 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	
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

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_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_RESPONSIBI
LITY_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: • Primary Flag is lookup code in lookup type YES/NO • It is unique per cust_account_role_id
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: • Mandatory attribute • Validated against value in the database for the existing record Comment: • Pass current object_version_number of the record from hz_role_responsibility[2] • Return new value after update

Other Validations

The combination of cust_account_role_id and responsibility_type should be unique.

13

Source System Management API Use

This chapter describes Source System Management APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Source System Managment APIs

Source System Managment APIs

PL/SQL Package Name: HZ_ORIG_SYSTEM_REF_PUB

Java Class Name: HzOrigSystemRefPub

PL/SQL Record Structure for Source System Reference

```
TYPE  ORIG_SYS_REFERENCE_REC_TYPE  IS RECORD (
    ORIG_SYSTEM_REF_ID              NUMBER(15),
    ORIG_SYSTEM                     VARCHAR2(30),
    ORIG_SYSTEM_REFERENCE           VARCHAR2(255),
    OWNER_TABLE_NAME                VARCHAR2(30),
    OWNER_TABLE_ID                  NUMBER(15),
    STATUS                          VARCHAR(1),
    START_DATE_ACTIVE               DATE,
    END_DATE_ACTIVE                 DATE,
    REASON_CODE                     VARCHAR2(30),
    OLD_ORIG_SYSTEM_REFERNCE        VARCHAR2(255),
    CREATED_BY_MODULE               VARCHAR2(150),
    APPLICATION_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)
);
```

Java Inner Class for Source System Reference

```
public static class OrigSysReferenceRec {
    public BigDecimal orig_system_ref_id;
    public String orig_system;
    public String orig_system_reference;
    public String owner_table_name;
    public BigDecimal owner_table_id;
    public String status;
    public String reason_code;
    public String old_orig_system_reference;
    public java.sql.Timestamp start_date_active;
    public java.sql.Timestamp end_date_active;
    public String created_by_module;
    public BigDecimal application_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 attribute16;
    public String attribute17;
    public String attribute18;
    public String attribute19;
    public String attribute20;

    public OrigSysReferenceRec()
    public OrigSysReferenceRec(boolean __RosettaUseGMISSValues);
```

Create Source System Reference API

Description

You can use this routine to create a mapping between a source system reference and a TCA owner_table_id.

PL/SQL Procedure

```
PROCEDURE create_orig_system_reference(
p_init_msg_list              IN          VARCHAR2 := FND_API.G_FAL
SE,
p_orig_sys_reference_rec     IN                ORIG_SYS_REFERENCE_
REC_TYPE,
x_return_status              OUT    NOCOPY     VARCHAR2,
x_msg_count                  OUT    NOCOPY     NUMBER,
x_msg_data                   OUT    NOCOPY     VARCHAR2,
x_orig_system_ref_id         OUT    NOCOPY     NUMBER);
);
```

Java Method

```
public static void createOrigSystemReference(
    OracleConnection _connection,
    String p_init_msg_list,
    OrigSysReferenceRec p_orig_sys_reference_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 Source System Reference 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
orig_system_ref_id	IN	NUMBER	No	Generated from sequence
orig_system	IN	VARCHAR2	Yes	Validations: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system where status is active. If multiple_flag = N, then the combination of orig_system, orig_system_reference, owner_table_ name, and status=A must be unique. If multiple_flag = Y, then the combination of orig_system, orig_system_reference, owner_table_ name, owner_table_id, and status=A must be unique.
orig_system_ reference	IN	VARCHAR2	Yes	Validations: If multiple_flag = N, then the combination of orig_system, orig_system_reference, owner_table_ name, and status=A must be unique. If multiple_flag = Y, then the combination of orig_system, orig_system_reference, owner_table_ name, owner_table_id, and status=A must be unique.

Parameter Name	Type	Data Type	Required	Validation Default Comment
owner_table_name	IN	VARCHAR2	Yes	Validations: Valid lookup code under TCA_OWNER_TABLE lookup type. If multiple_flag = N, then the combination of orig_system, orig_system_reference, and owner_table_name must be unique. If multiple_flag = Y, then the combination of orig_system, orig_system_reference, owner_table_name, owner_table_id, and status=A must be unique.
owner_table_id	IN	NUMBER	Yes	Validation: The combination of owner_table_name and owner_table_id must be valid.
status	IN	VARCHAR	No	Validation: Validated against the MOSR_STATUS lookup type. Default: A Sync up status and start/end_date_active.
start_date_active	IN	DATE	No	Validation: Start date cannot be a future date.
end_date_active	IN	DATE	No	Validation: End date cannot be a past date.
reason_code	IN	VARCHAR2	Yes	Validation: Validated against the MOSR_REASON lookup type.
old_orig_system_refernce	IN	VARCHAR2	Yes	Validation: The combination of orig_system, old_orig_system_reference, and owner_table_name must exist.
created_by _module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text to indicate module that initiates creation of record.
application_id	IN	NUMBER	No	Comment : Text to indicate application that initiates creation of record.
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	

Parameter Name	Type	Data Type	Required Validation Default Comment
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

Update Source System Reference API

Description

This routine is used to re-map or update source system references.

PL/SQL Procedure

```
PROCEDURE update_orig_system_reference(
   p_init_msg_list                 IN               VARCHAR2 := FN
D_API.G_FALSE,
   p_orig_sys_reference_rec        IN               ORIG_SYS_REFER
ENCE_REC_TYPE,
   p_object_version_number         IN OUT  NOCOPY   NUMBER,
   x_return_status                 OUT     NOCOPY   VARCHAR2,
   x_msg_count                     OUT     NOCOPY   NUMBER,
   x_msg_data                      OUT     NOCOPY   VARCHAR2
);
```

Java Method

```
public static void updateOrigSystemReference(
    OracleConnection _connection,
    String p_init_msg_list,
    OrigSysReferenceRec p_orig_sys_reference_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 Source System Reference 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
orig_system_ref_id	IN	NUMBER	No	Comment: This ID is based on the combination of orig_system, and owner_table_name.
orig_system	IN	VARCHAR2	Yes	Validations: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system where status is active. If the old_orig_system_ref is not passed in the combination of orig_system, old_orig_system,_ reference, and owner_table_name must exist.
orgi_system_ reference	IN	VARCHAR2	Yes	Validations: The combination of orig_system, orig_system_ reference, and owner_table_name must exist in the HZ_ORIG_SYSTEM_REFERENCE table. If the old_orig_system_ref is not passed in the combination of orig_system, old_orig_system_ reference, and owner_table_name must exist.
owner_table_name	IN	VARCHAR2	Yes	Validations: Valid lookup code under TCA_OWNER_TABLE lookup type. If the old_orig_system_ref is not passed in the combination of orig_system, old_orig_system,_ reference, and owner_table_name must exist.
owner_table_id	IN	NUMBER	Yes	Validation: The combination of owner_table_name and owner_table_id must be valid.
status	IN	VARCHAR	No	Validation: Validated against the MOSR_STATUS lookup type. Sync up status and start/end_date_active
start_date_active	IN	DATE	No	Validation : Start date must not be future date.
end_date_active	IN	DATE	No	Validation : End date cannot be a past date.
reason_code	IN	VARCHAR2	Yes	Validation : Validated against the MOSR_STATUS lookup type.
old_orig_system_ refernce	IN	VARCHAR2	Yes	Validation : The combination of orig_system, old_ orig_system_reference, and owner_table_name must exist.
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.
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			
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			

Remap Internal Identifier API

Description

This routine is used to re-map owner_table_id from existing owner table id to new owner table id for any system or certain system and to inactivate existing mapping with reason code.

PL/SQL Procedure

```
PROCEDURE  remap_internal_identifier(
    p_init_msg_list         IN                  VARCHAR2 := FND_AP
I.G_FALSE,
    p_old_owner_table_id    IN                  NUMBER,
    p_new_owner_table_id    IN                  NUMBER,
    p_owner_table_name      IN                  VARCHAR2,
    p_orig_system           IN                  VARCHAR2,
    p_orig_system_reference IN                  VARCHAR2,
    p_reason_code           IN                  VARCHAR2,
    x_return_status         OUT     NOCOPY      VARCHAR2,
    x_msg_count             OUT     NOCOPY      NUMBER,
    x_msg_data              OUT     NOCOPY      VARCHAR2
);
```

Java Method

```
public static void remapInternalIdentifier(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_old_owner_table_id,
    BigDecimal p_new_owner_table_id,
    String p_owner_table_name,
    String p_orig_system,
    String p_orig_system_reference,
    String p_reason_code,
    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 Remap Internal Identifier 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 Names	Type	Data Type	Required	Validation Default Comment
old_owner_table_id	IN	NUMBER	Yes	Validation : The combination of owner_table_name and owner_table_id must be valid.
new_owner_table_id	IN	NUMBER	Yes	Validation : The combination of owner_table_name and owner_table_id must be valid.
owner_table_name	IN	VARCHAR2	Yes	Validation : Validated against the OWNER_TABLE_ NAME lookup type.
orig_system	IN	VARCHAR2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system where status is active. If orig_system and orig_system_reference are passed in, then the combination of orig_system, orig_ system_reference, and owner_table_name must exist in the HZ_ORIG_SYSTEM_REFERENCE table.
orig_system_ reference	IN	VARCHAR2	No	Validation: If orig_system and orig_system_reference are passed in, then the combination of orig_system, orig_ system_reference, and owner_table_name must exist in the HZ_ORIG_SYSTEM_REFERENCE table.
reason_code	IN	VARCHAR2	Yes	Validation : Validated against the MOSR_STATUS lookup type.

14

Bulk Import API Use

This chapter describes Bulk Import APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Bulk Import APIs

Bulk Import APIs

PL/SQL Package Name: HZ_IMP_BATCH_SUMMARY_V2PUB

Description

The Bulk Import APIs are needed for using the feature Bulk Import. The two APIs in the package allow users to create an import batch and activate an import batch for processing in Bulk Import.

Create Import Batch API

Description

Use this API to create an import batch. Before processing any data through the Bulk Import process, you must call this API to create a batch and get a batch id. The API creates a record in the HZ_IMP_BATCH_SUMMARY table. The API returns a batch id to use for populating data for that batch in all the TCA interface tables.

PL/SQL Procedure

```
PROCEDURE create_import_batch (
      p_init_msg_list                    IN             VARCHAR2
:= FND_API.G_FALSE,
      p_batch_name                       IN             VARCHAR2,
      p_description                      IN             VARCHAR2,
      p_original_system                  IN             VARCHAR2,
      p_load_type                        IN             VARCHAR2
:= NULL,
      p_est_no_of_records                IN             NUMBER
:= NULL,
      x_batch_id                         OUT NOCOPY     NUMBER,
      x_return_status                    OUT NOCOPY     VARCHAR2,
      x_msg_count                        OUT NOCOPY     NUMBER,
      x_msg_data                         OUT NOCOPY     VARCHAR2
  )
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Import Batch 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	Default : FND_API.G_FALSE
p_batch_name	IN	VARCHAR2	Yes	
p_description	IN	VARCHAR2	No	
p_original_system	IN	VARCHAR2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
p_load_type	IN	VARCHAR2	No	
p_est_no_of_records	IN	NUMBER	No	
p_no_of_records	IN	NUMBER	No	Comment: Number of records to be returned. Maximum is 100. Default: 100
x_batch_id	OUT	NUMBER	Yes	Comment: Returns the system generated batch id
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 the x_msg_count = 1

Activate Import Batch API

Description

Use this API to activate an import batch. Before requesting processing of a batch by Bulk Import concurrent program, you must call this API to activate a batch for processing. The API updates the batch status column to ACTIVE for the batch record

in the HZ_IMP_BATCH_SUMMARY table. After you complete loading data in TCA interface tables and are ready to process the data, call this API to activate the batch and make it available in the Batch id list of values for the Bulk Import concurrent program.

PL/SQL Procedure

```
PROCEDURE activate_batch (
      p_init_msg_list                    IN             VARCHAR2:=
 FND_API.G_FALSE,
      p_batch_id                         IN             NUMBER,
      x_return_status                    OUT NOCOPY     VARCHAR2,
      x_msg_count                        OUT NOCOPY     NUMBER,
      x_msg_data                         OUT NOCOPY     VARCHAR2
  )
```

Parameter Description and Validation

The following table lists information about the parameters in the Activate Import Batch 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_batch_id	IN	NUMBER	Yes	Comment : Batch id of the batch you want to activate
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

15

Address and Tax Validation API Use

This chapter describes Location Service, Adapter, and Tax Assignment APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Location Service APIs
- Adapter APIs
- Tax Assignment APIs

Location Service APIs

PL/SQL Package Name: HZ_LOCATION_SERVICES_PUB

Address Validation API

Description

This API sends XML document to vendor adapter to validate and receives validated address in XML format. The API depends on adapter_id or country code to call address validation against different adapter.

PL/SQL Procedure

```
PROCEDURE submit_addrval_doc(
    p_addrval_doc                IN OUT NOCOPY NCLOB,
    p_adapter_id                 IN NUMBER DEFAULT NULL,
    p_country_code               IN VARCHAR2 DEFAULT NULL,
    p_module                     IN VARCHAR2 DEFAULT NULL,
    p_module_id                  IN NUMBER DEFAULT NULL,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_msg_count                  OUT NOCOPY NUMBER,
    x_msg_data                   OUT NOCOPY VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Address Validation 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_addrval_doc	IN OUT	NCLOB	Yes	Comment: XML document that contains locations for address Validation: API returns validated locations in XML format.
p_adapter_id	IN	NUMBER	No	Validation: Validate adapter_id in the HZ_ADAPTERS table Comment: Identifier of an adapter used to validate locations in p_addrval_doc.
p_country_code	IN	VARCHAR2	No	Comment : Country code used to find adapter information for address validation, if p_adapter_id is not passed.
p_module	IN	VARCHAR2	No	Comment : Name of the module calling address validation. The API does not validate this parameter. This parameter is used for record keeping only.
p_module_id	IN	NUMBER	No	Comment : Identifier of the module calling address validation. The API does not validate this parameter. This parameter is used for record keeping only.
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

Adapter APIs

PL/SQL Package Name: HZ_ADAPTER_PUB

PL/SQL Record Structure for Adapter

```
TYPE adapter_rec_type IS RECORD(
    adapter_id                   NUMBER,
    adapter_content_source       VARCHAR2(30),
    adapter_meaning              VARCHAR2(80),
    adapter_description          VARCHAR2(240)   DEFAULT NULL,
    message_format_code          VARCHAR2(30),
    synchronous_flag             VARCHAR2(1)     DEFAULT 'Y',
    invoke_method_code           VARCHAR2(30),
    host_address                 VARCHAR2(240),
    enabled_flag                 VARCHAR2(1)     DEFAULT 'Y',
    maximum_batch_size           NUMBER,
    default_batch_size           NUMBER,
    default_replace_status_level VARCHAR2(30),
    username                     VARCHAR2(100),
    encrypted_password           VARCHAR2(100)
)
TYPE adapter_terr_rec_type IS RECORD(

    adapter_id                  NUMBER,
    territory_code              VARCHAR2(30),
    enabled_flag                VARCHAR2(1)      DEFAULT 'Y',
    default_flag                VARCHAR2(1),
)
```

Create Adapter API

Description

This API is used to create adapter. The record holds information about an adapter, such as name, meaning, description, host address, xml format and communication protocol. When creating an adapter, a new lookup code, meaning and description will be added under lookup type CONTENT_SOURCE_TYPE.

PL/SQL Procedure

```
   PROCEDURE create_adapter (
    p_adapter_rec                IN ADAPTER_REC_TYPE,
    x_adapter_id                 OUT NOCOPY NUMBER,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_msg_count                  OUT NOCOPY NUMBER,
    x_msg_data                   OUT NOCOPY VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Adapter 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
adapter_id	IN	NUMBER	No	Validation : Unique if passed in, else generated from sequence
adapter_content_source	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against CONTENT_SOURCE_TYPE lookup type
adapter_meaning	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against CONTENT_SOURCE_TYPE lookup type for duplicate meaning
adapter_description	IN	VARCHAR2	No	
message_format_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against HZ_MESSAGE_FORMAT lookup type
synchronous_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute; Y or N Default: Y
invoke_method_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against HZ_INVOKE_METHOD lookup type
host_address	IN	VARCHAR2	No	
enabled_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute; Y or N Default: Y
maximum_batch_size	IN	NUMBER	Yes	Validation : Mandatory attribute that must be greater than 0
default_batch_size	IN	NUMBER	Yes	Validation : Mandatory attribute that must be greater than 0
default_replace_status_code	IN	VARCHAR2	No	
username	IN	VARCHAR2	No	
encrypted_password	IN	VARCHAR2	No	
x_adapter_id	OUT	NUMBER	Yes	Comment : Return adapter_id of the record created
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

Other Validations

There can be only one record for a given adapter_content_source. There can be only one record for a given adapter_meaning.

Update Adapter API

Description

This API is used to update adapter. It updates a record in the HZ_ADAPTERS table. If meaning and description are changed, the API will update corresponding lookup code under lookup type CONTENT_SOURCE_TYPE.

PL/SQL Procedure

```
   PROCEDURE update_adapter (
    p_adapter_rec                 IN ADAPTER_REC_TYPE,
    px_object_version_number    IN OUT NOCOPY NUMBER,
    x_return_status               OUT NOCOPY VARCHAR2,
    x_msg_count                   OUT NOCOPY NUMBER,
    x_msg_data                    OUT NOCOPY VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Adapter 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
adapter_id	IN	NUMBER	Yes	Validation : Valid adapter_id should be passed in Comment: Pass the adapter_id from HZ_ADAPTERS table
adapter_content_source	IN	VARCHAR2	Yes	Validation : Mandatory attribute, Validated against CONTENT_SOURCE_TYPE lookup type
adapter_meaning	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against CONTENT_SOURCE_TYPE lookup type for duplicate meaning
adapter_description	IN	VARCHAR2	No	
message_format_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against HZ_MESSAGE_FORMAT lookup type
synchronous_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute; Y or N Default: Y
invoke_method_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute validated against HZ_INVOKE_METHOD lookup type
host_address	IN	VARCHAR2	No	
enabled_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute, Y or N Default: Y
maximum_batch_size	IN	NUMBER	Yes	Validation : Mandatory attribute, Must be greater than 0
default_batch_size	IN	NUMBER	Yes	Validation : Mandatory attribute, Must be greater than 0
default_replace_status_code	IN	VARCHAR2	No	
username	IN	VARCHAR2	No	
encrypted_password	IN	VARCHAR2	No	
px_object_version_number	IN OUT	NUMBER	Yes	Validation: Mandatory attribute validated against value in the database for the existing adapter record
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

Create Adapter Territory API

Description

You can use this API to assign a territory to an adapter. This API creates a record in the HZ_ADAPTER_TERRITORIES table. The adapter must be created before you can assign its territory. You can create multiple territory records for an adapter with different territory codes. The API can specify the default adapter for a territory.

PL/SQL Procedure

```
PROCEDURE create_adapter_terr (
    p_adapter_terr_rec              IN ADAPTER_TERR_REC_TYPE,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_msg_count                  OUT NOCOPY NUMBER,
    x_msg_data                   OUT NOCOPY VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Create Adapter Territory 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
adapter_id	IN	NUMBER	Yes	Validation : Unique if passed in, otherwise generated from sequence
territory_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute, Validated against FND_TERRITORIES table
enabled_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute; Y or N Default: Y
default_flag	IN	VARCHAR2	No	Validation : Y or 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

Other Validations

There can be only one record for a given adapter_id, territory_code.

Update Adapter Territory API

Description

This API is used to update territory for an adapter. This API updates a record in HZ_ADAPTER_TERRITORIES table. You cannot update the territory code but you can change the default and enabled flag of the territory record.

PL/SQL Procedure

```
PROCEDURE create_adapter_terr (
    p_adapter_terr_rec        IN ADAPTER_TERR_REC_TYPE,
    px_object_version_number  IN OUT   NOCOPY    NUMBER,
    x_return_status           OUT      NOCOPY    VARCHAR2,
    x_msg_count               OUT      NOCOPY    NUMBER,
    x_msg_data                OUT      NOCOPY    VARCHAR2
)
```

Parameter Description and Validation

The following table lists information about the parameters in the Update Adapter 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
adapter_id	IN	NUMBER	Yes	Validation : Unique if passed in, otherwise generated from sequence
territory_code	IN	VARCHAR2	Yes	Validation : Mandatory attribute, Validated against FND_TERRITORIES table
enabled_flag	IN	VARCHAR2	Yes	Validation : Mandatory attribute; Y or N Default: Y
default_flag	IN	VARCHAR2	No	Validation : Y or 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

Other Validations

You cannot change enabled_flag to 'N' if the default flag of the record is 'Y'.

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 or modified. It also creates or updates a record in the HZ_LOCATIONS_PROFILES table. 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                                  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;
```

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: • Mandatory attribute • valid location_id from the HZ_LOCATIONS table should be passed in.
p_lock_flag	IN	NUMBER	No	
p_created_by_ module	IN	VARCHAR2	Yes	Validation: Mandatory attribute Comment: Text indicating which module initiated the creation of a record.
p_application_id	IN	NUMBER	No	Comment: Text indicating which module initiated the creation of a record.
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 or modified. It also creates or updates a record in the HZ_LOCATIONS_PROFILES table. 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: • Mandatory attribute • Valid location_id from the HZ_LOCATIONS table should be passed in.
p_lock_flag	IN	NUMBER	No	
p_created_by_ module	IN	VARCHAR2	No	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

16

Data Quality Management API Use

This chapter describes Data Quality Management Search and Duplicate Identification, Availability, and Transformation APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Data Quality Management Search and Duplicate Identification APIs
- Data Quality Management Availability APIs
- Data Quality Management Transformation APIs

Data Quality Management Search and Duplicate Identification APIs

PL/SQL Package Name: HZ_PARTY_SEARCH

> **Note:** The code for all the Data Quality Management (DQM) APIs, except the transformation and availability APIs, is dynamically generated based on how a match rule is set up. The code for each API is in the appropriate match rule package, and the HZ_PARTY_SEARCH package is a wrapper on top of all match rule packages, to facilitate the use of these APIs with a nomenclature that is not cumbersome for the user.

Java Class Name: HzPartySearch

Find Parties API

Description

Finds parties based on the passed search criteria. The API finds parties that match party level search criteria, and/or have addresses, contacts, and/or contact points that match corresponding address, contact, or contact point criteria. When the matching is based on address and contact point search criteria, the API finds parties of type Organization, looking at the organization end of relationships. The API returns the set of matches to the HZ_MATCHED_PARTIES_GT table, which holds the PARTY_ID and score of all matches. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
find_parties (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_search_rec      IN      party_search_rec_type,
     p_party_site_list       IN      party_site_list,
     p_contact_list          IN      contact_list,
     p_contact_point_list    IN      contact_point_list,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     p_search_merged         IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findParties(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    PartySearchRec p_party_search_rec,
    PartySiteSearchRec []p_party_site_list,
    ContactSearchRec []p_contact_list,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    String p_search_merged,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Parties 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_search_rec	IN	party_ search_rec_ type	No	Comments: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	party_site_ list	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
p_search_merged	IN	VARCHAR2	Yes	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTIES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of parties that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.

- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Find Persons API

Description

Finds persons based on the passed search criteria. The API finds persons that match party level search criteria, and/or have addresses, contacts, and/or contact points that match corresponding address, contact, or contact point criteria. The API always returns parties of type Person, even if the matching is based on address or contact point search criteria, by looking at the person end of relationships. The API returns the set of matches to the HZ_MATCHED_PARTIES_GT table, which holds the PARTY_ID and score of all matches. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
find_persons (
      p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
      p_rule_id               IN      NUMBER,
      p_party_search_rec      IN      party_search_rec_type,
      p_party_site_list       IN      party_site_list,
      p_contact_list          IN      contact_list,
      p_contact_point_list    IN      contact_point_list,
      p_restrict_sql          IN      VARCHAR2,
      p_match_type            IN      VARCHAR2,
      x_search_ctx_id         OUT     NUMBER,
      x_num_matches           OUT     NUMBER,
      x_return_status         OUT     VARCHAR2,
      x_msg_count             OUT     NUMBER,
      x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findPersons(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    PartySearchRec p_party_search_rec,
    PartySiteSearchRec []p_party_site_list,
    ContactSearchRec []p_contact_list,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Persons 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_search_rec	IN	party_ search_rec_ type	No	Comments: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	party_site_ list	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
p_search_merged	IN	VARCHAR2	Yes	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTIES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of parties that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.

- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Find Parties and Details API

Description

Identifies duplicates of a specific party. The API finds duplicates within a subset or across the entire TCA Registry, depending on what is passed into the p_restrict_sql parameter. The API inserts duplicates into the HZ_MATCHED_PARTIES_GT table if the p_dup_batch_id parameter is null. If this parameter is not null, then the API creates a duplicate set with the list of duplicates in the HZ_DUP_SET and HZ_DUP_SET_PARTIES tables. Use the x_search_ctx_id value that the API returns to filter results from the HZ_MATCHED_PARTIES_GT table.

PL/SQL Procedure

```
PROCEDURE
find_party_details (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_search_rec      IN      party_search_rec_type,
     p_party_site_list       IN      party_site_list,
     p_contact_list          IN      contact_list,
     p_contact_point_list    IN      contact_point_list,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     p_search_merged         IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findPartyDetails(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    PartySearchRec p_party_search_rec,
    PartySiteSearchRec []p_party_site_list,
    ContactSearchRec []p_contact_list,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    String p_search_merged,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Parties and Details 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_search_rec	IN	party_ search_rec_ type	No	Comments: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	party_site_ list	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
p_search_merged	IN	VARCHAR2	Yes	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTIES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of parties that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.

- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Find Party Sites API

Description

Finds party sites based on the passed search criteria. The API finds all party sites that match the address search criteria passed into the p_party_site_list parameter, and/or have contact points, defined for party sites, that match contact point criteria passed into the p_contact_point_list parameter. The API returns the set of matches to the HZ_MATCHED_PARTY_SITES_GT table, which holds the PARTY_SITE_ID, PARTY_ID, and score of all matches. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
get_matching_party_sites (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_party_site_list       IN      PARTY_SITE_LIST,
     p_contact_point_list    IN      CONTACT_POINT_LIST,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
getMatchingPartySites(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_party_id,
    PartySiteSearchRec []p_party_site_list,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Party Sites 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_id	IN	NUMBER	No	Comments: Optional parameter if the search for party sites is only to be performed within one party.
p_party_site_list	IN	party_site_ list	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTY_SITES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of party sites that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Find Contacts API

Description

Finds contacts based on the passed search criteria. The API finds all contacts that match the contact search criteria passed into the p_contact_list parameter, and/or have contact points,defined for contacts, that match contact point criteria passed into the p_contact_point_list parameter.The API returns the set of matches to the HZ_MATCHED_CONTACTS_GT table, which holds the ORG_CONTACT_ID, PARTY_ID, and score of all matches. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
get_matching_contacts (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_contact_list          IN      CONTACT_LIST,
     p_contact_point_list    IN      CONTACT_POINT_LIST,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
getMatchingContacts(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_party_id,
    ContactSearchRec []p_contact_list,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Contacts 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_id	IN	NUMBER	No	Comments: Optional parameter if the search for contacts is only to be performed within one party.
p_contact_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_CONTACTS_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of contacts that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Find Contact Points API

Description

Finds contact points based on the passed search criteria. The API finds all contact points that match the contact point search criteria passed into the p_contact_point_list parameter. The API returns the set of matches to the HZ_MATCHED_CPTS_GT table, which holds the CONTACT_POINT_ID,PARTY_ID, and score of all matches. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
get_matching_contact_points (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_contact_point_list    IN      CONTACT_POINT_LIST,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
getMatchingContactPoints(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_party_id,
    ContactPointSearchRec []p_contact_point_list,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Find Contact Points 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_id	IN	NUMBER	No	Comments: Optional parameter if the search for contacts is only to be performed within one party.
p_contact_point_list	IN	contact_ point_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_CPTS_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of contact points that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Identify Duplicate Parties API

Description

Identifies duplicates of a specific party. The API finds duplicates within a subset or across the entire TCA Registry, depending on what is passed into the p_restrict_sql parameter. The API inserts duplicates into the HZ_MATCHED_PARTIES_GT table if the p_dup_batch_id parameter is null. If this parameter is not null, then the API creates a duplicate set with the list of duplicates in the HZ_DUP_SET and HZ_DUP_SET_PARTIES tables. Use the x_search_ctx_id value that the API returns to filter results from the HZ_MATCHED_PARTIES_GT table.

PL/SQL Procedure

```
PROCEDURE
find_duplicate_parties (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     p_dup_batch_id          IN      NUMBER,
     p_search_merged         IN      VARCHAR2,
     x_dup_set_id            OUT     NUMBER,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findDuplicateParties(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_party_id,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal p_dup_batch_id,
    String p_search_merged,
    BigDecimal  [] x_dup_set_id,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Identify Duplicate Parties 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_id	IN	NUMBER	Yes	Comments: ID of a party whose duplicates are to be found.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
p_dup_batch_id	IN	NUMBER	No	Comment: Specifies a duplicate batch in which a duplicate set is to be created. If this parameter is null, then the duplicates found by the APIS are inserted into HZ_MATCHED_PARTIES_GT. If not, then the duplicates are inserted into HZ_DUP_SETS with the dup_batch_id specified.
p_search_merged	IN	VARCHAR2	No	Comment: Specifies if parties that have been merged should be returned as matches.
x_dup_set_id	OUT	NUMBER	No	Comment: The ID of a duplicate set that was created in the HZ_DUP_SETS table.
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTIES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of parties that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.

- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Identify Duplicate Party Sites API

Description

Identifies duplicates of a specific party site. The API finds duplicates within a subset defined by what is passed into the p_restrict_sql parameter, within the party passed into the p_party_id parameter, or across the entire TCA Registry. The API inserts duplicates into the HZ_MATCHED_PARTY_SITES_GT table. Use the x_search_ctx_id value that the API returns to filter results from this table.c

PL/SQL Procedure

```
PROCEDURE
find_duplicate_party_sites (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_site_id         IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findDuplicatePartySites(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_party_site_id,
    BigDecimal p_party_id,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Identify Duplicate Party Sites 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_site_id	IN	NUMBER	Yes	Comment: ID of the party site whose duplicates are to be found.
p_party_id	IN	NUMBER	No	Comment: Optional parameter which specifies a particular party_id whose party sites to find the duplicates in.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_PARTY_SITES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of party sites that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Identify Duplicate Contacts API

Description

Identifies duplicates of a specific contact. The API finds duplicates within a subset defined by what is passed into the p_restrict_sql parameter, within the party passed into the p_party_id parameter, or across the entire TCA Registry. The API inserts duplicates into the HZ_MATCHED_CONTACTS_GT table. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
find_duplicate_contacts (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_org_contact_id        IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_restrict_sql          IN      VARCHAR2,
     p_match_type            IN      VARCHAR2,
     x_search_ctx_id         OUT     NUMBER,
     x_num_matches           OUT     NUMBER,
     x_return_status         OUT     VARCHAR2,
     x_msg_count             OUT     NUMBER,
     x_msg_data              OUT     VARCHAR2
);
```

Java Method

```
findDuplicateContacts(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_org_contact_id,
    BigDecimal p_party_id,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Identify Duplicate Contacts 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_org_contact_id	IN	NUMBER	Yes	Comment: ID of the contact whose duplicates are to be found.
p_party_id	IN	NUMBER	No	Comment: Optional parameter which specifies a particular party_id whose contacts to find the duplicates in.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_CONTACTS_SITES_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of contacts that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Identify Duplicate Contact Points API

Description

Identifies duplicates of a specific contact point. The API finds duplicates within a subset defined by what is passed into the p_restrict_sql parameter, within the party passed into the p_party_id parameter, or across the entire TCA Registry. The API inserts duplicates into the HZ_MATCHED_CPTS_GT table. Use the x_search_ctx_id value that the API returns to filter results from this table.

PL/SQL Procedure

```
PROCEDURE
 find_duplicate_contact_points (
        p_init_msg_list       IN      VARCHAR2:= FND_API.G_FALSE,
        p_rule_id             IN      NUMBER,
        p_contact_point_id    IN      NUMBER,
        p_party_id            IN      NUMBER,
        p_restrict_sql        IN      VARCHAR2,
        p_match_type          IN      VARCHAR2,
        x_search_ctx_id       OUT     NUMBER,
        x_num_matches         OUT     NUMBER,
        x_return_status       OUT     VARCHAR2,
        x_msg_count           OUT     NUMBER,
        x_msg_data            OUT     VARCHAR2
)
```

Java Method

```
findDuplicateContactPoints(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_contact_point_id,
    BigDecimal p_party_id,
    String p_restrict_sql,
    String p_match_type,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data
  )
```

Parameter Description and Validation

The following table lists information about the parameters in the Identify Duplicate Contact Points 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_contact_point_id	IN	NUMBER	Yes	Comment: ID of the contact point whose duplicates are to be found.
p_party_id	IN	NUMBER	No	Comment: ID of the party whose contact points to find the duplicates in.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule).
p_search_merged	IN	NUMBER	No	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_SPTS_GT to query the results.
x_num_matches	OUT	NUMBER	No	Comment: Number of contact points that matched.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Get Score Details API

Description

Gets details about how a party matches the input search criteria. Though not required, the API is usually called after calls to the find_parties API, to display how a match is determined. The API compares the input search criteria against the party passed into the p_party_id paramter, and inserts all matching attributes into the HZ_PARTY_SCORE_DTLS_GT table. The columns in this table include: ATTRIBUTE, the matching attribute; ENTERED_VALUE, the attribute value entered for the search criterion; MATCHED_VALUE, the attribute value for the p_party_id party, and ASSIGNED_SCORE, the score assigned to the match.

The x_search_ctx_id is used as an IN/OUT parameter. If this API is called right after a call to find_parties, then this API can use the same search_context_id and would retain x_search_context_id as is. If the search_context_id is not passed in, then this API generates and populates a search_context_id in the x_search_context_id variable. In either case, use the x_search_context_id value that the API returns to filter results from the HZ_PARTY_SCORE_DTLS_GT table.

PL/SQL Procedure

```
get_score_details (
     p_init_msg_list         IN      VARCHAR2:= FND_API.G_FALSE,
     p_rule_id               IN      NUMBER,
     p_party_id              IN      NUMBER,
     p_party_search_rec      IN      party_search_rec_type,
     p_party_site_list       IN      party_site_list,
     p_contact_list          IN      contact_list,
     p_contact_point_list    IN      contact_point_list,
     x_search_ctx_id         IN OUT  NUMBER,
     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 Get Score Details 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	OUT	VARCHAR2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_party_id	IN	NUMBER	Yes	Comment: ID of the party against which the input search criteria is to be evaluated.
p_party_search_rec	IN	NUMBER	No	Comment: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	NUMBER	No	Comment: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	NUMBER	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	NUMBER	No	Comment: The PL/SQL table of records structure that has the contact point search record information
x_search_ctxt_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_PARTY_SCORE_DTLS_GT to query the results. If the ID is passed in, it is used. If not passed, it is generated.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Call API Dynamic API

Description

Calls the appropriate API based on attribute ID values. The API accepts up to 20 attribute ID value pairs as search criteria and dispatches a call to the corresponding search API that is passed into the p_api_name parameter. Use the x_search_ctx_id value that the API returns to filter results from the appropriate table.

PL/SQL Procedure

```
PROCEDURE call_api_dynamic (
p_init_msg_list   IN  VARCHAR2:= FND_API.G_FALSE,
p_rule_id         IN  NUMBER,
p_attrib_id1      IN  NUMBER,
p_attrib_id2      IN  NUMBER,
p_attrib_id3      IN  NUMBER,
p_attrib_id4      IN  NUMBER,
p_attrib_id5      IN  NUMBER,
p_attrib_id6      IN  NUMBER,
p_attrib_id7      IN  NUMBER,
p_attrib_id8      IN  NUMBER,
p_attrib_id9      IN  NUMBER,
p_attrib_id10     IN  NUMBER,
p_attrib_id11     IN  NUMBER,
p_attrib_id12     IN  NUMBER,
p_attrib_id13     IN  NUMBER,
p_attrib_id14     IN  NUMBER,
p_attrib_id15     IN  NUMBER,
p_attrib_id16     IN  NUMBER,
p_attrib_id17     IN  NUMBER,
p_attrib_id18     IN  NUMBER,
p_attrib_id19     IN  NUMBER,
p_attrib_id20     IN  NUMBER,
p_attrib_val1     IN  VARCHAR2,
p_attrib_val2     IN  VARCHAR2,
p_attrib_val3     IN  VARCHAR2,
p_attrib_val4     IN  VARCHAR2,
p_attrib_val5     IN  VARCHAR2,
p_attrib_val6     IN  VARCHAR2,
p_attrib_val7     IN  VARCHAR2,
p_attrib_val8     IN  VARCHAR2,
p_attrib_val9     IN  VARCHAR2,
p_attrib_val10    IN  VARCHAR2,
p_attrib_val11    IN  VARCHAR2
p_attrib_val12    IN  VARCHAR2
p_attrib_val13    IN  VARCHAR2
p_attrib_val14    IN  VARCHAR2
p_attrib_val15    IN  VARCHAR2
p_attrib_val16    IN  VARCHAR2
p_attrib_val17    IN  VARCHAR2
p_attrib_val18    IN  VARCHAR2
p_attrib_val19    IN  VARCHAR2
p_attrib_val20    IN  VARCHAR2
p_restrict_sql    IN  VARCHAR2
        p_api_name          IN   VARCHAR2
        p_match_type        IN   VARCHAR2
        p_party_id          IN   NUMBER,
        p_search_merged     IN   VARCHAR2
        x_search_ctx_id     OUT  NUMBER,
        x_num_matches       OUT  NUMBER,
        x_return_status     OUT  VARCHAR2,
        x_msg_count         OUT  NUMBER,
        x_msg_data          OUT  VARCHAR2

)
```

Java Method

```
callApiDynamic(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    BigDecimal p_attrib_id1,
    BigDecimal p_attrib_id2,
    BigDecimal p_attrib_id3,
    BigDecimal p_attrib_id4,
    BigDecimal p_attrib_id5,
    BigDecimal p_attrib_id6,
    BigDecimal p_attrib_id7,
    BigDecimal p_attrib_id8,
    BigDecimal p_attrib_id9,
    BigDecimal p_attrib_id10,
    BigDecimal p_attrib_id11,
    BigDecimal p_attrib_id12,
    BigDecimal p_attrib_id13,
    BigDecimal p_attrib_id14,
    BigDecimal p_attrib_id15,
    BigDecimal p_attrib_id16,
    BigDecimal p_attrib_id17,
    BigDecimal p_attrib_id18,
    BigDecimal p_attrib_id19,
    BigDecimal p_attrib_id20,
    String p_attrib_val1,
    String p_attrib_val2,
    String p_attrib_val3,
    String p_attrib_val4,
    String p_attrib_val5,
    String p_attrib_val6,
    String p_attrib_val7,
    String p_attrib_val8,
    String p_attrib_val9,
    String p_attrib_val10,
    String p_attrib_val11,
    String p_attrib_val12,
    String p_attrib_val13,
    String p_attrib_val14,
    String p_attrib_val15,
    String p_attrib_val16,
    String p_attrib_val17,
    String p_attrib_val18,
    String p_attrib_val19,
    String p_attrib_val20,
    String p_restrict_sql,
    String p_api_name,
    String p_match_type,
    BigDecimal p_party_id,
    String p_search_merged,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    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 Call API Dynamic 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_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_attrib_id1 to p_attrib_id20	IN	NUMBER	No	Comment: Attribute IDs in the match rule for which search criteria is passed in.
p_attrib_val1 to p_attrib_val20	IN	NUMBER	No	Comment: Corresponding search criteria for the match rule attributes.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_api_name	IN	VARCHAR2	No	Comment: The name of the API to call. Validation: Value can be: FIND_PARTIES, FIND_PARTY_DETAILS, FIND_PERSONS, GET_MATCHING_PARTY_SITES, GET_MATCHING_CONTACTS, or GET_MATCHING_CONTACT_POINTS.
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule). **Note:** Thresholds are still applied to filter results. If p_match_type is AND, then it matches using the AND rule. If p_match_type is OR, then it matches using the OR rule. Default: As specified in the match rule.
p_party_id	IN	NUMBER	No	Comment: Only used for the get_matching APIs, if you want to restrict the search for details on a particular party.
p_search_merged	IN	VARCHAR2	No	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctx_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_MATCHED_XXX_GT to query the results, where XXX is PARTIES, PARTY_SITES, CONTACTS, or CPTS.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_num_matches	OUT	NUMBER	No	Comment: Number of matches.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Call API Dynamic Names API

Description

Calls the appropriate API based on attribute name values. The API accepts up to 20 attribute name value pairs as search criteria and dispatches a call to the corresponding search API that is passed into the p_api_name parameter. Use the x_search_ctx_id value that the API returns to filter results from the appropriate table.

PL/SQL Procedure

```
PROCEDURE  call_api_dynamic_names (
p_init_msg_list   IN  VARCHAR2:= FND_API.G_FALSE,
p_rule_id         IN  NUMBER,
p_attrib_name1      IN  NUMBER,
p_attrib_name2      IN  NUMBER,
p_attrib_name3      IN  NUMBER,
p_attrib_name4      IN  NUMBER,
p_attrib_name5      IN  NUMBER,
p_attrib_name6      IN  NUMBER,
p_attrib_name7      IN  NUMBER,
p_attrib_name8      IN  NUMBER,
p_attrib_name9      IN  NUMBER,
p_attrib_name10     IN  NUMBER,
p_attrib_name11     IN  NUMBER,
p_attrib_name12     IN  NUMBER,
p_attrib_name13     IN  NUMBER,
p_attrib_name14     IN  NUMBER,
p_attrib_name15     IN  NUMBER,
p_attrib_name16     IN  NUMBER,
p_attrib_name17     IN  NUMBER,
p_attrib_name18     IN  NUMBER,
p_attrib_name19     IN  NUMBER,
p_attrib_name20     IN  NUMBER,
p_attrib_val1     IN  VARCHAR2,
p_attrib_val2     IN  VARCHAR2,
p_attrib_val3     IN  VARCHAR2,
p_attrib_val4     IN  VARCHAR2,
p_attrib_val5     IN  VARCHAR2,
p_attrib_val6     IN  VARCHAR2,
p_attrib_val7     IN  VARCHAR2,
p_attrib_val8     IN  VARCHAR2,
p_attrib_val9     IN  VARCHAR2,
p_attrib_val10    IN  VARCHAR2,
p_attrib_val11    IN  VARCHAR2
p_attrib_val12    IN  VARCHAR2
p_attrib_val13    IN  VARCHAR2
p_attrib_val14    IN  VARCHAR2
p_attrib_val15    IN  VARCHAR2
p_attrib_val16    IN  VARCHAR2
p_attrib_val17    IN  VARCHAR2
p_attrib_val18    IN  VARCHAR2
p_attrib_val19    IN  VARCHAR2
p_attrib_val20    IN  VARCHAR2
p_restrict_sql    IN  VARCHAR2
        p_api_name          IN   VARCHAR2
        p_match_type        IN   VARCHAR2
        p_party_id          IN   NUMBER,
        p_search_merged     IN   VARCHAR2
        x_search_ctx_id     OUT  NUMBER,
        x_num_matches       OUT  NUMBER,
        x_return_status     OUT  VARCHAR2,
        x_msg_count         OUT  NUMBER,
        x_msg_data          OUT  VARCHAR2

)
```

Java Method

```
public static void callApiDynamicNames(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_rule_id,
    String p_attrib_name1,
    String p_attrib_name2,
    String p_attrib_name3,
    String p_attrib_name4,
    String p_attrib_name5,
    String p_attrib_name6,
    String p_attrib_name7,
    String p_attrib_name8,
    String p_attrib_name9,
    String p_attrib_name10,
    String p_attrib_name11,
    String p_attrib_name12,
    String p_attrib_name13,
    String p_attrib_name14,
    String p_attrib_name15,
    String p_attrib_name16,
    String p_attrib_name17,
    String p_attrib_name18,
    String p_attrib_name19,
    String p_attrib_name20,
    String p_attrib_val1,
    String p_attrib_val2,
    String p_attrib_val3,
    String p_attrib_val4,
    String p_attrib_val5,
    String p_attrib_val6,
    String p_attrib_val7,
    String p_attrib_val8,
    String p_attrib_val9,
    String p_attrib_val10,
    String p_attrib_val11,
    String p_attrib_val12,
    String p_attrib_val13,
    String p_attrib_val14,
    String p_attrib_val15,
    String p_attrib_val16,
    String p_attrib_val17,
    String p_attrib_val18,
    String p_attrib_val19,
    String p_attrib_val20,
    String p_restrict_sql,
    String p_api_name,
    String p_match_type,
    BigDecimal p_party_id,
    String p_search_merged,
    BigDecimal  [] x_search_ctx_id,
    BigDecimal  [] x_num_matches,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data
  )
```

Parameter Description and Validation

The following table lists information about the parameters in the Call API Dynamic Names 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_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.
p_attrib_name1 to p_attrib_name20	IN	NUMBER	No	Comment: Attribute names in the match rule for which search criteria is passed in.
p_attrib_val1 to p_attrib_val20	IN	NUMBER	No	Comment: Corresponding search criteria for the match rule attributes.
p_restrict_sql	IN	VARCHAR2	No	Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows. The SQL clause will be substituted into the following SELECT statement: SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'<intermedia query_string>') AND <p_restrict_sql>;
p_api_name	IN	VARCHAR2	No	Comment: The name of the API to call. Validation: Value can be: FIND_PARTIES, F IND_PARTY_DETAILS, FIND_PERSONS, GET_ MATCHING_PARTY_SITES, GET_MATCHING_ CONTACTS, or GET_MATCHING_CONTACT_PO INTS.
p_match_type	IN	VARCHAR2	No	Comment: Indicates if matches are returned only if, for all input attributes, it matches at least one transformation (AND rule), or any one of them for any transformation (OR rule). **Note:** Thresholds are still applied to filter results. If p_match_type is AND, then it matches using the AND rule. If p_match_type is OR, then it matches using the OR rule. Default: As specified in the match rule.
p_party_id	IN	NUMBER	No	Comment: Only used for the get_matching APIs, if you want to restrict the search for details on a particular party.
p_search_merged	IN	VARCHAR2	No	Comment: Specifies if parties that have been merged should be returned as matches.
x_search_ctx_id	OUT	NUMBER	No	Comment: An ID used to join back to HZ_ MATCHED_XXX_GT to query the results, where XXX is PARTIES, PARTY_SITES, CONTACTS, or CPTS.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_num_matches	OUT	NUMBER	No	Comment: Number of matches.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

Other Validations

- The corresponding match rule is compiled, before calling the above API from HZ_PARTY_SEARCH.
- All the attributes and transformations that are used as part of the match rule have been staged, in order for the above API to return results.

Data Quality Management Availability APIs

PL/SQL Package Name: HZ_DQM_SEARCH_UTILS

Is DQM Match Rule Available API

Description

Checks if a match rule is available, meaning that the rule is compiled and all its active transformations are staged. The function accordingly returns FND_API.G_TRUE or FND_API.G_FALSE.

PL/SQL Function

```
FUNCTION is_dqm_available (
  p_match_rule_id NUMBER)
 RETURN VARCHAR2
```

Parameter Description and Validation

The following table lists information about the parameters in the Is DQM Match Rule Available 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_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.

Is DQM Index Available API

Description

Checks if *inter*Media indexes in all Data Quality Management staging tables are created and valid. The function accordingly returns FND_API.G_TRUE or FND_API.G_FALSE.

PL/SQL Function

```
FUNCTION is_dqm_available
RETURN VARCHAR2
```

Data Quality Management Transformation APIs

PL/SQL Package Name: HZ_TRANS_PKG

Replace Word API

Description

Performs a word replacement in Data Quality Management. The function takes an input string, tokenizes it using spaces, replaces each token based on the passed word replacement list, and returns the concatenated replaced tokens. There are two versions of this API, one with the p_has_spc parameter, the other without.

PL/SQL Function

The p_has_spc parameter is only in one version of this API.

```
FUNCTION word_replace (
p_input_str IN VARCHAR2,
p_word_list_id IN NUMBER,
p_language IN VARCHAR2,
p_has_spc IN BOOLEAN DEFAULT FALSE)

RETURN VARCHAR2
```

Parameter Description and Validation

The following table lists information about the parameters in the Replace Word 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_input_str	IN	VARCHAR2	Yes	Comment: The original string that needs to be replaced.
p_word_list_id	IN	NUMBER	Yes	Comment: The word list ID of the word dictionary that forms the basis of the word replacement.
p_language	IN	VARCHAR2	No	Comment: Not currently used.
p_has_spc	IN	BOOLEAN	No	Comment: Not currently used. Only in one version of this API.

17

Extensions API Use

This chapter describes Extensions APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Extensions APIs

Extensions APIs

PL/SQL Package Name: HZ_EXTENSIBILITY_PUB

Create or Update Organization Profile Extension

Description

Use this routine to create or update information in the organization extensions tables. This API can be used to maintain records in the HZ_ORG_PROFILES_EXT_B and HZ_ORG_PROFILES_EXT_TL tables for a given organization. The extension tables hold additional information that you decide to capture about an organization.

PL/SQL Procedure

```
PROCEDURE Process_Organization_Record (
 p_api_version              IN NUMBER,
 p_org_profile_id           IN NUMBER,
 p_attributes_row_table     IN EGO_USER_ATTR_ROW_TABLE,
 p_attributes_data_table    IN EGO_USER_ATTR_DATA_TABLE,
 p_change_info_table        IN EGO_USER_ATTR_CHANGE_TABLE DEFAULT N
ULL,
 p_entity_id                IN NUMBER DEFAULT NULL,
 p_entity_index             IN NUMBER DEFAULT NULL,
 p_entity_code              IN VARCHAR2 DEFAULT NULL,
 p_debug_level              IN NUMBER DEFAULT 0,
 p_init_error_handler       IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_write_to_concurrent_log IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_init_fnd_msg_list        IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_log_errors               IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_add_errors_to_fnd_stack IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_commit                   IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 x_failed_row_id_list       OUT NOCOPY VARCHAR2,
 x_return_status            OUT NOCOPY VARCHAR2,
 x_errorcode                OUT NOCOPY NUMBER,
 x_msg_count                OUT NOCOPY NUMBER,
 x_msg_data                 OUT NOCOPY VARCHAR2)
```

Parameter Description and Validation

This table lists information about the parameters in the Create or Update Organization Profile Extension 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_api_version	IN	NUMBER	Yes	Comment: A decimal number indicating revisions to the API.
p_org_profile_id	IN	NUMBER	Yes	Comment: Unique organization profile ID of the profile record for which you are creating the extension record.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_row_table	IN	EGO_USER_ATTRS_ROW_TABLE	Yes	Comments: EGO_USER_ATTR_ROW_TABLE is a table of EGO_USER_ATTR_ROW_OBJ EGO_USER_ATTR_ROW_OBJ contains row-level data about one attribute group row. ROW_IDENTIFIER is the unique numeric identifier for this attribute group row within a set of rows to be processed; no two EGO_USER_ATTR_ROW_OBJ elements in any single API call can share the same ROW_IDENTIFIER value. The attribute group whose row-level data this EGO_USER_ATTR_ROW_OBJ contains is identified either by ATTR_GROUP_ID or by the combination of ATTR_GROUP_APP_ID, ATTR_GROUP_TYPE, and ATTR_GROUP_NAME. (The first field is the numeric key for an attribute group, and the latter three fields form the composite key for an attribute group.) If the attribute group type has data levels defined and the attribute group is associated at a data level other than the highest data level defined for the attribute group type, the data level values are passed in DATA_LEVEL_1, DATA_LEVEL_2, and DATA_LEVEL_3 (as necessary). TRANSACTION_TYPE indicates the mode of DML operation to be performed on this attribute group row; valid values are EGO_USER_ATTRS_DATA_PVT.G_CREATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_UPDATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_DELETE_MODE, or, if the caller is uncertain whether this row exists in the database, EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE, which indicates that the API should determine whether to create or update this attribute group row.
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ. EGO_USER_ATTR_DATA_OBJ is an object type that contains data for one attribute in an attribute group row. ROW_IDENTIFIER is a foreign key that associates each EGO_USER_ATTR_DATA_OBJ to one EGO_USER_ATTR_ROW_OBJ. ATTR_NAME holds the internal name of the attribute. The value being passed for the attribute is stored in ATTR_VALUE_STR if the attribute is a string (translatable or not), in ATTR_VALUE_NUM if the attribute is a number, in ATTR_VALUE_DATE if the attribute is a date or date time, or in ATTR_DISP_VALUE if the attribute has a value set with distinct internal and display values. **Note:** The attribute value must be passed in exactly one of these four fields

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
				If the attribute is a number that has a Unit of Measure class associated with it, ATTR_UNIT_OF_MEASURE stores the UOM Code for the unit of measure in which the attribute's value will be displayed. However, the value itself will always be passed in ATTR_VALUE_NUM in the base units for the Unit of Measure class, not in the display units (unless they happen to be the same). For example, consider an attribute whose unit of measure class is Length with base unit of centimeters. If the caller wants data for this attribute to be displayed in feet (UOM_CODE *FT*), then ATTR_UNIT_OF_MEASURE should be passed with *FT*. However, no matter in what unit the caller wants to display this attribute, the value in ATTR_VALUE_NUM will always be the attribute's value expressed in centimeters. The final field in the object type, USER_ROW_IDENTIFIER, is a numeric value used when reporting errors for this EGO_USER_ATTR_DATA_OBJ. When the errors are written to the MTL_INTERFACE_ERRORS table, the TRANSACTION_ID column stores the value passed in USER_ROW_IDENTIFIER. To find errors logged for this EGO_USER_ATTR_DATA_OBJ, search for rows in MTL_INTERFACE_ERRORS whose TRANSACTION_ID column values match the passed-in USER_ROW_IDENTIFIER.
p_change_info_table	IN	EGO_USER_ATTR_CHANGE_TABLE	Yes	Comment: Not currently supported.
p_entity_id	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_index	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_code	IN	VARCHAR2	No	Comment: Used in error reporting.
p_debug_level	IN	NUMBER	Yes	Comment: Used in debugging.
p_init_error_handler	IN	VARCHAR2	No	Comment: Indicates whether to initialize ERROR_HANDLER message stack, and open debug session, if applicable.
p_write_to_concurrent_log	IN	VARCHAR2	No	Indicates whether to log ERROR_HANDLER messages to concurrent log. Only applicable when called from concurrent program and when p_log_errors is passed as FND_API.G_TRUE.
p_init_fnd_msg_list	IN	VARCHAR2	No	Comment: Indicates whether to initialize FND_MSG_PUB message stack.
p_add_errors_to_fnd_stack	IN	VARCHAR2	No	Comment: Indicates whether messages written to ERROR_HANDLER message stack will also be written to FND_MSG_PUB message stack.
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_failed_row_id_list	OUT	VARCHAR2	No	Comment: Returns a comma-delimited list of ROW_IDENTIFIERs indicating which attribute group rows failed to be processed.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_errorcode	OUT	NUMBER	No	Comment: Reserved for future use.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on ERROR_HANDLER message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on ERROR_HANDLER message stack upon completion of processing, this parameter contains that message.

Other Validations

- The attribute group and group associations are set up properly.
- The transaction type set in the p_attributes_row_table (EGO_USER_ATTR_ROW_TABLE) is one of the following values: G_CREATE_MODE, G_UPDATE_MODE, G_DELETE_MODE, or G_SYNC_MODE.

Create or Update Person Profile Extension

Description

Use this routine to create or update information in the person extensions tables. This API can be used to maintain records in the HZ_PER_PROFILES_EXT_B and HZ_PER_PROFILES_EXT_TL tables for a given person. The extension tables hold additional information that you decide to capture about a person.

PL/SQL Procedure

```
PROCEDURE Process_Person_Record (
 p_api_version              IN NUMBER,
 p_person_profile_id        IN NUMBER,
 p_attributes_row_table     IN EGO_USER_ATTR_ROW_TABLE,
 p_attributes_data_table    IN EGO_USER_ATTR_DATA_TABLE,
 p_change_info_table        IN EGO_USER_ATTR_CHANGE_TABLE DEFAULT N
ULL,
 p_entity_id                IN NUMBER DEFAULT NULL,
 p_entity_index             IN NUMBER DEFAULT NULL,
 p_entity_code              IN VARCHAR2 DEFAULT NULL,
 p_debug_level              IN NUMBER DEFAULT 0,
 p_init_error_handler       IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_write_to_concurrent_log IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_init_fnd_msg_list        IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_log_errors               IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_add_errors_to_fnd_stack IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_commit                   IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 x_failed_row_id_list       OUT NOCOPY VARCHAR2,
 x_return_status            OUT NOCOPY VARCHAR2,
 x_errorcode                OUT NOCOPY NUMBER,
 x_msg_count                OUT NOCOPY NUMBER,
 x_msg_data                 OUT NOCOPY VARCHAR2)
```

Parameter Description and Validation

This table lists information about the parameters in the Create or Update Person Profile Extension 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_api_version	IN	NUMBER	Yes	Comment: A decimal number indicating revisions to the API.
p_person_profile_id	IN	NUMBER	Yes	Comment: Unique person profile ID of the profile record for which you are creating the extension record.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_row_table	IN	EGO_USER_ATTRS_ROW_TABLE	Yes	Comments: EGO_USER_ATTR_ROW_TABLE is a table of EGO_USER_ATTR_ROW_OBJ EGO_USER_ATTR_ROW_OBJ contains row-level data about one attribute group row. ROW_IDENTIFIER is the unique numeric identifier for this attribute group row within a set of rows to be processed; no two EGO_USER_ATTR_ROW_OBJ elements in any single API call can share the same ROW_IDENTIFIER value. The attribute group whose row-level data this EGO_USER_ATTR_ROW_OBJ contains is identified either by ATTR_GROUP_ID or by the combination of ATTR_GROUP_APP_ID, ATTR_GROUP_TYPE, and ATTR_GROUP_NAME. (The first field is the numeric key for an attribute group, and the latter three fields form the composite key for an attribute group.) If the attribute group type has data levels defined and the attribute group is associated at a data level other than the highest data level defined for the attribute group type, the data level values are passed in DATA_LEVEL_1, DATA_LEVEL_2, and DATA_LEVEL_3 (as necessary). TRANSACTION_TYPE indicates the mode of DML operation to be performed on this attribute group row; valid values are EGO_USER_ATTRS_DATA_PVT.G_CREATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_UPDATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_DELETE_MODE, or, if the caller is uncertain whether this row exists in the database, EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE, which indicates that the API should determine whether to create or update this attribute group row.
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ. EGO_USER_ATTR_DATA_OBJ is an object type that contains data for one attribute in an attribute group row. ROW_IDENTIFIER is a foreign key that associates each EGO_USER_ATTR_DATA_OBJ to one EGO_USER_ATTR_ROW_OBJ. ATTR_NAME holds the internal name of the attribute. The value being passed for the attribute is stored in ATTR_VALUE_STR if the attribute is a string (translatable or not), in ATTR_VALUE_NUM if the attribute is a number, in ATTR_VALUE_DATE if the attribute is a date or date time, or in ATTR_DISP_VALUE if the attribute has a value set with distinct internal and display values. **Note:** The attribute value must be passed in exactly one of these four fields

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
				If the attribute is a number that has a Unit of Measure class associated with it, ATTR_UNIT_OF_MEASURE stores the UOM Code for the unit of measure in which the attribute's value will be displayed. However, the value itself will always be passed in ATTR_VALUE_NUM in the base units for the Unit of Measure class, not in the display units (unless they happen to be the same). For example, consider an attribute whose unit of measure class is Length with base unit of centimeters. If the caller wants data for this attribute to be displayed in feet (UOM_CODE *FT*), then ATTR_UNIT_OF_MEASURE should be passed with *FT*. However, no matter in what unit the caller wants to display this attribute, the value in ATTR_VALUE_NUM will always be the attribute's value expressed in centimeters. The final field in the object type, USER_ROW_IDENTIFIER, is a numeric value used when reporting errors for this EGO_USER_ATTR_DATA_OBJ. When the errors are written to the MTL_INTERFACE_ERRORS table, the TRANSACTION_ID column stores the value passed in USER_ROW_IDENTIFIER. To find errors logged for this EGO_USER_ATTR_DATA_OBJ, search for rows in MTL_INTERFACE_ERRORS whose TRANSACTION_ID column values match the passed-in USER_ROW_IDENTIFIER.
p_change_info_table	IN	EGO_USER_ATTR_CHANGE_TABLE	Yes	Comment: Not currently supported.
p_entity_id	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_index	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_code	IN	VARCHAR2	No	Comment: Used in error reporting.
p_debug_level	IN	NUMBER	Yes	Comment: Used in debugging.
p_init_error_handler	IN	VARCHAR2	No	Comment: Indicates whether to initialize ERROR_HANDLER message stack, and open debug session, if applicable.
p_write_to_concurrent_log	IN	VARCHAR2	No	Indicates whether to log ERROR_HANDLER messages to concurrent log. Only applicable when called from concurrent program and when p_log_errors is passed as FND_API.G_TRUE.
p_init_fnd_msg_list	IN	VARCHAR2	No	Comment: Indicates whether to initialize FND_MSG_PUB message stack.
p_add_errors_to_fnd_stack	IN	VARCHAR2	No	Comment: Indicates whether messages written to ERROR_HANDLER message stack will also be written to FND_MSG_PUB message stack.
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_failed_row_id_list	OUT	VARCHAR2	No	Comment: Returns a comma-delimited list of ROW_IDENTIFIERs indicating which attribute group rows failed to be processed.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_errorcode	OUT	NUMBER	No	Comment: Reserved for future use.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on ERROR_HANDLER message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on ERROR_HANDLER message stack upon completion of processing, this parameter contains that message.

Other Validations

- The attribute group and group associations are set up properly.
- The transaction type set in the p_attributes_row_table (EGO_USER_ATTR_ROW_TABLE) is one of the following values: G_CREATE_MODE, G_UPDATE_MODE, G_DELETE_MODE, or G_SYNC_MODE.

Create or Update Location Extension

Description

Use this routine to create or update information in the location extensions tables. This API can be used to maintain records in the HZ_LOCATIONS_EXT_B and HZ_LOCATIONS_EXT_TL tables for a given location. The extension tables hold additional information that you decide to capture about a location.

PL/SQL Procedure

```
PROCEDURE Process_Location_Record (
 p_api_version              IN NUMBER,
 p_Location_id              IN NUMBER,
 p_attributes_row_table     IN EGO_USER_ATTR_ROW_TABLE,
 p_attributes_data_table    IN EGO_USER_ATTR_DATA_TABLE,
 p_change_info_table        IN EGO_USER_ATTR_CHANGE_TABLE DEFAULT N
ULL,
 p_entity_id                IN NUMBER DEFAULT NULL,
 p_entity_index             IN NUMBER DEFAULT NULL,
 p_entity_code              IN VARCHAR2 DEFAULT NULL,
 p_debug_level              IN NUMBER DEFAULT 0,
 p_init_error_handler       IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_write_to_concurrent_log IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_init_fnd_msg_list        IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_log_errors               IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_add_errors_to_fnd_stack IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_commit                   IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 x_failed_row_id_list       OUT NOCOPY VARCHAR2,
 x_return_status            OUT NOCOPY VARCHAR2,
 x_errorcode                OUT NOCOPY NUMBER,
 x_msg_count                OUT NOCOPY NUMBER,
 x_msg_data                 OUT NOCOPY VARCHAR2)
```

Parameter Description and Validation

This table lists information about the parameters in the Create or Update Location Extension 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_api_version	IN	NUMBER	Yes	Comment: A decimal number indicating revisions to the API.
p_location_id	IN	NUMBER	Yes	Comment: Unique location ID of the location record for which you are creating the extension record.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_row_table	IN	EGO_USER_ATTRS_ROW_TABLE	Yes	Comments: EGO_USER_ATTR_ROW_TABLE is a table of EGO_USER_ATTR_ROW_OBJ EGO_USER_ATTR_ROW_OBJ contains row-level data about one attribute group row. ROW_IDENTIFIER is the unique numeric identifier for this attribute group row within a set of rows to be processed; no two EGO_USER_ATTR_ROW_OBJ elements in any single API call can share the same ROW_IDENTIFIER value. The attribute group whose row-level data this EGO_USER_ATTR_ROW_OBJ contains is identified either by ATTR_GROUP_ID or by the combination of ATTR_GROUP_APP_ID, ATTR_GROUP_TYPE, and ATTR_GROUP_NAME. (The first field is the numeric key for an attribute group, and the latter three fields form the composite key for an attribute group.) If the attribute group type has data levels defined and the attribute group is associated at a data level other than the highest data level defined for the attribute group type, the data level values are passed in DATA_LEVEL_1, DATA_LEVEL_2, and DATA_LEVEL_3 (as necessary). TRANSACTION_TYPE indicates the mode of DML operation to be performed on this attribute group row; valid values are EGO_USER_ATTRS_DATA_PVT.G_CREATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_UPDATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_DELETE_MODE, or, if the caller is uncertain whether this row exists in the database, EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE, which indicates that the API should determine whether to create or update this attribute group row.
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ. EGO_USER_ATTR_DATA_OBJ is an object type that contains data for one attribute in an attribute group row. ROW_IDENTIFIER is a foreign key that associates each EGO_USER_ATTR_DATA_OBJ to one EGO_USER_ATTR_ROW_OBJ. ATTR_NAME holds the internal name of the attribute. The value being passed for the attribute is stored in ATTR_VALUE_STR if the attribute is a string (translatable or not), in ATTR_VALUE_NUM if the attribute is a number, in ATTR_VALUE_DATE if the attribute is a date or date time, or in ATTR_DISP_VALUE if the attribute has a value set with distinct internal and display values. **Note:** The attribute value must be passed in exactly one of these four fields

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
				If the attribute is a number that has a Unit of Measure class associated with it, ATTR_UNIT_OF_MEASURE stores the UOM Code for the unit of measure in which the attribute's value will be displayed. However, the value itself will always be passed in ATTR_VALUE_NUM in the base units for the Unit of Measure class, not in the display units (unless they happen to be the same). For example, consider an attribute whose unit of measure class is Length with base unit of centimeters. If the caller wants data for this attribute to be displayed in feet (UOM_CODE *FT*), then ATTR_UNIT_OF_MEASURE should be passed with *FT*. However, no matter in what unit the caller wants to display this attribute, the value in ATTR_VALUE_NUM will always be the attribute's value expressed in centimeters. The final field in the object type, USER_ROW_IDENTIFIER, is a numeric value used when reporting errors for this EGO_USER_ATTR_DATA_OBJ. When the errors are written to the MTL_INTERFACE_ERRORS table, the TRANSACTION_ID column stores the value passed in USER_ROW_IDENTIFIER. To find errors logged for this EGO_USER_ATTR_DATA_OBJ, search for rows in MTL_INTERFACE_ERRORS whose TRANSACTION_ID column values match the passed-in USER_ROW_IDENTIFIER.
p_change_info_table	IN	EGO_USER_ATTR_CHANGE_TABLE	Yes	Comment: Not currently supported.
p_entity_id	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_index	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_code	IN	VARCHAR2	No	Comment: Used in error reporting.
p_debug_level	IN	NUMBER	Yes	Comment: Used in debugging.
p_init_error_handler	IN	VARCHAR2	No	Comment: Indicates whether to initialize ERROR_HANDLER message stack, and open debug session, if applicable.
p_write_to_concurrent_log	IN	VARCHAR2	No	Indicates whether to log ERROR_HANDLER messages to concurrent log. Only applicable when called from concurrent program and when p_log_errors is passed as FND_API.G_TRUE.
p_init_fnd_msg_list	IN	VARCHAR2	No	Comment: Indicates whether to initialize FND_MSG_PUB message stack.
p_add_errors_to_fnd_stack	IN	VARCHAR2	No	Comment: Indicates whether messages written to ERROR_HANDLER message stack will also be written to FND_MSG_PUB message stack.
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_failed_row_id_list	OUT	VARCHAR2	No	Comment: Returns a comma-delimited list of ROW_IDENTIFIERs indicating which attribute group rows failed to be processed.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_errorcode	OUT	NUMBER	No	Comment: Reserved for future use.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on ERROR_HANDLER message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on ERROR_HANDLER message stack upon completion of processing, this parameter contains that message.

Other Validations

- The attribute group and group associations are set up properly.
- The transaction type set in the p_attributes_row_table (EGO_USER_ATTR_ROW_TABLE) is one of the following values: G_CREATE_MODE, G_UPDATE_MODE, G_DELETE_MODE, or G_SYNC_MODE.

Create or Update Party Site Extension

Description

Use this routine to create or update information in the party site extensions tables. This API can be used to maintain records in the HZ_PARTY_SITES_EXT_B and HZ_PARTY_SITES_EXT_TL tables for a given party site. The extension tables hold additional information that you decide to capture about a party site.

PL/SQL Procedure

```
PROCEDURE Process_PartySite_Record (
 p_api_version             IN NUMBER,
 p_party_site_id           IN NUMBER,
 p_attributes_row_table    IN EGO_USER_ATTR_ROW_TABLE,
 p_attributes_data_table   IN EGO_USER_ATTR_DATA_TABLE,
 p_change_info_table       IN EGO_USER_ATTR_CHANGE_TABLE DEFAULT N
ULL,
 p_entity_id               IN NUMBER DEFAULT NULL,
 p_entity_index            IN NUMBER DEFAULT NULL,
 p_entity_code             IN VARCHAR2 DEFAULT NULL,
 p_debug_level             IN NUMBER DEFAULT 0,
 p_init_error_handler      IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_write_to_concurrent_log IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_init_fnd_msg_list       IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_log_errors              IN VARCHAR2 DEFAULT FND_API.G_TRUE,
 p_add_errors_to_fnd_stack IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 p_commit                  IN VARCHAR2 DEFAULT FND_API.G_FALSE,
 x_failed_row_id_list      OUT NOCOPY VARCHAR2,
 x_return_status           OUT NOCOPY VARCHAR2,
 x_errorcode               OUT NOCOPY NUMBER,
 x_msg_count               OUT NOCOPY NUMBER,
 x_msg_data                OUT NOCOPY VARCHAR2)
```

Parameter Description and Validation

This table lists information about the parameters in the Create or Update Party Site Extension 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_api_version	IN	NUMBER	Yes	Comment: A decimal number indicating revisions to the API.
p_party_site_id	IN	NUMBER	Yes	Comment: Unique party site ID of the party site record for which you are creating the extension record.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_row_table	IN	EGO_USER_ATTRS_ROW_TABLE	Yes	Comments: EGO_USER_ATTR_ROW_TABLE is a table of EGO_USER_ATTR_ROW_OBJ EGO_USER_ATTR_ROW_OBJ contains row-level data about one attribute group row. ROW_IDENTIFIER is the unique numeric identifier for this attribute group row within a set of rows to be processed; no two EGO_USER_ATTR_ROW_OBJ elements in any single API call can share the same ROW_IDENTIFIER value. The attribute group whose row-level data this EGO_USER_ATTR_ROW_OBJ contains is identified either by ATTR_GROUP_ID or by the combination of ATTR_GROUP_APP_ID, ATTR_GROUP_TYPE, and ATTR_GROUP_NAME. (The first field is the numeric key for an attribute group, and the latter three fields form the composite key for an attribute group.) If the attribute group type has data levels defined and the attribute group is associated at a data level other than the highest data level defined for the attribute group type, the data level values are passed in DATA_LEVEL_1, DATA_LEVEL_2, and DATA_LEVEL_3 (as necessary). TRANSACTION_TYPE indicates the mode of DML operation to be performed on this attribute group row; valid values are EGO_USER_ATTRS_DATA_PVT.G_CREATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_UPDATE_MODE, EGO_USER_ATTRS_DATA_PVT.G_DELETE_MODE, or, if the caller is uncertain whether this row exists in the database, EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE, which indicates that the API should determine whether to create or update this attribute group row.
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ. EGO_USER_ATTR_DATA_OBJ is an object type that contains data for one attribute in an attribute group row. ROW_IDENTIFIER is a foreign key that associates each EGO_USER_ATTR_DATA_OBJ to one EGO_USER_ATTR_ROW_OBJ. ATTR_NAME holds the internal name of the attribute. The value being passed for the attribute is stored in ATTR_VALUE_STR if the attribute is a string (translatable or not), in ATTR_VALUE_NUM if the attribute is a number, in ATTR_VALUE_DATE if the attribute is a date or date time, or in ATTR_DISP_VALUE if the attribute has a value set with distinct internal and display values. **Note:** The attribute value must be passed in exactly one of these four fields

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
				If the attribute is a number that has a Unit of Measure class associated with it, ATTR_UNIT_OF_MEASURE stores the UOM Code for the unit of measure in which the attribute's value will be displayed. However, the value itself will always be passed in ATTR_VALUE_NUM in the base units for the Unit of Measure class, not in the display units (unless they happen to be the same). For example, consider an attribute whose unit of measure class is Length with base unit of centimeters. If the caller wants data for this attribute to be displayed in feet (UOM_CODE *FT*), then ATTR_UNIT_OF_MEASURE should be passed with *FT*. However, no matter in what unit the caller wants to display this attribute, the value in ATTR_VALUE_NUM will always be the attribute's value expressed in centimeters. The final field in the object type, USER_ROW_IDENTIFIER, is a numeric value used when reporting errors for this EGO_USER_ATTR_DATA_OBJ. When the errors are written to the MTL_INTERFACE_ERRORS table, the TRANSACTION_ID column stores the value passed in USER_ROW_IDENTIFIER. To find errors logged for this EGO_USER_ATTR_DATA_OBJ, search for rows in MTL_INTERFACE_ERRORS whose TRANSACTION_ID column values match the passed-in USER_ROW_IDENTIFIER.
p_change_info_table	IN	EGO_USER_ATTR_CHANGE_TABLE	Yes	Comment: Not currently supported.
p_entity_id	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_index	IN	NUMBER	No	Comment: Used in error reporting.
p_entity_code	IN	VARCHAR2	No	Comment: Used in error reporting.
p_debug_level	IN	NUMBER	Yes	Comment: Used in debugging.
p_init_error_handler	IN	VARCHAR2	No	Comment: Indicates whether to initialize ERROR_HANDLER message stack, and open debug session, if applicable.
p_write_to_concurrent_log	IN	VARCHAR2	No	Indicates whether to log ERROR_HANDLER messages to concurrent log. Only applicable when called from concurrent program and when p_log_errors is passed as FND_API.G_TRUE.
p_init_fnd_msg_list	IN	VARCHAR2	No	Comment: Indicates whether to initialize FND_MSG_PUB message stack.
p_add_errors_to_fnd_stack	IN	VARCHAR2	No	Comment: Indicates whether messages written to ERROR_HANDLER message stack will also be written to FND_MSG_PUB message stack.
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_failed_row_id_list	OUT	VARCHAR2	No	Comment: Returns a comma-delimited list of ROW_IDENTIFIERs indicating which attribute group rows failed to be processed.
x_return_status	OUT	VARCHAR2	No	Comment: A code indicating whether any errors occurred during processing.
x_errorcode	OUT	NUMBER	No	Comment: Reserved for future use.
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on ERROR_HANDLER message stack upon completion of processing.
x_msg_data	OUT	VARCHAR2	No	Comment: If exactly one message exists on ERROR_HANDLER message stack upon completion of processing, this parameter contains that message.

Other Validations

- The attribute group and group associations are set up properly.
- The transaction type set in the p_attributes_row_table (EGO_USER_ATTR_ROW_TABLE) is one of the following values: G_CREATE_MODE, G_UPDATE_MODE, G_DELETE_MODE, or G_SYNC_MODE.

18

Formatting API Use

This chapter describes Phone Parsing and Formatting, Name and Address Formatting, and General Data Formatting APIs. The provided information includes: PL/SQL record structure and procedure, Java inner class and method, and parameter descriptions and validations.

This chapter covers the following topics:

- Phone Parsing and Formatting APIs
- Name and Address Formatting APIs
- General Data Formatting API

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
)
```

Java Method

```
public static void phoneParse(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_raw_phone_number,
    String p_territory_code,
    String  [] x_phone_country_code,
    String  [] x_phone_area_code,
    String  [] x_phone_number,
    String  [] x_mobile_flag,
    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 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.

There are two overloaded procedures for this API. Use the first signature if the contact point ID for the phone number which is to be formatted is known. If not, use the second signature where the individual phone number components, country code, area code, and phone number, can be passed for formatting the phone number.

Note: Setting up user preferences is intended for future release.

PL/SQL Procedure

The p_contact_point_id parameter is only in the first procedure, which is used when the contact point ID of the phone number is known. These parameters are only in the second procedure, which is used when the contact point ID is unknown: p_territory_code, p_phone_country_code, p_phone_area_code, and p_phone_number.

```
PROCEDURE    phone_display(
   p_init_msg_list          IN   VARCHAR2 := FND_API.G_FALSE,
   p_contact_point_id       IN   NUMBER,
   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
)
```

Java Method

The p_contact_point_id parameter is only in the first procedure, which is used when the contact point ID of the phone number is known. These parameters are only in the second procedure, which is used when the contact point ID is unknown: p_territory_code, p_phone_country_code, p_phone_area_code, and p_phone_number.

```
public static void phoneDisplay(
    OracleConnection _connection,
    String p_init_msg_list,
    BigDecimal p_contact_point_id,
    String p_territory_code,
    String p_phone_country_code,
    String p_phone_area_code,
    String p_phone_number,
    String  [] x_formatted_phone_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 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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. This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.
p_phone_country_code	IN	VARCHAR2	No	Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.
p_phone_area_code	IN	VARCHAR2	No	Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.
p_phone_number	IN	VARCHAR2	No	Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.
p_contact_point_id	IN	NUMBER	No	Validation: Contact point type must be Phone. Comment: Identifies the contact point in HZ_CONTACT_POINTS which needs to be formatted. This parameter is only in the first procedure, which is used when the contact point ID of the phone number is known.
x_formatted_phone_number	OUT	VARCHAR2	No	Comment : Returns the number after formatting

Check Mobile Phone Number API

Description

Use this routine to check whether the country code, area code, and phone number combination is a mobile number. This API returns *Y* or *N* to indicate whether is it a mobile phone number or not.

PL/SQL Procedure

```
PROCEDURE check_mobile_phone (
p_init_msg_list      IN   VARCHAR2 := fnd_api.g_false,
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_mobile_flag        OUT  VARCHAR2,
x_return_status      OUT  VARCHAR2,
x_msg_count          OUT  NUMBER,
x_msg_data           OUT  VARCHAR2)
```

Java Method

```
public static void checkMobilePhone(
    OracleConnection _connection,
    String p_init_msg_list,
    String p_phone_country_code,
    String p_phone_area_code,
    String p_phone_number,
    String  [] x_mobile_flag,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data  ) throws SQLException;
```

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
)
```

Java Method

```
public static void formatName(

    OracleConnection _connection,
    BigDecimal p_party_id,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_ref_language_code,
    String p_ref_territory_code,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data,
    String  [] x_formatted_name,
    BigDecimal  [] x_formatted_lines_cnt,
    ARRAY  [] x_formatted_name_tbl  ) throws SQLException;
```

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.
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, * * for HTML output.
p_space_replace	In	Varchar2	No	Comment: Characters to substitute for *blank* valued delimiters. For example, * * 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. • S=Success • E=Error • U=Unexpected Error
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.
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
)
```

Java Method

```
public static void formatName(

    OracleConnection _connection,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_ref_language_code,
    String p_ref_territory_code,
    String p_person_title,
    String p_person_first_name,
    String p_person_middle_name,
    String p_person_last_name,
    String p_person_name_suffix,
    String p_person_known_as,
    String p_first_name_phonetic,
    String p_middle_name_phonetic,
    String p_last_name_phonetic,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data,
    String  [] x_formatted_name,
    BigDecimal  [] x_formatted_lines_cnt,
    ARRAY  [] x_formatted_name_tbl  ) throws SQLException;
```

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 *II* or *III*, or a string such as *Jr.* or *Sr.*
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

There is also a function version of the format_name procedure that can be used in a SQL statement.

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
```

Java Method

```
public static String formatName(
    OracleConnection _connection,
    BigDecimal p_party_id,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_ref_language_code,
    String p_ref_territory_code  ) throws SQLException;
```

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
)
```

Java Method

```
public static void formatAddress(
    OracleConnection _connection,
    BigDecimal p_location_id,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_to_language_code,
    String p_country_name_lang,
    String p_from_territory_code,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data,
    String  [] x_formatted_address,
    BigDecimal  [] x_formatted_lines_cnt,
    ARRAY  [] x_formatted_address_tbl  ) throws SQLException;
```

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.
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, * * for HTML output.
p_space_replace	In	Varchar2	No	Comment: Characters to substitute for *blank* valued delimiters. For example, * * 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. • S=Success • E=Error • U=Unexpected Error
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.
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.

PL/SQL Procedure

```
PROCEDURE format_address (
-- input
    p_style_code            IN   VARCHAR2 DEFAULT NULL,
    p_style_format_code    IN   VARCHAR2 DEFAULT NULL,
    p_line_break            IN   VARCHAR2 DEFAULT NULL,
    p_space_replace     IN  VARCHAR2 DEFAULT NULL,
-- optional 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,
-- address components
    p_address_line_1  IN  VARCHAR2 DEFAULT NULL,
    p_address_line_2  IN  VARCHAR2 DEFAULT NULL,
    p_address_line_3  IN  VARCHAR2 DEFAULT NULL,
    p_address_line_4  IN  VARCHAR2 DEFAULT NULL,
    p_city            IN  VARCHAR2 DEFAULT NULL,
    p_postal_code  IN  VARCHAR2 DEFAULT NULL,
    p_state   IN  VARCHAR2 DEFAULT NULL,
    p_province    IN  VARCHAR2 DEFAULT NULL,
    p_county   IN  VARCHAR2 DEFAULT NULL,
    p_country   IN  VARCHAR2 DEFAULT NULL,
    p_address_lines_phonetic  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  string_tbl_type);
```

Java Method

```
public static void formatAddress(
    OracleConnection _connection,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_to_language_code,
    String p_country_name_lang,
    String p_from_territory_code,
    String p_address_line_1,
    String p_address_line_2,
    String p_address_line_3,
    String p_address_line_4,
    String p_city,
    String p_postal_code,
    String p_state,
    String p_province,
    String p_county,
    String p_country,
    String p_address_lines_phonetic,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data,
    String  [] x_formatted_address,
    BigDecimal  [] x_formatted_lines_cnt,
    ARRAY  [] x_formatted_address_tbl  ) throws SQLException;
```

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

There is also a function version of the format_address procedure that can be used in a SQL statement.

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
```

Java Method

```
public static String formatAddress(
    OracleConnection _connection,
    BigDecimal p_location_id,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_to_language_code,
    String p_country_name_lang,
    String p_from_territory_code  ) throws SQLException;
```

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_data(
-- 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
)
```

Java Method

```
public static void formatData(
    OracleConnection _connection,
    String p_object_code,
    String p_object_key_1,
    String p_object_key_2,
    String p_object_key_3,
    String p_object_key_4,
    String p_style_code,
    String p_style_format_code,
    String p_line_break,
    String p_space_replace,
    String p_ref_language_code,
    String p_ref_territory_code,
    String  [] x_return_status,
    BigDecimal  [] x_msg_count,
    String  [] x_msg_data,
    String  [] x_formatted_data,
    BigDecimal  [] x_formatted_lines_cnt,
    ARRAY  [] x_formatted_data_tbl  ) throws SQLException;
```

Parameter Description and Validation

The following table lists information about the parameters in the General Data 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_object_code	In	Varchar2	No	Comment: Object code, such as table name or view name, for which you want the data formatted.
p_object_key_1	In	Varchar2	No	Object keys combination form the primary key of the object for which you wish the data formatted.
p_object_key_2	In	Varchar2	No	
p_object_key_3	In	Varchar2	No	
p_object_key_4	In	Varchar2	No	
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, which is the usual case. This parameter is ignored if p_style_format_code is passed.
p_style_format_code	In	Varchar2	No	Comment: Supplied if the caller wants to 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 * * for HTML output.
p_space_replace	In	Varchar2	No	Comment: Characters to substitute for blank valued delimiters, for example, *&nsbp;* for HTML output.
p_ref_language_ code	In	Varchar2	No	Comment: Reference Language (context). Default: The value from the HZ: Reference Language profile option or the current session NLS Language setting.
p_ref_territory_code	In	Varchar2	No	Comment: Reference Territory (context). Default: The value from the HZ: Reference Territory profile option or the current session NLS Territory setting.
x_formatted_data	Out	Varchar2	No	Comment: The formatted output data.

19

Business Event System (BES) Callouts

This chapter includes information about the use of Business Event System Callouts.

This chapter covers the following topics:

- Overview
- Installation and Setup
- Trading Community Architecture Events

Overview

These callouts, to the Oracle Workflow Business Event System, enable you to customize data usage in the TCA registry and the customer account layer without modifying the core TCA product.

- Third party applications can subscribe to TCA events.
- Easy to administer and maintain third party processes.
- Includes published events.

Installation and Setup

Installation

The TCA callout structure is based on the Business Event System (BES) using Oracle Workflow 2.6.

If you are using Oracle E-Business Suite, Release 11*i*.7, then no specific installation is required. Otherwise, you must install the 11*i*.HZ.G mini-pack, patch 2111967.

Setup

Event Subscription

1. Log on to Oracle Applications using the System Administrator responsibility.
2. Navigate to menu: Workflow > Add Subscription to Event

Each TCA event uses the same naming convention:

```
oracle.apps.ar.hz<Entity name>.<action>
```

For example, if you subscribe your business process to the Customer Account creation event, then you must subscribe your routine to the oracle.apps.ar.hz.CustAccount.create event. Or, if you subscribe your business process to the modification of an organization, then you must subscribe your routine to the oracle.apps.ar.hz.Organization.update event.

Profile Option Control

A system administrator can enable or prevent the use of the TCA callout structure by using the HZ: Execute API Callout profile option. If this option is set to Yes, then you can raise TCA events when manipulating TCA entities.

Data Cleaning Subscription

When TCA events are fired, a transactional table stores the data corresponding to that event so that a user subscription can access the data when the event was raised.

For data purging purposes, in addition to the user subscriptions, you can use the HZ_PARAM_ELT.PARAM_DELETE cleaning subscription, which must be subscribed to the event on which you have added your subscription.

> **Note:** You can control the execution sequence of subscriptions by controlling the Phase attribute of a subscription. To give your subscriptions access to TCA data, you must specify the highest phase number to the TCA cleaning subscription. The recommended phase number for the HZ_PARAM_ELT.PARAM_DELETE subscription is 500.

Workflow Agent Listener Concurrent Program

You should run subscriptions in deferred mode using the Oracle Workflow 2.6 Business Event System, so that no overhead is added to the process that raised the event. The phase number of an user defined subscription should be greater than 99.

When the TCA callout structure raises an event, the event message, corresponding to that instance of event, is sent to the deferred queue. Subscriptions are executed when the event message is consumed. That is when the Workflow Agent Listener with the WF_DEFERRED parameter runs. When this concurrent program runs, every subscription from every instance of events currently in the DEFERRED queue runs. The request group of the System Administrator responsibility includes this concurrent program.

Trading Community Architecture Events

This table provides the following information about the TCA events:

- Name of the event
- Description of the physical entity
- ID parameter name

Name	Description of the Physical Entity	ID Parameter Name
Person: • oracle.apps.ar.hz.Person.create • oracle.apps.ar.hz.Person.update	Party is a generic concept about an entity that can establish relationships with another entity. For example, a party can be an Organization like Oracle, a Person like Jane Doe, a Group like World Wide Web Consortium, or a Relationship like Jane Doe at Oracle. Based on this concept, a person is a party of the Person type. The HZ_PERSON_PROFILES table stores personal and family information about a party of the Person type. For example, this table could contain the correct spelling and phonetic pronunciation of the person's name. HZ_PARTIES table contains denormalized information from the HZ_PERSON_PROFILES, HZ_ORGANIZAT ION_PROFILES, HZ_LOCATIONS, HZ_CONTACT_PO INTS and HZ_PERSON_LANGUAGE tables.	PARTY_ID
Organization: • oracle.apps.ar.hz.Organization. create • oracle.apps.ar.hz.Organization. update	An organization is a party of the Organization type. HZ_ORGANIZATION_PROFILES table stores a variety of information about a party. This table gets populated during the creation of a party of the Organization type. This table can also store historical data for the organization. Each time an organization's information is updated, the effective end date column for the original record is updated and a new record that contains the updated information is created.	PARTY_ID
Group: • oracle.apps.ar.hz.Group.create • oracle.apps.ar.hz.Group.update	A group is a party of the Group type. There is no specific table for groups. The parties are stored in the HZ_PARTIES table.	PARTY_ID
Location: • oracle.apps.ar.hz.Location.create • oracle.apps.ar.hz.Location.update	HZ_LOCATIONS table stores information about a delivery or postal address such as building number, street address, postal code, and directions to location. This table provides physical location information about parties (organizations and persons) and customer accounts.	LOCATION_ ID
Party Site: • oracle.apps.ar.hz.PartySite.create • oracle.apps.ar.hz.PartySite.update	A party can have one or more locations and a location can be used by one or more parties. The Party Site relational entity represents the association of a party and its locations. The HZ_PARTY_SITES table links a party (HZ_PARTIES) and a location (HZ_LOCATIONS), and stores location-specific party information such as MAILSTOP and ADDRESSEE. For example, 500 Oracle Parkway can be specified as a party site for Oracle. This party site can n be used for multiple customer accounts with the same party.	PARTY_SITE_ ID
Party Site Use: • oracle.apps.ar.hz.PartySiteUse. create • oracle.apps.ar.hz.PartySiteUse. update	A relation between a party and a location can have several uses such as delivery or billing. The HZ_PARTY_SITE_USES table stores information about how a party site is used. Party sites can have multiple uses, for example; Ship-To and Bill-To.	PARTY_SITE_ USE_ID

Name	Description of the Physical Entity	ID Parameter Name
Relationship: • oracle.apps.ar.hz.Relationship.create • oracle.apps.ar.hz.Relationship.update	The HZ_RELATIONSHIPS table stores information about relationships between one party and another party. The table specifies the subject and object, as well as the relationship that exists between the two parties. For example, if the party relationship is a Parent Of relationship, a holding company could be the subject in the relationship while one of its subsidiaries could be the object.	RELATIONSHIP_ID
Organization Contact: • oracle.apps.ar.hz.OrgContact.create • oracle.apps.ar.hz.OrgContact.update	A contact is a contact person for a party. HZ_ORG_CONTACTS table stores information about the position of the contact for a party or party site. The records in this table provide information about a contact's position such as JOB_TITLE, RANK and general contact information. This table is not used to store information about a specific person or organization. For example, the table may include a record for the position of vice president of manufacturing that indicates that the contact is a senior executive, but it would not include the name of the person in that position.	ORG_CONTACT_ID
Organization Contact Role: • oracle.apps.ar.hz.OrgContactRole.create • oracle.apps.ar.hz.OrgContactRole.update	Contacts could have multiple roles. The HZ_ORG_CONTACT_ROLES table stores information about the role of the contact position that is specified in HZ_ORG_CONTACTS table. For example, a vice president of manufacturing may have a custom-defined role as a member of a capital expenditures review board.	ORG_CONTACT_ROLE_ID
Customer Account: • oracle.apps.ar.hz.CustAccount.create • oracle.apps.ar.hz.CustAccount.update	A customer account is a customer relationship established with a party. The HZ_CUST_ACCOUNTS table stores information about customer relationships established with a party. Because a party can have multiple customer accounts, this table could contain several records for a same party. For example, an individual person may establish a personal account, a family account, and a professional account for a consulting practice. Note the focus of this table is a business relationship and how the transactions are conducted in the relationship.	CUST_ACCOUNT_ID PARTY_ID CUST_ACCOUNT_PROFILE_ID P_CREATE_PROFILE_AMT
Customer Account Role: • oracle.apps.ar.hz.CustAccountRole.create • oracle.apps.ar.hz.CustAccountRole.update	A party can play a role in a customer account. HZ_CUST_ACCOUNT_ROLES table stores information about the role or function that a party performs in relation to a customer account. For example, Jane Doe might be a legal contact for Vision Corporation.	CUST_ACCT_ROLE_ID

Name	Description of the Physical Entity	ID Parameter Name
Role Responsibility: • oracle.apps.ar.hz.RoleResponsibility.create • oracle.apps.ar.hz.RoleResponsibility.update	A party can play a role in a customer account. A role in a customer account can have several responsibilities. HZ_ROLE_RESPONSIBILITY table stores information about the required or expected activities of a party based on the party's role or function in relation to an account. Note that the role referred to in this table may not be the same as the role of a contact in an organization. For example, in this table you may store "Accepts quality of received materials" as a responsibility of the Incoming Material Quality Control role specified in the HZ_CUST_ACCOUNT_ROLES table. The party with the role responsibility may not be an individual person.	RESPONSIBILITY_ID
Customer Account Relationship: • oracle.apps.ar.hz.CustAcctRelate.create • oracle.apps.ar.hz.CustAcctRelate.update	Customer Accounts can have relationships between m. HZ_CUST_ACCT_RELATE_ALL table stores information about relationships between customer accounts. A flag lets you indicate whether a relationship is reciprocal.	CUST_ACCOUNT_ID RELATED_CUST_ACCOUNT_ID
Customer Account Site: • oracle.apps.ar.hz.CustAcctSite.create • oracle.apps.ar.hz.CustAcctSite.update	A customer account can have several customer account sites. Each customer account site is located in one location. HZ_CUST_ACCT_SITE_ALL table stores information about customer sites, or locations, for customer accounts. One customer account can have multiple sites. Address information for a site is stored in HZ_LOCATIONS table.	CUST_ACCT_SITE_ID
Customer Account Site Use: • oracle.apps.ar.hz.CustAcctSiteUse.create • oracle.apps.ar.hz.CustAcctSiteUse.update	A customer site in one location can have several business purposes. HZ_CUST_SITE_USES_ALL table stores information about the business purposes assigned to a customer account site. A customer account site can have multiple purposes for a customer account site; however each record in this table only specifies one purpose for a customer account site. For example, a customer account site may be assigned as the Ship-To site in one record and as the Bill-To site in another. Note that address information is stored in the HZ_LOCATIONS table.	SITE_USE_ID CUST_ACCOUNT_PROFILE_ID P_CREATE_PROFILE P_CREATE_PROFILE_AMT

Name	Description of the Physical Entity	ID Parameter Name
Customer Profile Amount: • oracle.apps.ar.hz.CustProfileAmt.create • oracle.apps.ar.hz.CustProfileAmt.update	A customer account has an updateable, defaulted credit limit, which is part of the customer's profile information. HZ_CUST_PROFILE_AMTS table stores information about the credit limits specified for a customer profile class for a single currency. The credit limits of the profile class can be assigned to specific customer accounts or customer account sites. Many of the values in this table may be default values from the HZ_CUST_PROF_CLASS_AMTS table, which can be modified in this table for a specific customer account or customer account site. For example while the profile class Large Independent Retail Stores may have an overall credit limit of 250,000 Irish punts, you may limit a new customer in this class to an overall credit limit of 50,000 Irish punts.	CUST_ACCT_PROFILE_AMT_ID
Customer Profile: • oracle.apps.ar.hz.CustomerProfile.create • oracle.apps.ar.hz.CustomerProfile.update	A customer account has a customer profile class that provides default customer account attributes. The HZ_CUST_PROFILES table stores information about credit characteristics that are common across a group of customer accounts. The characteristics specified in this table can be used as default characteristics for similar customer accounts. For example, you can create a profile class called Large Independent Retail Stores and specify several attributes that describe this class of customer. In future, you can assign new customers to this class so that new customer inherits characteristics of class.	
Contact Point: • oracle.apps.ar.hz.ContactPoint.create • oracle.apps.ar.hz.ContactPoint.update	The HZ_CONTACT_POINT table stores information about how to communicate to parties or party sites using electronic media such as e-mail, Electronic Data Exchange Interface (EDI), telephone, telex and the Internet. For example, telephone-related data can include the type of telephone line, a touch-tone indicator, a country code, the area code, the telephone number, and an extension number to a specific handset.	CONTACT_POINT_ID
Contact Preference: • oracle.apps.ar.hz.Contact Preference.create • oracle.apps.ar.hz.Contact Preference.update	A customer account can express preferences on when and how to be contacted. The HZ_CONTACT_PREFERENCES table stores information on how to contact a customer. The major related tables are the HZ_PARTIES, HZ_PARTY_SITES table and the HZ_CONTACT_POINTS table. For example, a customer of the Person type, Ms Jane Doe could express the preference not to be contacted from October 1, 2000 to October 15, 2000, in this case the record in the HZ_CONTACT_PREFERENCES table is related to the HZ_PARTIES table. Ms Doe can also restrict her contact preference to a particular address, in which case, the record in the HZ_CONTACT_PREFERENCES table is related to the HZ_PARTY_SITE table.	CONTACT_PREFERENCE_ID

Name	Description of the Physical Entity	ID Parameter Name
Credit Rating: • oracle.apps.ar.hz.CreditRating.create • oracle.apps.ar.hz.CreditRating.update	Creditworthiness is part of the information about a party. The HZ_CREDIT_RATINGS table stores information about the creditworthiness of a party. A credit rating firm, such as D&B, usually supplies this information.	CREDIT_RATING_ID
Certification: • oracle.apps.ar.hz.Certification.create • oracle.apps.ar.hz.Certification.update	Certification is part of an organization's information. The HZ_CERTIFICATIONS table stores information about accreditation that is usually awarded to an Organization party following the evaluation of the party by the awarding organization. For example, the International Organization for Standardization has established standards for quality systems. An ISO9000 certification is awarded after an evaluation of an organization that demonstrates adherence to the standards specified by ISO.	CERTIFICATION_ID
Citizenship: • oracle.apps.ar.hz.Citizenship.create • oracle.apps.ar.hz.Citizenship.update	Citizenship is part of a person's information. The HZ_CITIZENSHIP table stores information about a person's claimed nationality. People can have more than one citizenship in their lifetimes and can have multiple citizenships at the same time.	CITIZENSHIP_ID
Class Category: • oracle.apps.ar.hz.ClassCategory.create • oracle.apps.ar.hz.ClassCategory.update	Class Categories provides a way to classify parties and party sites. The HZ_CLASS_CATEGORIES table stores the information about categories that is used to classify parties and related entities. The possible values for class categories include NAICS, SIC1987 and customer category.	CLASS_CATEGORY
Class Category Use: • oracle.apps.ar.hz.ClassCategoryUse.create • oracle.apps.ar.hz.ClassCategoryUse.update	Classification is an open structure. Class Category Use indicates which table uses which class category. The HZ_CLASS_CATEGORY_USES table stores information about the tables that use a particular class category. For example, the SIC 1987 class category can be used to classify the parties that have the Organization party type.	CLASS_CATEGORY OWNER_TABLE COLUMN_NAME
Class Code Relationship: • oracle.apps.ar.hz.ClassCodeRelation.create • oracle.apps.ar.hz.ClassCodeRelation.update	Class Code Relationships provides a method to build hierarchy structures within Class Codes. The HZ_CLASS_CODE_RELATIONS table stores information about relationships between parent and child classification codes and the categories in the HZ_CLASS_CATEGORIES table. For example, within the North America Industry Classification System NAICS Manufacturing category (code 33), the code for radio and TV manufacturers (code 33422) is the child in relation to the classification code for Communications Equipment manufacturer (code 3342).	CLASS_CATEGORY CLASS_CODE SUB_CLASS_CODE

Name	Description of the Physical Entity	ID Parameter Name
Code Assignment: • oracle.apps.ar.hz.CodeAssignment.create • oracle.apps.ar.hz.CodeAssignment.update	Code Assignment 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 FND_LOOKUP_VALUES table to the names of the parties or other entities stored in the table identified in the OWNER_TABLE_NAME column.	CODE_ASSIGNMENT_ID
Education: • oracle.apps.ar.hz.Education.create • oracle.apps.ar.hz.Education.update	A party of the Person type can have an education history. The HZ_EDUCATION table stores general information about the educational history of a party of the Person type. For example, a person may have attended the University of California from 1995 to 1999 and received an undergraduate degree in Computer Science with a GPA of 4.0. This table stores the information about the university, the period when the person attended, and the subject that the student majored in.	EDUCATION_ID
Employment History: • oracle.apps.ar.hz.EmploymentHistory.create • oracle.apps.ar.hz.EmploymentHistory.update	A party of the Person type can have an employment history. The HZ_EMPLOYMENT_HISTORY table stores information about the employment and military service history of a party of the Person type. For example, John Smith, an accountant, whose supervisor was Jane Doe, worked in US Operation division of Vision Corporate from July 12, 1994 to January 23, 2000.	EMPLOYMENT_HISTORY_ID
Financial Profile: • oracle.apps.ar.hz.FinancialProfile.create • oracle.apps.ar.hz.FinancialProfile.update	The HZ_FINANCIAL_PROFILE table stores information about the financial accounts owned by a party. For example, when you perform a credit evaluation of a prospective customer, data may be entered about the balance in he prospective customer's bank account.	FINANCIAL_PROFILE_ID
Financial Report: • oracle.apps.ar.hz.FinancialReport.create • oracle.apps.ar.hz.FinancialReport.update	The HZ_FINANCIAL_REPORTS table stores information about the reports on the financial status of a party. Detailed nmerical financial data is available in the HZ_FINANCIAL_NUMBERS table. For example, the records in this table can indicate whether the numerical data in the HZ_FINANCIAL_NUMBERS table are actual, estimated, consolidated, and so on.	FINANCIAL_REPORT_ID
Financial Number: • oracle.apps.ar.hz.FinancialNumber.create • oracle.apps.ar.hz.FinancialNumber.update	The HZ_FINANCIAL_NUMBERS table stores detailed financial information for the HZ_FINANCIAL_REPORTS table. For example, a record could store the information that projected leasing revenue includes 1,000,000 Italian lira.	FINANCIAL_NUMBER_ID

Name	Description of the Physical Entity	ID Parameter Name
Person Interest: • oracle.apps.ar.hz.PersonInterest.create • oracle.apps.ar.hz.PersonInterest.update	The HZ_PERSON_INTEREST table stores information about sports, hobbies and personal interests of a party of the Person type. This information could be helpful in developing a more personalized relationship with a business contact. For example, a customer may indicate that he has had an interest in driving in sports car races for the past seven years or that he enjoys attending baseball games.	PERSON_INTEREST_ID
Person Language: • oracle.apps.ar.hz.PersonLanguage.create • oracle.apps.ar.hz.PersonLanguage.update	The HZ_PERSON_LANGUAGE table stores information about a language spoken by a party of the Person type. For example, a person may speak Spanish as her primary language. You would create another record if she speaks French, but it is not her primary language. Note that a separate record must exist for each language.	LANGUAGE_USE_REFERENCE_ID
Work Class: • oracle.apps.ar.hz.WorkClass.create • oracle.apps.ar.hz.WorkClass.update	The HZ_WORK_CLASS table stores custom-defined classification information about a person's job title and work experience. For example, one system may use classifications such as "Electrician" and "Master" while another may use job codes and seniority indicators such as "EM3" and "over 20" to indicate a class 3 Master Electrician with over 20 years of experience. Note that there may be multiple records in the HZ_WORK_CLASS table for each record in the HZ_EMPLOYMENT_HISTORY table.	WORK_CLASS_ID
Original System Reference: • oracle.apps.ar.hz.origSystemRef.create • oracle.apps.ar.hz.origSystemRef.update	The HZ_ORIG_SYSTEM_REFERENCES table stores the mapping between source system references and the owner tables. The source system reference identifies the system that was the source of the data stored in a record. The owner table is the TCA table that stores that data.	ORIG_SYSTEM_REF_ID

A

Sample TCA API Code

This appendix provides sample code that uses the Oracle Trading Community Architecture APIs.

This appendix covers the following topics:

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

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(I||'. '||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 HzPartyV
2Pub.OrganizationRec();
    BigDecimal[]                  x_party_id = new BigDecimal[1];
    String[]                             x_party_number = new Strin
g[1];
    BigDecimal[]                 x_profile_id = new BigDecimal[1];

    String[]                             x_return_status = new Stri
ng[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.

PARTY_ID	PARTY_NUMBER	PARTY_NAME	OBJ_VER	CREATED_BY_MOD
1272023	1268621	ABC Corporation	1	TCA_EXAMPLE

```
select organization_profile_id, party_id, organization_name, obje
ct_version_number, created_by_module
from hz_organization_profiles
where party_id = 1272023;
```

The following table provides information about this select statement.

ORG_PROFILE_ID	PARTY_ID	ORGANIZATION_ NAME	OBJ_VER_NUM	CREATED_BY_M
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(

               '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(I||'. '||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 HzLocationV2P
ub.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_V2PU
B.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(I||'. '||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 HzPartySi
teV2Pub.PartySiteRec();
    BigDecimal[]                x_party_site_id = new BigDecimal[
1];
    String[]                            x_party_site_number = new
```

```
 String[1];

    String[]                              x_return_status = new Str
ing[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 ? := 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_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

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

```
select address1, address2, city, state, postal_code, country
from hz_parties
where party_id = 1272023;
```

Create a Party Site Use

Using the same party site created in the previous example.

The following example uses the party site created in the previous example.

PL/SQL Example

```
DECLARE
p_party_site_use_rec         HZ_PARTY_SITE_V2PUB.PARTY_SITE_USE_RE
C_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(I||'. '||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 Hz
PartySiteV2Pub.PartySiteUseRec();


    BigDecimal[]                 x_party_site_use_id = new BigDeci
mal[1];

    String[]                           x_return_status = new Stri
```

```
ng[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_p
er_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_TY
PE;
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_coun t           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_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);

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(I||'. '||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 HzContactP
ointV2Pub.EdiRec();
    HzContactPointV2Pub.EmailRec      p_email_rec = new HzContactP
ointV2Pub.EmailRec();
    HzContactPointV2Pub.PhoneRec      p_phone_rec = new HzContactP
ointV2Pub.PhoneRec();
    HzContactPointV2Pub.TelexRec       p_telex_rec = new HzContact
PointV2Pub.TelexRec();
    HzContactPointV2Pub.WebRec         p_web_rec = new HzContactPo
intV2Pub.WebRec();
    BigDecimal[]                 x_contact_point_id = new BigDecim
al[1];
    String[]                            x_return_status = new Str
ing[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 HzContactPointV2
Pub();

      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, ow
ner_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 = '||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(I||'. '||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 HzPa
rtyContactV2Pub.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 HzPartyContactV2
Pub();

      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_PAR
TIES";
      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_PART
IES";
      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.value
Of("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, ti
tle, 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_n
ame,
          object_id, object_type, object_table_name, directional_f
lag
          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.

PARTY_ID	PARTY_NUMBER	PARTY_NAME	PARTY_TYPE
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 curren
t 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(I||'. '||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 HzLocationV2P
ub.LocationRec();

    String[]                              x_return_status = new Str
ing[1];
    BigDecimal[]                          x_msg_count = new BigDeci
mal[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.CU
ST_ACCOUNT_REC_TYPE;
p_person_rec                                HZ_PARTY_V2PUB.PERSON_RE
C_TYPE;
p_customer_profile_rec                      HZ_CUSTOMER_PROFILE_V2PU
B.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(I||'. '||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 HzC
ustAccountV2Pub.CustAccountRec();
    HzPartyV2Pub.PersonRec p_person_rec = new HzPartyV2Pub.PersonR
ec();
    HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_r
ec = new HzCustomerProfileV2Pub.CustomerProfileRec();
    BigDecimal[]                x_cust_account_id = new BigDecima
l[1];
    String[]                           x_account_number = new Str
ing[1];
    BigDecimal[]                x_party_id = new BigDecimal[1];
    String[]                           x_party_number = new Strin
g[1];
    BigDecimal[]                x_profile_id = new BigDecimal[1];

    String[]                           x_return_status = new Stri
ng[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_las
t_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(I||'. '||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;

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 BigD
ecimal[1];

    String[]                            x_return_status = new Stri
ng[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 HzCustAcco
untSiteV2Pub();

      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_R
EC_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(I||'. '||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 = ne
w HzCustAccountSiteV2Pub.CustSiteUseRec();
    HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_r
```

```
ec = new HzCustomerProfileV2Pub.CustomerProfileRec();
    BigDecimal[]                          x_site_use_id = new BigDe
cimal[1];

    String[]                              x_return_status = new Str
ing[1];
    BigDecimal[]                          x_msg_count = new BigDeci
mal[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 HzCustAcco
untSiteV2Pub();

      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, o
rg_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(I||'. '||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 HzC
ustAccountV2Pub.CustAccountRec();

    String[]                              x_return_status = new Stri
ng[1];
    BigDecimal[]                          x_msg_count = new BigDecim
al[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	

Create an Organization Profile Extensions Record

PL/SQL Example

This example involves processing two attribute groups.

```
DECLARE
    ------------------------------
    --  Declare Primary Keys    --
    ------------------------------
  l_org_profile_id         NUMBER;

    ------------------------------------------------
    --  Declare row and data objects and metadata    --
    ------------------------------------------------
    l_user_attr_data_table   EGO_USER_ATTR_DATA_TABLE;
    l_user_attr_row_table    EGO_USER_ATTR_ROW_TABLE;

    l_application_id        NUMBER;
    l_attr_group_type       VARCHAR2(40);
    l_attr_group1_name      VARCHAR2(30);
    l_attr_group2_name      VARCHAR2(30);

    ---------------------------------------
    -- Declare various testing variables --
    ---------------------------------------
    l_start_time             DATE;
    l_end_time               DATE;
    x_failed_row_id_list     VARCHAR2(10000);
```

```
    x_return_status          VARCHAR2(1);
    x_errorcode              NUMBER;
    x_msg_count              NUMBER;
    x_msg_data               VARCHAR2(1000);

BEGIN

    ------------------------------------------------
    --  Initialize variables for this sample run    --
    ------------------------------------------------

    l_org_profile_id    := 204;
    l_application_id    := 222;

    l_attr_group_type   := 'HZ_ORG_PROFILES_GROUP';

    l_attr_group1_name  := 'TCA_SAMPLE_GROUP';

    l_attr_group2_name  := 'TCA_SAMPLE_GROUP2';

    ------------------------------------------------------
    -- Create three row objects. (If any attribute       --
    -- groups were associated at the Revision level,     --
    -- pass the appropriate Revision ID as DATA_LEVEL_1) --
    ------------------------------------------------------
    l_user_attr_row_table :=
      EGO_USER_ATTR_ROW_TABLE(
        EGO_USER_ATTR_ROW_OBJ(
          1                                     --ROW_IDENTIFIER -
identifies the row number within the table
         ,null                                  --ATTR_GROUP_ID
         ,l_application_id               --ATTR_GROUP_APP_ID
         ,l_attr_group_type           --ATTR_GROUP_TYPE
         ,l_attr_group1_name       --ATTR_GROUP_NAME
         ,null                                  --DATA_LEVEL_1
         ,null                                  --DATA_LEVEL_2
         ,null                                  --DATA_LEVEL_3
         ,EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE) --TRANSACTION_TYPE
( This control the Mode(CREATE/UPDATE/DELETE)
       ,EGO_USER_ATTR_ROW_OBJ(
         2
        ,null
        ,l_application_id
        ,l_attr_group_type
        ,l_attr_group2_name
        ,null
        ,null
        ,null
        ,EGO_USER_ATTRS_DATA_PVT.G_SYNC_MODE)
      );

    -- NOTE: OTHER ALLOWED MODES
    ------------------------------
    -- G_CREATE_MODE
    -- G_UPDATE_MODE
    -- G_DELETE_MODE
    -- G_SYNC_MODE
```

```
    -------------------------------------------------------
    -- Create a data object for each attribute value to --
    -- process in each row.                              --
    -------------------------------------------------------
    l_user_attr_data_table :=
      EGO_USER_ATTR_DATA_TABLE(
        EGO_USER_ATTR_DATA_OBJ(
           1                                --ROW_IDENTIFIER
          ,'Attribute1'                  --ATTR_NAME      ( Interna
l Name )
          ,'Test Data Updated'   --ATTR_VALUE_STR ( Value for Attr
ibute1 )
          ,null                            --ATTR_VALUE_NUM
          ,null                            --ATTR_VALUE_DATE
          ,null                            --ATTR_DISP_VALUE
          ,null                            --ATTR_UNIT_OF_MEASURE
          ,null                            --USER_ROW_IDENTIFIER
        ),
        EGO_USER_ATTR_DATA_OBJ(
           1
          ,'Attribute2'
          ,null
          ,100
          ,null
          ,null
          ,null
          ,null
        ),
        EGO_USER_ATTR_DATA_OBJ(
           2
          ,'Attribute1'
          ,'ACME Materials UPDATED'
          ,null
          ,null
          ,null
          ,null
          ,null
        ),
        EGO_USER_ATTR_DATA_OBJ(
           2
          ,'Attribute2'
          ,'Test Data for Attribute group 2'
          ,null
          ,null
          ,null
          ,null
          ,null
        )
      );

    l_start_time := SYSDATE;

    ------------------------------------------------------------
    -- In the following call, several parameters were left    --
    -- to be defaulted; for more details and the complete     --
    -- parameter list consult the EGO_ITEM_PUB documentation  --
    ------------------------------------------------------------
    HZ_EXTENSIBILITY_PUB.Process_Organization_Record
```

```
    (
      p_api_version                   => 1.0
     ,p_org_profile_id               => l_org_profile_id
     ,p_attributes_row_table     => l_user_attr_row_table
     ,p_attributes_data_table     => l_user_attr_data_table
     ,p_debug_level                  => 3
     ,p_commit                        => FND_API.G_TRUE
     ,x_failed_row_id_list         => x_failed_row_id_list
     ,x_return_status                => x_return_status
     ,x_errorcode                    => x_errorcode
     ,x_msg_count                    => x_msg_count
     ,x_msg_data                     => x_msg_data
    );

    l_end_time := SYSDATE;

    DBMS_OUTPUT.Put_Line('After Process_User_Attrs_For_Item, x_ret
urn_status is:
'||x_return_status||', x_msg_count is: '||x_msg_count||' and x_msg
_data is: '||x_msg_data);


    IF (LENGTH(x_failed_row_id_list) > 0) THEN
      DBMS_OUTPUT.Put_Line('List of rows that failed: '||x_failed_
row_id_list);

      DECLARE

        l_errors_tbl         ERROR_HANDLER.Error_Tbl_Type;

      BEGIN

        ERROR_HANDLER.Get_Message_List(l_errors_tbl);

        FOR i IN 1..l_errors_tbl.COUNT
        LOOP
          DBMS_OUTPUT.Put_Line('Message:   '||l_errors_tbl(i).mess
age_text);
          DBMS_OUTPUT.Put_Line('Msg Type:  '||l_errors_tbl(i).mess
age_type);
        END LOOP;

      END;

    END IF;

    DBMS_OUTPUT.Put_Line('========== Performance ==========');
    DBMS_OUTPUT.Put_Line('The call started at:
         '||TO_CHAR(l_start_time,EGO_USER_ATTRS_COMMON_PVT.G_DATE_
FORMAT));
    DBMS_OUTPUT.Put_Line('The call ended at:      '||TO_CHAR(l_end
_time,
        EGO_USER_ATTRS_COMMON_PVT.G_DATE_FORMAT));
    DBMS_OUTPUT.Put_Line('Elapsed time in seconds: '||TO_CHAR((l_e
nd_time -
        l_start_time)*86400));

END;
```

Run the following query and verify the data.

```
SELECT * FROM HZ_ORG_PROFILES_EXT_VL WHERE ORGANIZATION_PROFILE_I
D = :1
```

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_ACCO
UNTS_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(
1,
'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_ACCOU
NT_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 TCA API Messages

This appendix provides the number, code, and text of the Oracle Trading Community Architecture API messages.

This appendix covers the following topics:

- Messages for the TCA APIs

Messages for the TCA APIs

This appendix describes messages that may be raised during the processing of API calls.

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.

Number	Code	Message
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.
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

Number	Code	Message
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 &
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	At least one of the phrases (codes) is already defined with the same subject and object type combination in this relationship type. Please use different phrases, or define the phrase pair with another relationship type.
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.
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.

Number	Code	Message
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.
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_MISSMATCH	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.
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.

Number	Code	Message
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.
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.
294290	HZ_DQM_NO_SCORE_DETAILS	Score details for this party is not available. Please execute the search again in the same database session.

Number	Code	Message
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.
294340	HZ_CREATED_BY_MISMATCH	The following information for this person comes from Oracle Human Resources and can only be updated in that application. Please contact a user who has permission to update human resources records.&COLUMN.
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.

Number	Code	Message
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.
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.
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.

Number	Code	Message
294531	HZ_API_CLASS_REL_OVERLAP	A class code relationship already exists between these class codes, with a date range that overlaps the range of the relationship that you are creating or updating. Please adjust the date range of either relationship so that they do not overlap.
294532	HZ_API_INVALID_COMBINATION1	You cannot provide a value for the &COLUMN2 column because no value is provided for the &COLUMN1 column.
294533	HZ_API_INVALID_COMBINATION2	You must provide a value for either the &COLUMN1 or &COLUMN2 column, but not for both.
294534	HZ_API_NOT_PRIMARY_KEY	The &COLUMN column is not a primary key of the &ENTITY table.
294535	HZ_API_PARTY_NOT_ORG	In the &TABLE_NAME table, the party that the &PARTY_ID_COL column references must exist and be of type Organization.
294536	HZ_API_PARTY_NOT_PERSON	In the &TABLE_NAME table, the party that the &PARTY_ID_COL column references must exist and be of type Person.
294537	HZ_API_INVALID_PRIMARY_KEY	The value entered is not a valid primary key of the entity being classified.
294538	HZ_API_DUP_CREDIT_RATING_REC	A credit rating already exists for this party, as of the same date and by the same rating organization.
294539	HZ_API_DUP_FIN_REPORT_REC	A financial report already exists for this party with the same report type and document reference, and for the same time period.
294540	HZ_API_DUP_FIN_NUMBER_REC	A financial number already exists with the same name for this financial report.
294541	HZ_API_INVALID_OBJ_NAME	The object name must exist in the FND_OBJECTS table.
	HZ_API_PRI_CODE_OVERLAP	You cannot set this class code as primary because its date range overlaps with an existing primary class code in the same class category.
	HZ_PARTY_SITE_PREFER_OVERLAP	You cannot create this contact preference because the date range overlaps the range of an existing contact preference for this party site.
	HZ_PARTY_PREFERENCE_OVERLAP	You cannot create this contact preference because the date range overlaps the range of an existing contact preference for this party.
	HZ_EDI_EFT_ORG_PARTIES_ONLY	You can only assign EDI and EFT contact points to Organization-type parties.
	HZ_API_NO_ACTIVE_RECORD	There are no active customer account relationships between accounts &ACCOUNT1 and & ACCOUNT2.

Number	Code	Message
	HZ_API_GLOBAL_LOC_NUM_ERRORS	The Global Location Number must have 13 digits, be a numeric value, and pass the Check Digit validation.
	HZ_ACTIVE_CUST_ACCT_RELATE	You cannot activate this relationship because an active customer account relationship already exists between accounts &ACCOUNT1 and &ACCOUNT2.

C

API Mapping

This appendix provides information about mapping the first, internal-only version of the Oracle Trading Community Architecture APIs to the second, public version of the APIs.

This appendix covers the following topics:

- Mapping Version 1 to Version 2 APIs

Mapping Version 1 to Version 2 APIs

This appendix provides mapping from the first, internal-only version of the API to the second, public version of the TCA API.

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 All procedure names identical to Version 1	Account, account relate	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

Version 1 API	Version 2 API	Comments	File Name
create/update cust_ account_role	HZ_CUST_ACCOUNT_ ROLE_V2PUB	Account role, role responsibility	ARH2CRSS/B.pls
create/update role_ responsibility	create/update cust_ account_role		
	create/update role_ responsibility		
	HZ_CUSTOMER_PROF ILE_V2PUB	Customer profile, profile amt	ARH2CFSS/B.pls
HZ_LOCATION_PUB	HZ_LOCATION_V2PUB		ARH2LOSS/B.pls
create/update location	create/update location		
HZ_PARTY_PUB	HZ_PARTY_V2PUB	Person, group, organization	ARH2PASS/B.pls
create/update person			
create/update organization			
create/update group			
create/update party site	HZ_PARTY_SITE_V2PUB	Site, site uses	ARH2PSSS/B.pls
create/update party site use			
create update org_contact	HZ_PARTY_CONTACT_ V2PUB	Contact, contact role	ARH2PCSS/B.pls
create/update org contact roles			
HZ_PER_INFO_PUB.	HZ_PERSON_INFO_V2 PUB.		ARH2PISS/B.pls
Create/update person_ language	Create/update person_ language		
HZ_RELATIONSHIP_PUB	HZ_RELATIONSHIP_V2 PUB		ARH2RESS/B.pls
HZ_RELATIONSHIP_TY PE_PUB	HZ_RELATIONSHIP_TY PE_V2PUB		ARH2RTSS/B.pls
HZ_TAX_ASSIGNMENT_ PUB	HZ_TAX_ASSIGNMENT_ V2PUB		ARH2TASS/B.pls
	HZ_CONTACT_ PREFERENCE_V2PUB		ARH2CTSS/B.pls
	HZ_REGISTRY_VAL IDATE_V2PUB	Public validation package for API in party level	ARH2RGVS/B.pls
	HZ_ACCOUNT_VAL IDATE_V2PUB	Public validation package for API in account level	ARH2ACVS/B.pls
	HZ_FORMAT_PHONE_V2 PUB	Phone Parsing and formatting	ARHPHFMS/B.pls