

**Oracle® Trading Community Architecture**  
Technical Implementation Guide  
Release 12.2  
**Part No. E48943-07**

August 2019

Oracle Trading Community Architecture Technical Implementation Guide, Release 12.2

Part No. E48943-07

Copyright © 2018, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Laurel Dale

Contributor: Ajai Singh, Amy Wu, Anish Stephen, Avinash Jha, Haikrishnan Radhakrishnan, Leela Krishna, Nishant Singhai, Ramanasudhir Gokavarapu, Shankar Bharadwaj

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

---

# Contents

## Send Us Your Comments

## Preface

### 1 Overview

Introduction.....	1-1
API Overview.....	1-2
Business Needs Met.....	1-3
Granular Versus Business Object APIs.....	1-3
Business Events.....	1-5

### 2 Granular API Features, Parameters, Attributes and Messages

Major Features and Solution Outline.....	2-1
Standard Parameters and Missing Attributes (PL/SQL API).....	2-4
Debug Messages (PL/SQL API).....	2-6
Standard Parameters and Missing Attributes (Java API).....	2-7
Debug Messages (Java API).....	2-10
Security of Oracle Human Resources Data.....	2-11
Third Party Data Integration.....	2-11

### 3 Party and Party Information API Use

Party APIs.....	3-1
Create Organization API.....	3-12
Update Organization API.....	3-28
Create Person API.....	3-41
Update Person API.....	3-49

Create Group API.....	3-58
Update Group API.....	3-62
<b>Party Information APIs.....</b>	<b>3-66</b>
Create Credit Ratings.....	3-68
Update Credit Ratings.....	3-75

## 4 Party Contact API Use

<b>Party Contact APIs.....</b>	<b>4-1</b>
Create Org Contact API.....	4-4
Update Org Contact API.....	4-15
Create Org Contact Role API.....	4-26
Update Org Contact Role API.....	4-29

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

<b>Person Information APIs.....</b>	<b>5-1</b>
Create Person Language API.....	5-3
Update Person Language API.....	5-6
Create Citizenship API.....	5-8
Update Citizenship API.....	5-9
Create Education API.....	5-10
Update Education API.....	5-12
Create Employment History API.....	5-13
Update Employment History API.....	5-16
Create Work Classes API.....	5-18
Update Work Classes API.....	5-19
Create Person Interest API.....	5-20
Update Person Interest API.....	5-22
<b>Organization Information APIs.....</b>	<b>5-23</b>
Create Financial Report API.....	5-24
Update Financial Report API.....	5-26
Create Financial Number API.....	5-29
Update Financial Number API.....	5-30
<b>Location APIs.....</b>	<b>5-31</b>
Create Location API.....	5-35
Update Location API.....	5-40
<b>Party Site APIs.....</b>	<b>5-46</b>
Create Party Site API.....	5-49
Update Party Site API.....	5-54
Create Party Site Use API.....	5-58
Update Party Site Use API.....	5-60

## **6 Contact Point API Use**

<b>Contact Point APIs.....</b>	<b>6-1</b>
Create Contact Point API.....	6-6
Update Contact Point API.....	6-17

## **7 Relationship and Hierarchy API Use**

<b>Relationship Type APIs.....</b>	<b>7-1</b>
Create Relationship Type API.....	7-2
Update Relationship Type API.....	7-8
<b>Relationship APIs.....</b>	<b>7-11</b>
Create Relationship API.....	7-15
Update Relationship API.....	7-27
<b>Hierarchy Retrieval APIs.....</b>	<b>7-37</b>
Top Parent Check API.....	7-40
Parent Child Relationship Check API.....	7-40
Get Parent Nodes API.....	7-41
Get Child Nodes API.....	7-42
Get Top Parent Nodes API.....	7-43

## **8 Classification API Use**

<b>Classification APIs.....</b>	<b>8-1</b>
Create Class Category API.....	8-3
Update Class Category API.....	8-5
Create Class Code Relation API.....	8-7
Update Class Code Relation API.....	8-9
Create Code Assignment API.....	8-11
Update Code Assignment API.....	8-15
Create Class Category Use API.....	8-17
Update Class Category Use API.....	8-19
Is Valid Category API.....	8-20

## **9 Contact Preference API Use**

<b>Contact Preference APIs.....</b>	<b>9-1</b>
Create Contact Preference API.....	9-2
Update Contact Preference API.....	9-10

## **10 Customer Account API Use**

<b>Customer Account APIs.....</b>	<b>10-1</b>
-----------------------------------	-------------

Create Customer Account API (Person or Organization).....	10-7
Update Customer Account API.....	10-20
Create Customer Account Relationship API.....	10-30
Update Customer Account Relationship API.....	10-33
<b>11 Customer Account Site API Use</b>	
<b>Customer Account Site APIs.....</b>	<b>11-1</b>
Create Customer Account Site API.....	11-9
Update Customer Account Site API.....	11-14
Create Customer Account Site Use API.....	11-20
Update Customer Account Site Use API.....	11-29
<b>12 Customer Profile and Customer Account Role API Use</b>	
<b>Customer Profile APIs.....</b>	<b>12-1</b>
Create Customer Profile API.....	12-9
Update Customer Profile API.....	12-22
Create Customer Profile Amount API.....	12-31
Update Customer Profile Amount API.....	12-38
<b>Customer Account Role APIs.....</b>	<b>12-44</b>
Create Customer Account Role API.....	12-48
Update Customer Account Role API.....	12-53
Create Role Responsibility API.....	12-57
Update Role Responsibility API.....	12-60
<b>13 Source System Management API Use</b>	
<b>Source System Management APIs.....</b>	<b>13-1</b>
Create Source System Reference API.....	13-3
Update Source System Reference API.....	13-7
Remap Internal Identifier API.....	13-10
<b>14 Bulk Import API Use</b>	
<b>Bulk Import APIs.....</b>	<b>14-1</b>
Create Import Batch API.....	14-1
Activate Import Batch API.....	14-3
<b>15 Address and Tax Validation API Use</b>	
<b>Location Service APIs.....</b>	<b>15-1</b>
Address Validation API.....	15-1
<b>Adapter APIs.....</b>	<b>15-2</b>

Create Adapter API.....	15-3
Update Adapter API.....	15-5
Create Adapter Territory API.....	15-7
Update Adapter Territory API.....	15-8
<b>Tax Assignment APIs.....</b>	<b>15-9</b>
Create Location Assignment API.....	15-9
Update Location Assignment API.....	15-11

## 16 Data Quality Management API Use

<b>Data Quality Management Search and Duplicate Identification APIs.....</b>	<b>16-1</b>
Find Parties API.....	16-1
Find Persons API.....	16-4
Find Parties and Details API.....	16-7
Find Party Sites API.....	16-10
Find Contacts API.....	16-12
Find Contact Points API.....	16-15
Identify Duplicate Parties API.....	16-18
Identify Duplicate Party Sites API.....	16-20
Identify Duplicate Contacts API.....	16-23
Identify Duplicate Contact Points API.....	16-25
Get Score Details API.....	16-28
Call API Dynamic API.....	16-30
Call API Dynamic Names API.....	16-35
<b>Data Quality Management Availability APIs.....</b>	<b>16-40</b>
Is DQM Match Rule Available API.....	16-40
Is DQM Index Available API.....	16-40
<b>Data Quality Management Transformation APIs.....</b>	<b>16-41</b>
Replace Word API.....	16-41
<b>Find Parties Object API.....</b>	<b>16-42</b>
<b>Data Quality Management Merge APIs.....</b>	<b>16-43</b>
Create Merge Request API.....	16-44
Get Party Merge Details API.....	16-47
Create Account Merge Request API.....	16-48
Get Account Merge Details API.....	16-50

## 17 Extensions API Use

<b>Extensions APIs.....</b>	<b>17-1</b>
Create or Update Organization Profile Extension.....	17-1
Create or Update Person Profile Extension.....	17-6
Create or Update Location Extension.....	17-11

Create or Update Party Site Extension.....	17-16
<b>18 Formatting API Use</b>	
<b>Phone Parsing and Formatting APIs.....</b>	<b>18-1</b>
Phone Number Parsing API.....	18-1
Phone Number Formatting API.....	18-3
Check Mobile Phone Number API.....	18-5
<b>Name and Address Formatting APIs.....</b>	<b>18-5</b>
Name Formatting API.....	18-6
Address Formatting API.....	18-12
<b>General Data Formatting API.....</b>	<b>18-18</b>
<b>19 Sample Granular API Code</b>	
<b>Sample Code - Demonstrating the APIs.....</b>	<b>19-1</b>
Create an Organization.....	19-2
Create a Location.....	19-5
Create a Party Site.....	19-7
Create a Party Site Use.....	19-10
Create a Contact Point (Phone).....	19-13
Create an Organization Contact.....	19-17
Update a Location Record.....	19-22
Create a Customer Account.....	19-27
Create a Customer Account Site.....	19-30
Create a Customer Account Site Use.....	19-33
Update a Customer Account Record.....	19-36
Create an Organization Profile Extensions Record.....	19-39
<b>20 Business Objects</b>	
<b>Business Objects Overview.....</b>	<b>20-2</b>
<b>Customer Account Business Object.....</b>	<b>20-6</b>
<b>Customer Account Contact Business Object.....</b>	<b>20-11</b>
<b>Customer Account Site Business Object.....</b>	<b>20-14</b>
<b>EDI Business Object.....</b>	<b>20-16</b>
<b>EFT Business Object.....</b>	<b>20-19</b>
<b>E-Mail Business Object.....</b>	<b>20-21</b>
<b>Organization Business Object.....</b>	<b>20-23</b>
<b>Organization Contact Business Object.....</b>	<b>20-31</b>
<b>Organization Customer Business Object.....</b>	<b>20-34</b>
<b>Party Site Business Object.....</b>	<b>20-35</b>
<b>Person Business Object.....</b>	<b>20-38</b>

Person Customer Business Object.....	20-43
Phone Business Object.....	20-44
SMS Business Object.....	20-46
Telex Business Object.....	20-49
Web Business Object.....	20-51
Contact Person Information Business Structure.....	20-53
Customer Account Site Use Business Structure.....	20-56
Customer Profile Business Structure.....	20-61
Employment History Business Structure.....	20-65
Financial Report Business Structure.....	20-68
Location Business Structure.....	20-70

## **21 Business Objects V2**

Business Objects V2 Overview.....	21-1
Customer Account V2 Business Object.....	21-1
Customer Account Site Use V2 Business Object.....	21-2
Customer Account Site V2 Business Object .....	21-2
Organization Customer V2 Business Object.....	21-2
Person Customer V2 Business Object.....	21-3

## **22 Business Object API Features and Procedures**

Business Object APIs Overview.....	22-1
Seeded Business Object APIs.....	22-2
Source System Management Business Object API.....	22-3
Business Object API Features.....	22-5
Create Business Object API Procedures.....	22-6
Update Business Object API Procedures.....	22-9
Save Business Object API Procedures.....	22-11
Get Business Object API Procedures.....	22-13
Event Independent Get Procedures.....	22-13
Event Dependent Get Procedures.....	22-14

## **23 Business Object V2 API Procedures**

Customer Account V2 Business Object.....	23-1
Create Customer Account V2 Business Object API.....	23-1
Update Customer Account V2 Business Object API.....	23-2
Save Customer Account V2 Business Object API.....	23-2
Get Customer Account V2 Business Object API.....	23-2
Customer Account Site V2 Business Object API.....	23-3
Create Customer Account Site V2 Business Object API.....	23-3

Update Customer Account Site V2 Business Object API.....	23-3
Save Customer Account Site V2 Business Object API.....	23-3
Get Customer Account Site V2 Business Object API.....	23-4
<b>Organization Customer V2 Business Object API.....</b>	<b>23-4</b>
Create Organization Customer V2 Business Object API.....	23-4
Update Organization Customer V2 Business Object API.....	23-4
Save Organization Customer V2 Business Object API.....	23-4
Get Organization Customer V2 Business Object API.....	23-5
<b>Person Customer V2 Business Object API.....</b>	<b>23-5</b>
Create Person Customer V2 Business Object API.....	23-5
Update Person Customer V2 Business Object API.....	23-6
Save Person Customer V2 Business Object API.....	23-6
Get Person Customer V2 Business Object API.....	23-6

## 24 Business Object API Attributes Information

<b>Business Object API Attributes Information Overview.....</b>	<b>24-2</b>
<b>Customer Account Business Object API.....</b>	<b>24-3</b>
Create Customer Account Business Object.....	24-3
Update Customer Account Business Object.....	24-11
Get Customer Account Business Object.....	24-20
<b>Customer Account Contact Business Object API.....</b>	<b>24-25</b>
Create Customer Account Contact Business Object.....	24-25
Update Customer Account Contact Business Object.....	24-28
Get Customer Account Contact Business Object.....	24-31
<b>Customer Account Site Business Object API.....</b>	<b>24-33</b>
Create Customer Account Site Business Object.....	24-34
Update Customer Account Site Business Object.....	24-37
Get Customer Account Site Business Object.....	24-40
<b>EDI Business Object API.....</b>	<b>24-42</b>
Create EDI Business Object.....	24-43
Update EDI Business Object.....	24-46
Get EDI Business Object.....	24-49
<b>EFT Business Object API.....</b>	<b>24-51</b>
Create EFT Business Object.....	24-52
Update EFT Business Object.....	24-54
Get EFT Business Object.....	24-57
<b>E-Mail Business Object API.....</b>	<b>24-59</b>
Create E-Mail Business Object.....	24-60
Update E-Mail Business Object.....	24-62
Get E-Mail Business Object.....	24-65

<b>Organization Business Object API.....</b>	24-67
Create Organization Business Object.....	24-68
Update Organization Business Object.....	24-78
Get Organization Business Object.....	24-89
<b>Organization Contact Business Object API.....</b>	24-96
Create Organization Contact Business Object.....	24-97
Update Organization Contact Business Object.....	24-101
Get Organization Contact Business Object.....	24-105
<b>Organization Customer Business Object API.....</b>	24-108
Create Organization Customer Business Object.....	24-109
Update Organization Customer Business Object.....	24-109
Get Organization Customer Business Object.....	24-110
<b>Party Site Business Object API.....</b>	24-111
Create Party Site Business Object.....	24-112
Update Party Site Business Object.....	24-115
Get Party Site Business Object.....	24-118
<b>Person Business Object API.....</b>	24-121
Create Person Business Object.....	24-122
Update Person Business Object.....	24-127
Get Person Business Object.....	24-132
<b>Person Customer Business Object API.....</b>	24-136
Create Person Customer Business Object.....	24-137
Update Person Customer Business Object.....	24-137
Get Person Customer Business Object.....	24-138
<b>Phone Business Object API.....</b>	24-139
Create Phone Business Object.....	24-140
Update Phone Business Object.....	24-143
Get Phone Business Object.....	24-146
<b>SMS Business Object API.....</b>	24-148
Create SMS Business Object.....	24-149
Update SMS Business Object.....	24-152
Get SMS Business Object.....	24-156
<b>Telex Business Object API.....</b>	24-158
Create Telex Business Object.....	24-159
Update Telex Business Object.....	24-161
Get Telex Business Object.....	24-164
<b>Web Business Object API.....</b>	24-166
Create Web Business Object.....	24-166
Update Web Business Object.....	24-169
Get Web Business Object.....	24-172
<b>Business Structure Attributes.....</b>	24-174

Contact Person Information Business Structure Attributes.....	24-174
Customer Account Site Use Business Structure Attributes.....	24-185
Customer Profile Business Structure Attributes.....	24-200
Employment History Business Structure Attributes.....	24-220
Financial Report Business Structure Attributes.....	24-228
Location Business Structure Attributes.....	24-237
<b>Granular Entity Attributes.....</b>	<b>24-245</b>
Bank Account Use Entity Attributes.....	24-247
Certification Entity Attributes.....	24-251
Citizenship Entity Attributes.....	24-254
Classification Entity Attributes.....	24-259
Contact Preference Entity Attributes.....	24-264
Credit Rating Entity Attributes.....	24-276
Customer Account Relationship Entity Attributes.....	24-293
Customer Profile Amount Entity Attributes.....	24-298
Education Entity Attributes.....	24-306
Extension Entity Attributes.....	24-311
Financial Number Entity Attributes.....	24-315
Financial Profile Entity Attributes.....	24-318
Organization Contact Role Entity Attributes.....	24-323
Party Preference Entity Attributes.....	24-327
Party Site Use Entity Attributes.....	24-331
Payment Method Entity Attributes.....	24-335
Person Interest Entity Attributes.....	24-339
Person Language Entity Attributes.....	24-343
Relationship Entity Attributes.....	24-348
Account Contact Role Entity Attributes.....	24-354
Source System Information Entity Attributes.....	24-358
Work Class Entity Attributes.....	24-359

## 25 Business Events

<b>Overview.....</b>	<b>25-1</b>
<b>Installation and Setup.....</b>	<b>25-1</b>
<b>Trading Community Architecture Granular Events.....</b>	<b>25-3</b>
<b>Trading Community Architecture Business Object Events.....</b>	<b>25-15</b>

## 26 Resource Manager APIs Overview and Features

<b>Resource Manager APIs.....</b>	<b>26-1</b>
<b>Private APIs .....</b>	<b>26-1</b>
<b>Public APIs.....</b>	<b>26-2</b>

<b>Public, published APIs .....</b>	<b>26-2</b>
Parameter Specifications.....	26-3
Standard IN Parameters.....	26-3
Standard OUT Parameters.....	26-4
Parameter Size.....	26-5
Missing Parameter Attributes.....	26-5
Parameter Validations.....	26-6
Invalid Parameters.....	26-6
Version Information.....	26-6
Status Messages.....	26-7

## **27 Resource Manager Public APIs**

<b>Resource Manager Public Packages.....</b>	<b>27-2</b>
<b>Resource Manager Public APIs.....</b>	<b>27-2</b>
<b>Package JTF_RS_RESOURCE_PUB.....</b>	<b>27-3</b>
<b>Create_Resource .....</b>	<b>27-3</b>
<b>Update_Resource.....</b>	<b>27-20</b>
<b>Package JTF_RS_GROUPS_PUB.....</b>	<b>27-35</b>
<b>Create_Resource_Group.....</b>	<b>27-35</b>
<b>Update_Resource_Group.....</b>	<b>27-41</b>
<b>Package JTF_RS_SALESREPS_PUB.....</b>	<b>27-46</b>
<b>Create_SalesRep.....</b>	<b>27-46</b>
<b>Update_SalesRep.....</b>	<b>27-52</b>
<b>Messages and Notifications.....</b>	<b>27-56</b>
<b>JTF_RS_RESOURCE_PUB.....</b>	<b>27-57</b>
<b>JTF_RS_GROUPS_PUB.....</b>	<b>27-66</b>
<b>JTF_RS_SALESREPS_PUB.....</b>	<b>27-70</b>
<b>Sample Code.....</b>	<b>27-74</b>
<b>JTF_RS_RESOURCE_PUB.....</b>	<b>27-74</b>
<b>JTF_RS_GROUPS_PUB.....</b>	<b>27-82</b>
<b>JTF_RS_SALESREPS_PUB.....</b>	<b>27-88</b>

## **Index**



---

## Send Us Your Comments

**Oracle Trading Community Architecture Technical Implementation Guide, Release 12.2**

**Part No. E48943-07**

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and [www.oracle.com](http://www.oracle.com). It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: [appsdoc\\_us@oracle.com](mailto:appsdoc_us@oracle.com)

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at [www.oracle.com](http://www.oracle.com).



---

# Preface

## Intended Audience

Welcome to Release 12.2 of the *Oracle Trading Community Architecture Technical Implementation Guide*.

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

- The principles and customary practices of your business area.
- Computer desktop application usage and terminology.

If you have never used Oracle E-Business Suite, we suggest you attend one or more of the Oracle E-Business training classes available through Oracle University.

See Related Information Sources on page xviii for more Oracle E-Business Suite product information.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Structure

### 1 Overview

- 2 Granular API Features, Parameters, Attributes and Messages**
- 3 Party and Party Information API Use**
- 4 Party Contact API Use**
- 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 Sample Granular API Code**
- 20 Business Objects**
- 21 Business Objects V2**
- 22 Business Object API Features and Procedures**
- 23 Business Object V2 API Procedures**
- 24 Business Object API Attributes Information**
- 25 Business Events**
- 26 Resource Manager APIs Overview and Features**
- 27 Resource Manager Public APIs**

## Related Information Sources

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

## Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the Oracle E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

## Online Documentation

All Oracle E-Business Suite documentation is available online (HTML or PDF).

- **PDF** - See the Oracle E-Business Suite Documentation Library for current PDF documentation for your product with each release. The Oracle E-Business Suite Documentation Library is also available on My Oracle Support and is updated frequently
- **Online Help** - Online help patches (HTML) are available on My Oracle Support.
- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.
- **Oracle Electronic Technical Reference Manual** - The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available on My Oracle Support.

## Guides Related to All Products

### Oracle E-Business Suite User's Guide

This guide explains how to navigate, enter and query data, and run concurrent requests using the user interface (UI) of Oracle E-Business Suite. It includes information on setting preferences and customizing the UI. In addition, this guide describes accessibility features and keyboard shortcuts for Oracle E-Business Suite.

## Guides Related to This Product

### Oracle Common Application Calendar Implementation Guide

This guide describes how to define tasks and note types, set up task statuses and status transition rules, define task priorities, set up data security, and map notes and references to source objects such as a sales lead to Task Manager. In addition, it describes how to create users and run concurrent programs to retrieve new and updated tasks.

### Oracle Common Application Calendar User Guide

Oracle Common Application Calendar enables you to manage daily tasks and appointments, create and maintain notes, and schedule resources. It provides a central place to store and view resource schedules, utilization, and availability. It also enables you to synchronize your calendar with external calendars such as Microsoft Outlook or handheld devices.

## **Oracle Customer Data Librarian Implementation Guide**

Oracle Customer Data Librarian includes all of the functionality of Oracle Customers Online with the additional features of maintaining the quality of customer data. Therefore, you must first implement Oracle Customers Online fully. Then, use this guide to assign responsibilities and access to users and set the necessary profile options for data librarian deployment, data import, mapping, search and duplication removal, and data security.

## **Oracle Customers Online Implementation Guide**

This guide describes how to set up customer accounts, set up additional display attributes, set up data quality management, define the source systems for customer data and map customers to the source system. Before you can use Oracle Customers Online, you must implement Oracle Common Application Calendar, Oracle Customer Interaction History, and Oracle Trading Community Architecture. Oracle Customers Online features and data come from the marketing and sales applications and the additional applications of Oracle Order Management, Oracle Credit Management, and Oracle Receivables.

## **Oracle Customers Online User Guide**

Oracle Customers Online enables you to view, create, and maintain customer or party information, create customer relationships and hierarchies, manage tasks and employees for your organization, and use reports to view customer profile trends and data quality information. You can import customer data from external sources, and administer and control the usage of this data across the Oracle E-Business Suite.

## **Oracle Financials Concepts Guide**

This guide discusses the conceptual architecture of Oracle Financials. It introduces you to the financial concepts used in the application, and helps you compare real world business, organization, and processes to those used in the applications. Understanding the concepts enables you to exploit the features of the Oracle Financials suite of applications for better financial performance, reporting, control, compliance, and security.

## **Oracle Financials Implementation Guide**

This guide describes how to implement the Oracle Financials E-Business Suite. It takes you through the steps of setting up your organizations, including legal entities, and their accounting, using the Accounting Setup Manager. You can find information on intercompany accounting and sequencing of accounting entries with relevant examples.

## **Oracle General Ledger User's Guide**

This guide provides you with information on how to use Oracle General Ledger. Use this guide to learn how to create and maintain ledgers, ledger currencies, budgets, and

journal entries. This guide also includes information about running financial reports.

## Oracle HRMS Documentation Set

This set of guides explains how to define your employees, so you can give them operating unit and job assignments. It also explains how to set up an organization (operating unit). Even if you do not install Oracle HRMS, you can set up employees and organizations using Oracle HRMS windows. Specifically, the following manuals will help you set up employees and operating units:

- **Oracle HRMS Enterprise and Workforce Management Guide**

This user guide explains how to set up and use enterprise modeling, organization management, and cost analysis.

- **Oracle HRMS Workforce Sourcing, Deployment, and Talent Management Guide**

Use this guide to find out about setting up employees and managing your people resources.

## Oracle Inventory User's Guide

This guide enables you to configure the Oracle Inventory structure to best represent your company's inventory sites and business units after you have defined your required ledger and key flexfields. You can also learn about centralized and decentralized inventory structures, and controls and reference options for using and maintaining inventory items such as categories, commodity codes, attributes, statuses, relationships, and picking rules.

## Oracle Order Management User's Guide

This guide provides information on how to use Oracle Order Management. Use this guide to learn how to enter and update sales orders, maintain sales agreements, combine sales orders with procurement orders, and process orders. In addition, this guide describes how you can authorize and manage returns, schedule across orders, apply charges and discounts, enter shipping information for orders, raise order invoices and process invoices and payments.

## Oracle Purchasing User's Guide

This guide describes how to create and approve purchasing documents, including requisitions, different types of purchase orders, quotations, RFQs, and receipts. This guide also describes how to manage your supply base through agreements, sourcing rules, and approved supplier lists. In addition, this guide explains how you can automatically create purchasing documents based on business rules through integration with Oracle Workflow technology, which automates many of the key procurement processes.

## **Oracle Receivables Implementation Guide**

This guide provides you with information on how to implement Oracle Receivables for your business activities. It helps you to set up your accounting distributions, your accounting structure, and various rules used to process transactions for accounting, charges, payments, and collections. You can learn how to use descriptive flexfields, Receivables system options, lookups, and profiles options to customize application behavior and define comprehensive defaults that Receivables uses to make data entry more efficient and accurate.

## **Oracle Receivables Reference Guide**

This guide describes the APIs and open interfaces that Oracle Receivables provides. You can use these to extend Oracle Receivables functionality. For example, you can learn how to use AutoLockbox to create and apply receipts and AutoInvoice to import and validate transactions from other systems. You can also learn how to archive and purge Receivables data.

## **Oracle Receivables User Guide**

This guide provides you with information on how to use Oracle Receivables. Use this guide to learn how to create and maintain transactions and bills receivable, enter and apply receipts, enter customer information, and manage revenue. This guide also includes information about accounting in Receivables. Use the Standard Navigation Paths appendix to find out how to access each Receivables window.

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

This guide enables you to define entities in the TCA Registry, create relationships, search, prevent duplication, and control access. In addition, you can use this guide to define time zones and phone formats, configure adapters for the processing of data in the TCA Registry, define sources that provide data for specific entities, and create user-defined attributes to extend the registry. You can administer these TCA tools and features from the Administration tab using the Trading Community Manager responsibility. This tab is also available in Oracle Customers Online and Oracle Customer Data Librarian.

## **Oracle Trading Community Architecture Reference Guide**

This guide provides information including a comprehensive glossary to supplement the documentation for Oracle Trading Community Architecture and to help you understand products in the Oracle Customer Data Management family. It describes customer interface tables and the interface tables used for bulk import of data from external sources, and D&B data elements. In addition, you can learn about available relationship types, available replacement words and attributes for Data Quality Management data, available matching rules for various TCA administration tasks, and the results and impact of the party and account merge processes initiated in Oracle E-

Business Suite applications.

## Oracle Trading Community Architecture User Guide

Oracle Trading Community Architecture (TCA) maintains information including relationships about parties, customers, organizations, and locations that belong to your commercial community in the TCA Registry. This guide enables you to use the features and user interfaces provided by TCA and by other Oracle E-Business Suite applications to view, create, and update Registry information. For example, you can import batches of party data in bulk from external source systems into the TCA Registry, merge duplicate parties, sites, and customer accounts, generate time zones for phones and locations, and run various customer reports.

## Installation and System Administration

### Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle E-Business Suite data.

### Oracle E-Business Suite Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12.2, or contemplating significant changes to a configuration. After describing the Oracle E-Business Suite architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus the installation and configuration choices that may be available.

### Oracle E-Business Suite CRM System Administrator's Guide

This manual describes how to implement the CRM Technology Foundation (JTT) and use its System Administrator Console.

### Oracle E-Business Suite Developer's Guide

This guide contains the coding standards followed by the Oracle E-Business Suite development staff. It describes the Oracle Application Object Library components needed to implement the Oracle E-Business Suite user interface described in the *Oracle E-Business Suite User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle E-Business Suite. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

### Oracle E-Business Suite Installation Guide: Using Rapid Install

This book is intended for use by anyone who is responsible for installing or upgrading

Oracle E-Business Suite. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle E-Business Suite Release 12.2, or as part of an upgrade to Release 12.2.

### **Oracle E-Business Suite Maintenance Guide**

This guide contains information about the strategies, tasks, and troubleshooting activities that can be used to help ensure an Oracle E-Business Suite system keeps running smoothly, together with a comprehensive description of the relevant tools and utilities. It also describes how to patch a system, with recommendations for optimizing typical patching operations and reducing downtime.

### **Oracle E-Business Suite Security Guide**

This guide contains information on a comprehensive range of security-related topics, including access control, user management, function security, data security, and auditing. It also describes how Oracle E-Business Suite can be integrated into a single sign-on environment.

### **Oracle E-Business Suite Setup Guide**

This guide contains information on system configuration tasks that are carried out either after installation or whenever there is a significant change to the system. The activities described include defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help.

### **Oracle E-Business Suite User Interface Standards for Forms-Based Products**

This guide contains the user interface (UI) standards followed by the Oracle E-Business Suite development staff. It describes the UI for the Oracle E-Business Suite products and tells you how to apply this UI to the design of an application built by using Oracle Forms.

## **Other Implementation Documentation**

### **Oracle Approvals Management Implementation Guide**

This guide describes transaction attributes, conditions, actions, and approver groups that you can use to define approval rules for your business. These rules govern the process for approving transactions in an integrated Oracle application. You can define approvals by job, supervisor hierarchy, positions, or by lists of individuals created either at the time you set up the approval rule or generated dynamically when the rule is invoked. You can learn how to link different approval methods together and how to run approval processes in parallel to shorten transaction approval process time.

## **Oracle Diagnostics Framework User's Guide**

This guide contains information on implementing, administering, and developing diagnostics tests for Oracle E-Business Suite using the Oracle Diagnostics Framework.

## **Oracle E-Business Suite Flexfields Guide**

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

## **Oracle E-Business Suite Integrated SOA Gateway Implementation Guide**

This guide explains the details of how integration repository administrators can manage and administer the entire service enablement process based on the service-oriented architecture (SOA) for both native packaged public integration interfaces and composite services - BPEL type. It also describes how to invoke Web services from Oracle E-Business Suite by working with Oracle Workflow Business Event System, manage Web service security, and monitor SOAP messages.

## **Oracle E-Business Suite Integrated SOA Gateway User's Guide**

This guide describes how users can browse and view the integration interface definitions and services that reside in Oracle Integration Repository.

## **Oracle E-Business Suite Multiple Organizations Implementation Guide**

This guide describes how to set up multiple organizations and the relationships among them in a single installation of an Oracle E-Business Suite product such that transactions flow smoothly through and among organizations that can be ledgers, business groups, legal entities, operating units, or inventory organizations. You can use this guide to assign operating units to a security profile and assign this profile to responsibilities such that a user can access data for multiple operating units from a single responsibility. In addition, this guide describes how to set up reporting to generate reports at different levels and for different contexts. Reporting levels can be ledger or operating unit while reporting context is a named entity in the selected reporting level.

## **Oracle e-Commerce Gateway Implementation Guide**

This guide describes implementation details, highlighting additional setup steps needed for trading partners, code conversion, and Oracle E-Business Suite. It also provides architecture guidelines for transaction interface files, troubleshooting information, and a description of how to customize EDI transactions.

## **Oracle e-Commerce Gateway User's Guide**

This guide describes the functionality of Oracle e-Commerce Gateway and the

necessary setup steps in order for Oracle E-Business Suite to conduct business with trading partners through Electronic Data Interchange (EDI). It also describes how to run extract programs for outbound transactions, import programs for inbound transactions, and the relevant reports.

### **Oracle iSetup User's Guide**

This guide describes how to use Oracle iSetup to migrate data between different instances of the Oracle E-Business Suite and generate reports. It also includes configuration information, instance mapping, and seeded templates used for data migration.

### **Oracle Product Hub Implementation Guide**

This guide explains how to set up hierarchies of items using catalogs and catalog categories and then to create user-defined attributes to capture all of the detailed information (such as cost information) about an object (such as an item or change order). It also explains how to set up optional features used in specific business cases; choose which features meet your business' needs. Finally, the guide explains the set up steps required to link to third party and legacy applications, then synchronize and enrich the data in a master product information repository.

### **Oracle Product Hub User's Guide**

This guide explains how to centrally manage item information across an enterprise, focusing on product data consolidation and quality. The item information managed includes item attributes, categorization, organizations, suppliers, multilevel structures/bills of material, packaging, changes, attachments, and reporting.

### **Oracle Web Applications Desktop Integrator Implementation and Administration Guide**

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite functionality to a spreadsheet, where familiar data entry and modeling techniques can be used to complete Oracle E-Business Suite tasks. You can create formatted spreadsheets on your desktop that allow you to download, view, edit, and create Oracle E-Business Suite data, which you can then upload. This guide describes how to implement Oracle Web Applications Desktop Integrator and how to define mappings, layouts, style sheets, and other setup options.

### **Oracle Workflow Administrator's Guide**

This guide explains how to complete the setup steps necessary for any Oracle E-Business Suite product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

## **Oracle Workflow Developer's Guide**

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

## **Oracle Workflow User's Guide**

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

## **Oracle XML Gateway User's Guide**

This guide describes Oracle XML Gateway functionality and each component of the Oracle XML Gateway architecture, including Message Designer, Oracle XML Gateway Setup, Execution Engine, Message Queues, and Oracle Transport Agent. It also explains how to use Collaboration History that records all business transactions and messages exchanged with trading partners.

The integrations with Oracle Workflow Business Event System, and the Business-to-Business transactions are also addressed in this guide.

## **Oracle XML Publisher Administration and Developer's Guide**

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce outputs to meet a variety of business needs. Outputs include: PDF, HTML, Excel, RTF, and eText (for EDI and EFT transactions). Oracle XML Publisher can be used to generate reports based on existing Oracle E-Business Suite report data, or you can use Oracle XML Publisher's data extraction engine to build your own queries. Oracle XML Publisher also provides a robust set of APIs to manage delivery of your reports via e-mail, fax, secure FTP, printer, WebDav, and more. This guide describes how to set up and administer Oracle XML Publisher as well as how to use the Application Programming Interface to build custom solutions. This guide is available through the Oracle E-Business Suite online help.

## **Oracle XML Publisher Report Designer's Guide**

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Using Microsoft Word or Adobe Acrobat as the design tool, you can create pixel-perfect reports from the Oracle E-Business Suite. Use this guide to design your report layouts. This guide is available through the Oracle E-Business Suite online help.

## **Training and Support**

### **Training**

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

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

### **Support**

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

## **Do Not Use Database Tools to Modify Oracle E-Business Suite Data**

Oracle STRONGLY RECOMMENDS that you never use SQL\*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite 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 E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, 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 E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite 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 covers the following topics:

- Introduction
- API Overview
- Business Events

### Introduction

This document describes the integration features that you can use to access and leverage the Oracle Trading Community Architecture (TCA) data model, which is the foundation for various applications in the Oracle E-Business Suite. Deploying companies, consultants, and Oracle internal development teams can all use these public objects.

- **Granular (V2, or Version 2) Application Programming Interfaces (APIs):** APIs that manipulate data for granular entities, which correspond to specific tables in the TCA data model.
- **Business Object APIs:** APIs that manipulate data for business objects, which are abstract groupings of granular entities that form operable, logical business units.
- **Business Events:** Events that signal the creation or update of data, and push data out to subscribing objects, based on triggers such as granular or business object API calls.

One major area to use these integration features is in a data hub, where data from various systems are connected and transacted in real-time via the TCA data model. Using APIs and business events, you can create or update in one system and ensure that the change is reflected in the other systems.

## Related Topics

API Overview, page 1-2

## API Overview

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 TCA 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

## Related Topics

[Business Needs Met, page 1-3](#)

[Granular Versus Business Object APIs, page 1-3](#)

[Introduction, page 1-1](#)

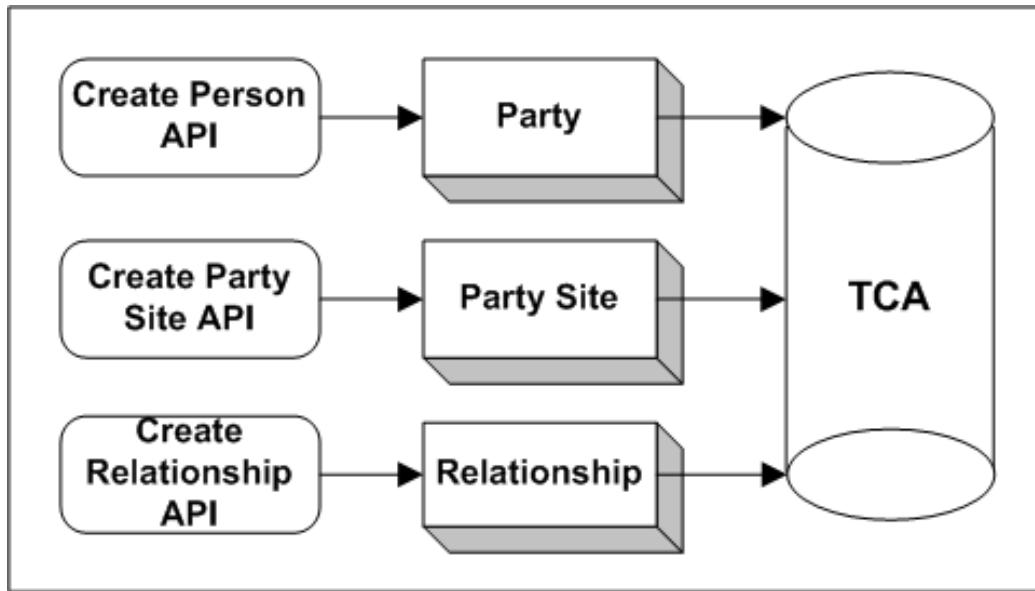
## Business Needs Met

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 D&B can be mass loaded into the TCA model by using the PL/SQL API. The D&B integration currently available with TCA also utilizes the TCA APIs.
- Data migration from legacy systems into the TCA model.
- Access to the TCA model from custom applications built by deploying companies and Oracle Consulting.

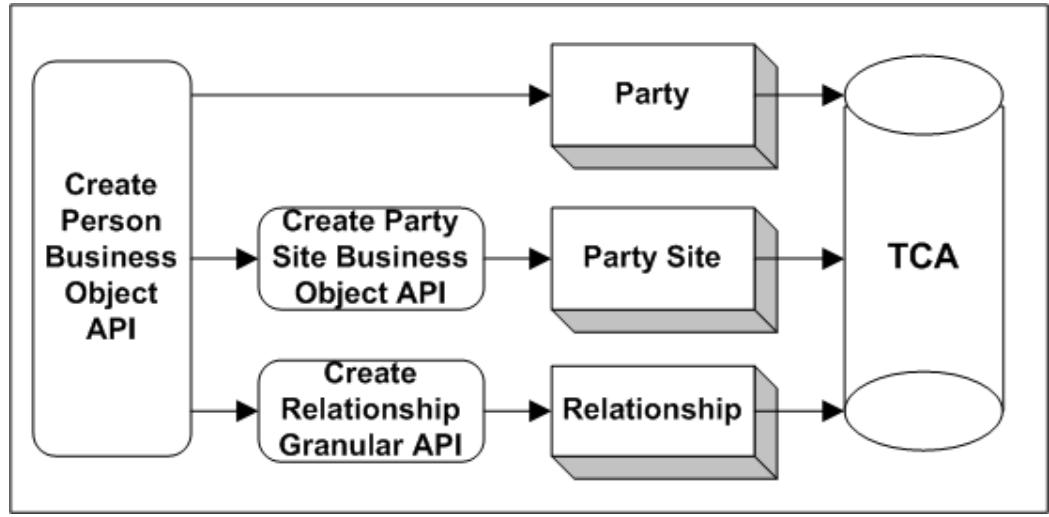
## Granular Versus Business Object APIs

Granular APIs manipulate data at the granular Oracle Trading Community Architecture entity level. For example, a person record consists of profile, address, and relationship information, so this record corresponds to the TCA Party, Party Site, and Relationship entities respectively. To create this person information, you need to use three granular API procedures to create a party of type person, party site, and relationship, as shown in this diagram.



Oracle Trading Community Architecture provides seeded business objects, which can contain other business objects, granular entities, or both. When a business object API is called, it can subsequently call granular APIs.

For example, the Person business object contains various business objects and granular entities, including the Party Site business object and Relationship entity. When the Create Person Business Object API procedure is called, it subsequently calls the Create Party Site Business Object procedure and the Create Relationship granular API procedure. So instead of three separate initial API calls, as is the case with the granular APIs, you can use just one business object API to create the same person information, as illustrated in this diagram.



## Related Topics

[API Overview, page 1-2](#)

## Business Events

TCA utilizes three concurrent programs to manage business events.

- TCA Business Object Events: Generate Infrastructure Packages Program
- TCA Business Object Events: Raise Events Program
- TCA Business Object Events: Cleanse Infrastructure Program



---

## Granular API Features, Parameters, Attributes and Messages

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 OracleMetaLink (<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.

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

## From PL/SQL code

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

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

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

## Standard Parameters and Missing Attributes (Java API)

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

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

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

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

## IN OUT/OUT Parameters

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

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

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

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

Java has no clear analogue to the concept of declaring parameters as OUT or IN OUT. Instead, in Java, there are certain types of objects that are immutable (changes by the

called procedure which cannot be seen by the caller) and other types which are mutable.

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

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

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

## Standard IN or IN/OUT Parameters

- OracleConnection \_connection Required

This parameter is for passing Oracle JDBC connection to APIs.

- BigDecimal [ ] p\_object\_version\_number Required

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

String p\_init\_msg\_list Optional

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

## Standard OUT Parameters

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

- String [ ] x\_return\_status

This returns the status of the API. The values returned in x\_return\_status[0] are one of the following:

- HzConstant.getGRetStsSuccess() - Success
- HzConstant.getGRetStsError() - Expected Error - validation or missing data.
- HzConstant.getGRetStsUnexpError() - Unexpected Error, not fixable by calling program.

- `BigDecimal [ ] x_msg_count`
- `String [ ] x_msg_data`  
`x_msg_count[0]` holds the number of messages in the message list. If the count is one, then `x_msg_data[0]` holds the decoded message.

## Nested Record Types

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

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

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

## Primary Key Generation

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

## Missing Attributes

Each record class has two constructors:

- `Public <entity>Rec();`

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

- `HzConstant.getGMissNum()` for `BigDecimal` type.  
The application returns `9.99E125`.
- `HzConstant.getGMissChar()` for `String` type.  
The application returns "`\0`".
- `HzConstant.getGMissDate()` for `java.sql.Timestamp` type.  
The application returns "`4713-01-01 00:00:00.0`"

- Public <entity>Rec(boolean \_\_RosettaUseGMISSValues);

This constructor provides two choices. Calling this constructor with a 'true' value corresponds to the first constructor. If we call the constructor by passing 'false', all BigDecimal, String and Timestamp attributes will be initialized to null values.

Similarly, this boolean parameter will be passed into included record classes if any is present.

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

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

## Debug Messages (Java API)

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

The profiles for controlling the debug mechanism are:

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

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

The value of the HZ\_API\_DEBUG\_FILE\_PATH profile specifies a directory file path that the database has write access to, as provided in init.ora. You can find this path

information by querying: select value from v\$parameter where the name equals 'utl\_file\_dir'.

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

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

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

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

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

## Security of Oracle Human Resources Data

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

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

## Third Party Data Integration

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

- Party API
  - Create Organization API
  - Update Organization API
  - Create Person API

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

## General Overview

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

From an API, you can populate the actual\_content\_source column to identify the third party data source. The content\_source\_type attribute is obsolete. You should use the actual\_content\_source attribute for third- party data integration applications. If necessary, you can use the content\_source\_type attribute to maintain backward compatibility.

Default values for the actual\_content\_source and content\_source\_type attributes are taken from the actual\_content\_source column, unless you have overwritten the default value in either column.

## Party API

In the Organization and Person record types; the values in the actual\_content\_source column default to SST and the values in the content\_source\_type column default to user\_entered.

## Create Organization API and Create Person API

These APIs first determine the data source based on the values in the actual\_content\_source and content\_source\_type columns. If the data source is either SST or user\_entered, the API errors out if the same party ID exists in the HZ\_PARTIES table. Otherwise, the API will create new party profiles and parties.

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

## **Update Organization API and Update Person API**

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

## **Location API**

The values in both the actual\_content\_source and content\_source\_type columns default to user\_entered. 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 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
    duns_number_c
    enquiry_duns
    ceo_name
    ceo_title
    principal_name
    principal_title
    legal_status
    control_yr
    employees_total
    hq_branch_ind
    branch_flag
    oob_ind
    line_of_business
    cong_dist_code
    sic_code
    import_ind
    export_ind
    labor_surplus_ind
    debarment_ind
    minority_owned_ind
    minority_owned_type
    woman_owned_ind
    disadv_8a_ind
    small_bus_ind
    rent_own_ind
    debarments_count
    debarments_date
    failure_score
    failure_score_natnl_percentile
    failure_score_override_code
    failure_score_commentary
    global_failure_score
    db_rating
    credit_score
    credit_score_commentary
    paydex_score
    paydex_three_months_ago
    paydex_norm
    best_time_contact_begin
    best_time_contact_end
    organization_name_phonetic
    tax_reference
    gsa_indicator_flag
    jgzz_fiscal_code
    analysis_fy
    fiscal_yearend_month
    curr_fy_potential_revenue
    next_fy_potential_revenue
    year_established
    mission_statement
    organization_type
    business_scope
    corporation_class
    known_as
    known_as2
    known_as3
    known_as4
    known_as5
    local_bus_iden_type
    local_bus_identifier
)

```

```

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_CONTENT_SOURCE_TYPE,
content_source_number             VARCHAR2( 30),
attribute_category               VARCHAR2(150),
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    NUMBER,
application_id       VARCHAR2(255),
do_not_confuse_with VARCHAR2(30) := G_SST_SOURCE_TYPE,
actual_content_source VARCHAR2(2),
home_country         PARTY_REC_TYPE:= G_MISS_PARTY _REC
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  
    VARCHAR2(60),  
    place_of_birth  
    DATE,  
    date_of_death  
    VARCHAR2(30),  
    gender  
    VARCHAR2(60),  
    declared_ethnicity  
    VARCHAR2(30),  
    marital_status  
    DATE,  
    marital_status_effective_date  
    NUMBER,  
    personal_income  
    head_of_household_flag  
    VARCHAR2(1),  
    household_income  
    NUMBER,  
    household_size  
    NUMBER,  
    rent_own_ind  
    VARCHAR2(30),  
    last_known_gps  
    VARCHAR2(60),  
    content_source_type  
    VARCHAR2(30) :=  
    G_MISS_CONTENT_SOURCE_TYPE,  
    internal_flag  
    VARCHAR2(2),  
    attribute_category  
    VARCHAR2(30),  
    attribute1  
    VARCHAR2(150),  
    attribute2  
    VARCHAR2(150),  
    attribute3  
    VARCHAR2(150),  
    attribute4  
    VARCHAR2(150),  
    attribute5  
    VARCHAR2(150),  
    attribute6  
    VARCHAR2(150),  
    attribute7  
    VARCHAR2(150),  
    attribute8  
    VARCHAR2(150),  
    attribute9  
    VARCHAR2(150),  
    attribute10  
    VARCHAR2(150),  
    attribute11  
    VARCHAR2(150),  
    attribute12  
    VARCHAR2(150),  
    attribute13  
    VARCHAR2(150),  
    attribute14  
    VARCHAR2(150),  
    attribute15  
    VARCHAR2(150),  
    attribute16  
    VARCHAR2(150),  
    attribute17  
    VARCHAR2(150),  
    attribute18  
    VARCHAR2(150),  
    attribute19  
    VARCHAR2(150),  
    attribute20  
    VARCHAR2(150),  
    created_by_module  
    VARCHAR2(150),
```

```

application_id          NUMBER,
actual_content_source  VARCHAR2(30) := 
G_SST_SOURCE_TYPE,
party_rec               PARTY_REC_TYPE:=
G_MISS_PARTY_REC
)

```

## PL/SQL Record Structure for Group

```

TYPE group_rec_type IS RECORD(
group_name           VARCHAR2(255),
group_type            VARCHAR2(30),
created_by_module    VARCHAR2(150),
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 attribute3;
public String attribute4;
public String attribute5;
public String attribute6;
public String attribute7;
public String attribute8;
public String attribute9;
public String attribute10;
public String attribute11;
public String attribute12;
public String attribute13;
public String attribute14;
public String attribute15;
public String attribute16;
public String attribute17;
public String attribute18;
public String attribute19;
public String attribute20;
public String attribute21;
public String attribute22;
public String attribute23;
public String attribute24;

public PartyRec();
public PartyRec(boolean _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 String control_yr;
    public String 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 String debarments_count;
    public String debarments_date;
    public String failure_score;
    public String 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 String best_time_contact_begin;
    public String 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 String curr_fy_potential_revenue;
    public String next_fy_potential_revenue;
    public String 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 Boolean public_private_ownership_flag;
public Boolean internal_flag;
public String local_activity_code_type;
public String local_activity_code;
public String emp_at_primary_addr;
public String emp_at_primary_addr_text;
public String emp_at_primary_addr_est_ind;
public String emp_at_primary_addr_min_ind;
public BigDecimal high_credit;
public BigDecimal avg_high_credit;
public BigDecimal total_payments;
public String credit_score_class;
public String credit_score_natl_percentile;
public String credit_score_incd_default;
public String credit_score_age;
public String 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 String failure_score_class;
public String failure_score_incd_default;
public String failure_score_age;
public String 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 String 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 created_by_module;
public String application_id;
public String do_not_confuse_with;
public String actual_content_source;
public String home_country;
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 String date_of_birth;  
    public String place_of_birth;  
    public String deceased_ind;  
    public String date_of_death;  
    public String gender;  
    public String declared_ethnicity;  
    public String marital_status;  
    public String marital_status_effective_date;  
    public String personal_income;  
    public String head_of_household_flag;  
    public String household_income;  
    public String 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 String 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\_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 create_organization (
    p_init_msg_list                      IN          VARCHAR2 := FND_API.
    G_FALSE,
    p_organization_rec                   IN          ORGANIZATION_REC_TYPE,
    x_return_status                      OUT         VARCHAR2,
    x_msg_count                          OUT         NUMBER,
    x_msg_data                           OUT         VARCHAR2,
    x_party_id                           OUT         NUMBER,
    x_party_number                       OUT         VARCHAR2,
    x_profile_id                         OUT         NUMBER
)
)
```

## Java Method

```
public static void createOrganization(
    OracleConnection connection,
    String p_init_msg_list,
    OrganizationRec p_organization_rec,
    String x_return_status,
    String x_msg_count,
    BigDecimal x_msg_data,
    String x_party_id,
    BigDecimal x_party_number,
    String 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	
ceo_name	IN	VARCHAR2	No	
ceo_title	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
principal_name	IN	VARCHAR R2	No	
principal_title	IN	VARCHAR R2	No	
legal_status	IN	VARCHAR R2	No	Validation: Validated against AR lookup type LEGAL_STATUS
control_yr	IN	NUMBER	No	
employees_total	IN	NUMBER	No	
hq_branch_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type HQ_BRANCH_IND
branch_flag	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
oob_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
line_of_business	IN	VARCHAR R2	No	
cong_dist_code	IN	VARCHAR R2	No	
sic_code	IN	VARCHAR R2	No	
import_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
export_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
labor_surplus_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
debarment_ind	IN	VARCHAR R2	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	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
minority_owned_type	IN	VARCHAR R2	No	
woman_owned_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
disadv_8a_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
small_bus_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
rent_own_ind	IN	VARCHAR R2	No	Validation: 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	VARCHAR R2	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_overrid_e_code	IN	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary	IN	VARCHAR R2	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
global_failure_score	IN	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
db_rating	IN	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score	IN	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary	IN	VARCHAR R2	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	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_three_months_ago	IN	VARCHAR R2	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_norm	IN	VARCHAR R2	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	VARCHAR R2	No	
tax_reference	IN	VARCHAR R2	No	
gsa_indicator_flag	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
jgzz_fiscal_code	IN	VARCHAR R2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
analysis_fy	IN	VARCHAR R2	No	
fiscal_yearend_month	IN	VARCHAR R2	No	Validation: Validated against AR lookup type MONTH
curr_fy_potential_rev_enue	IN	NUMBER	No	
next_fy_potential_rev_enue	IN	NUMBER	No	
year_established	IN	NUMBER	No	
mission_statement	IN	VARCHAR R2	No	
organization_type	IN	VARCHAR R2	No	
business_scope	IN	VARCHAR R2	No	
corporation_class	IN	VARCHAR R2	No	
known_as	IN	VARCHAR R2	No	
known_as2	IN	VARCHAR R2	No	
known_as3	IN	VARCHAR R2	No	
known_as4	IN	VARCHAR R2	No	
known_as5	IN	VARCHAR R2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
local_bus_iden_type	IN	VARCHAR R2	No	Validation: Validated against AR lookup type LOCAL_BUS_IDEN_TYPE
local_bus_identifier	IN	VARCHAR R2	No	
pref_functional_currency	IN	VARCHAR R2	No	
registration_type	IN	VARCHAR R2	No	Validation: Validated against AR lookup type REGISTRATION_TYPE
total_employees_text	IN	VARCHAR R2	No	
total_employees_ind	IN	VARCHAR R2	No	Validation: Validated against the TOTAL_EMPLOYEES_INDICATOR lookup type.
total_emp_est_ind	IN	VARCHAR R2	No	Validation: Validated against the TOTAL_EMP_EST_IND lookup type.
total_emp_min_ind	IN	VARCHAR R2	No	Validation: Validated against the TOTAL_EMP_MIN_IND lookup type.
parent_sub_ind	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
incorp_year	IN	NUMBER	No	
sic_code_type	IN	VARCHAR R2	No	Validation: Validated against AR lookup type SIC_CODE_TYPE
public_private_owner	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
internal_flag	IN	VARCHAR R2	No	Validation: Validated against AR lookup type YES/NO
				Default: N
local_activity_code_type	IN	VARCHAR R2	No	Validation: Validated against AR lookup type LOCAL_ACTIVITY_CODE_TYPE

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
local_activity_code	IN	VARCHAR R2	No	Validation: Validated against AR lookup type = value of local_activity_code_type.
emp_at_primary_adr	IN	VARCHAR R2	No	
emp_at_primary_adr_text	IN	VARCHAR R2	No	
emp_at_primary_adr_est_ind	IN	VARCHAR R2	No	Validation: Validated against the EMP_AT_PRIMARY_ADR_EST_IND lookup type.
emp_at_primary_adr_min_ind	IN	VARCHAR R2	No	Validation: Validated against the EMP_AT_PRIMARY_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_per_centile	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score_commentary2	IN	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY  Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score_commen tary10	IN	VARCHA R2	No	Validation: Validated against AR lookup type <b>CREDIT_SCORE_COMMENTARY</b>  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_de fault	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_comme ntary2	IN	VARCHA R2	No	Validation: Validated against AR lookup type <b>FAILURE_SCORE_COMMENTARY</b>  Comment: This parameter should only be populated with data provided by D&B.
failure_score_comme ntary3	IN	VARCHA R2	No	Validation: Validated against AR lookup type <b>FAILURE_SCORE_COMMENTARY</b>  Comment: This parameter should only be populated with data provided by D&B.
failure_score_comme ntary4	IN	VARCHA R2	No	Validation: Validated against AR lookup type <b>FAILURE_SCORE_COMMENTARY</b>  Comment: This parameter should only be populated with data provided by D&B.
failure_score_comme ntary5	IN	VARCHA R2	No	Validation: Validated against AR lookup type <b>FAILURE_SCORE_COMMENTARY</b>  Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
failure_score_commentary6	IN	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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	VARCHAR R2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
content_source_type	IN	VARCHAR R2	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	VARCHAR R2	No	
attribute_category	IN	VARCHAR R2	No	
attribute1	IN	VARCHAR R2	No	
attribute2	IN	VARCHAR R2	No	
attribute3	IN	VARCHAR R2	No	
attribute4	IN	VARCHAR R2	No	
attribute5	IN	VARCHAR R2	No	
attribute6	IN	VARCHAR R2	No	
attribute7	IN	VARCHAR R2	No	
attribute8	IN	VARCHAR R2	No	
attribute9	IN	VARCHAR R2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute10	IN	VARCHAR R2	No	
attribute11	IN	VARCHAR R2	No	
attribute12	IN	VARCHAR R2	No	
attribute13	IN	VARCHAR R2	No	
attribute14	IN	VARCHAR R2	No	
attribute15	IN	VARCHAR R2	No	
attribute16	IN	VARCHAR R2	No	
attribute17	IN	VARCHAR R2	No	
attribute18	IN	VARCHAR R2	No	
attribute19	IN	VARCHAR R2	No	
attribute20	IN	VARCHAR R2	No	
created_by_module	IN	VARCHAR R2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
do_not_confuse_with	IN	VARCHAR R2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
actual_content_source	IN	VARCHAR R2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.  Default - SST
home_country	IN	VARCHAR R2	No	Validation: Must exist in FND_TERRITORIES.

---

***party\_rec Record Type***

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated by from sequence
party_number	IN	VARCHAR 2	Yes/No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else mandatory.
validated_flag	IN	VARCHAR 2	No	Default: N
orig_system_reference	IN	VARCHAR 2	No	Default: party_id  Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR 2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR 2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
x_party_id	OUT	NUMBER	No	Comment: party_id of the party created
x_party_number	OUT	VARCHAR 2	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\_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_organization (
    p_init_msg_list          IN      VARCHAR2 := FND_API.G_FALSE,
    p_organization_rec        IN      ORGANIZATION_REC_TYPE,
    p_party_object_version_number IN OUT NUMBER,
    x_profile_id              OUT     NUMBER,
    x_return_status            OUT     VARCHAR2,
    x_msg_count                OUT     NUMBER,
    x_msg_data                 OUT     VARCHAR2
)
)
```

### Java Method

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

## Parameter Description and Validation

The following 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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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_in d	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
minority_owned_ty pe	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.

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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_curr ency	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_own ership_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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
local_activity_code_t_ype	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_ad_r	IN	VARCHAR2	No	
emp_at_primary_ad_r_text	IN	VARCHAR2	No	
emp_at_primary_ad_r_est_ind	IN	VARCHAR2	No	
emp_at_primary_ad_r_min_ind	IN	VARCHAR2	No	
high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
avg_high_credit	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
total_payments	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_natl_percentile	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_incd_default	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_age	IN	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_date	IN	DATE	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score_comme ntary2	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary3	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary4	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary5	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary6	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary7	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary8	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
credit_score_comme ntary9	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score_commentary10	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_class	IN	NUMBER	No	<p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_incd_default	IN	NUMBER	No	<p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_age	IN	NUMBER	No	<p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_date	IN	DATE	No	<p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary2	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary3	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary4	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary5	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
failure_score_commentary6	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary7	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary8	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary9	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
failure_score_commentary10	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTARY</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
maximum_credit_recommendation	IN	NUMBER	No	<p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
maximum_credit_currency_code	IN	VARCHAR2	No	<p>Validation: Foreign key to FND_CURRENCIES. CURRENCY_CODE</p> <p>Comment: This parameter should only be populated with data provided by D&amp;B.</p>
displayed_duns_party_id	IN	NUMBER	No	<p>Validation: Foreign key to HZ_PARTIES. PARTY_ID</p>
content_source_type	IN	VARCHAR2	No	<p>Comment: This parameter is no longer used. Use actual_content_source.</p> <p>Validation: Cannot be updated</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
do_not_confuse_with	IN	VARCHAR2	No	
actual_content_source	IN	VARCHAR2	No	Validation : Cannot be updated.
home_country	IN	VARCHAR2	No	Validation: Must exist in FND_TERRITORIES.

---

***party\_rec Record Type***

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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.

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	
attribute15	IN	VARCHAR2	No	

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

## 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\_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 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_pre_name_a_djunct	IN	VARCHAR 2	No	Validation: Validated against AR lookup type CONTACT_TITLE
person_first_name	IN	VARCHAR 2	Yes/No	Validation: Either one of person_first_name or person_last_name should be passed in
person_middle_name	IN	VARCHAR 2	No	
person_last_name	IN	VARCHAR 2	Yes/No	Validation: Either one of person_first_name or person_last_name should be passed in
person_name_suffix	IN	VARCHAR 2	No	
person_title	IN	VARCHAR 2	No	
person_academic_title	IN	VARCHAR 2	No	
person_previous_last_name	IN	VARCHAR 2	No	
person_initials	IN	VARCHAR 2	No	
known_as	IN	VARCHAR 2	No	
known_as2	IN	VARCHAR 2	No	
known_as3	IN	VARCHAR 2	No	
known_as4	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
known_as5	IN	VARCHAR 2	No	
person_name_phone tic	IN	VARCHAR 2	No	
person_first_name_p honetic	IN	VARCHAR 2	No	
person_last_name_p honetic	IN	VARCHAR 2	No	
middle_name_phon etic	IN	VARCHAR 2	No	
tax_reference	IN	VARCHAR 2	No	
jgzz_fiscal_code	IN	VARCHAR 2	No	
person_iden_type	IN	VARCHAR 2	No	Validation: Validated against AR lookup type HZ_PERSON_IDEN_TYPE.
person_identifier	IN	VARCHAR 2	No	
date_of_birth	IN	DATE	No	
place_of_birth	IN	VARCHAR 2	No	
deceased_ind	IN	VARCHAR 2	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
gender	IN	VARCHAR 2	No	Validation: Validated against AR lookup type HZ_GENDER.
declared_ethnicity	IN	VARCHAR 2	No	
marital_status	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
household_income	IN	NUMBER	No	
household_size	IN	NUMBER	No	
rent_own_ind	IN	VARCHAR 2	No	Validation: Validated against AR lookup type OWN_RENT_IND.
last_known_gps	IN	VARCHAR 2	No	
content_source_type	IN	VARCHAR 2	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	VARCHAR 2	No	Default: N
attribute_category	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
actual_content_source	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.  Default - SST

#### ***Party\_rec Record Type Attributes***

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_referen ce	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
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_a_djunct	IN	VARCHAR 2	No	Validation: Validated against AR lookup type CONTACT_TITLE
person_first_name	IN	VARCHAR 2	No	Validation: During update both person_first_name and person_last_name cannot be set to null

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_middle_name	IN	VARCHAR 2	No	
person_last_name	IN	VARCHAR 2	No	Validation: During update both person_first_name and person_last_name cannot be set to null
person_name_suffix	IN	VARCHAR 2	No	
person_title	IN	VARCHAR 2	No	
person_academic_title	IN	VARCHAR 2	No	
person_previous_lastname	IN	VARCHAR 2	No	
person_initials	IN	VARCHAR 2	No	
known_as	IN	VARCHAR 2	No	
known_as2	IN	VARCHAR 2	No	
known_as3	IN	VARCHAR 2	No	
known_as4	IN	VARCHAR 2	No	
known_as5	IN	VARCHAR 2	No	
person_name_phone	IN	VARCHAR 2	No	
person_first_name_phonetic	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_last_name_p honetic	IN	VARCHAR 2	No	
middle_name_phon etic	IN	VARCHAR 2	No	
tax_reference	IN	VARCHAR 2	No	
jgzz_fiscal_code	IN	VARCHAR 2	No	
person_iden_type	IN	VARCHAR 2	No	Validation: Validated against AR lookup type HZ_PERSON_IDEN_TYPE.
person_identifier	IN	VARCHAR 2	No	
date_of_birth	IN	DATE	No	
place_of_birth	IN	VARCHAR 2	No	
deceased_ind	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Validated against AR lookup type HZ_GENDER.
declared_ethnicity	IN	VARCHAR 2	No	
marital_status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type MARITAL_STATUS
marital_status_effect ive_date	IN	DATE	No	
personal_income	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
head_of_household_flag	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
household_income	IN	NUMBER	No	
household_size	IN	NUMBER	No	
rent_own_ind	IN	VARCHAR 2	No	Validation: If the value is changed, then the value is validated against the AR lookup type OWN_RENT_IND.
last_known_gps	IN	VARCHAR 2	No	
content_source_type	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Cannot be updated
internal_flag	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
actual_content_source	IN	VARCHAR 2	No	Validation : Cannot be updated

#### ***Party\_rec Record Type Attributes***

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS.</li><li>• Cannot be updated to null</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

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

## Create Group API

### Description

This routine is used to create a group. The API creates a record in the HZ\_PARTIES table with party type 'GROUP'. The HZ\_PARTIES table holds the basic information about the party. There is no profile information for a Group party unlike an Organization or Person party. 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\_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 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	VARCHAR 2	Yes	Validation: Mandatory attribute

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
group_type	IN	VARCHAR2	Yes	Validation: Mandatory attribute
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.

#### ***party\_rec Record Type Attributes***

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
party_number	IN	VARCHAR2	Yes/No	Validation: Generated from sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else unique and mandatory
validated_flag	IN	VARCHAR2	No	Default: N
orig_system_reference	IN	VARCHAR2	No	Default: party_id 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.
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	

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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
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
    GroupRec
    BigDecimal []
    String []
    BigDecimal []
    String []
) 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, else validated against AR lookup type HZ_CREATED_BY_MODULE.
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: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS</li><li>• Cannot be updated to null</li></ul>
category_code	IN	VARCHAR2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	

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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute24	IN	VARCHAR2	No	
p_party_object_vers ion_number	IN/OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing party record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number of the record from HZ_PARTIES</li> <li>• Return new value after update.</li> </ul>

## Party 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_natnl_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_convvs              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_evtnt_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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_rating_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence hz_credit_ratings_s.  Not updateable.
description	IN	VARCHAR 2	No	
party_id	IN	NUMBER	Yes	Validation: Must a valid party from hz_parties.  Non-updateable
rating	IN	VARCHAR 2	No	
rated_as_of_date	IN	DATE	No	
rating_organization	IN	VARCHAR 2	No	
comments	IN	VARCHAR 2	No	
det_history_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_embt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
criminal_proceeding _ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
claims_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
secured_flng_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_lgl_event_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
disaster_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
oprg_spec_evnt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
other_spec_evnt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
status	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS
avg_high_credit	IN	NUMBER	No	
credit_score	IN	VARCHAR 2	No	
credit_score_age	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	
credit_score_commentary	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary2	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary3	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary4	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary5	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary6	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary7	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary8	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
credit_score_commentary9	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary10	IN	VARCHAR 2	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	VARCHAR 2	No	
failure_score_age	IN	NUMBER	No	
failure_score_class	IN	NUMBER	No	
failure_score_commentary	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary2	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary3	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary4	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary5	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary6	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary7	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
failure_score_comm entary8	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_comm entary9	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_comm entary10	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_date	IN	DATE	No	
failure_score_incd_d efault	IN	NUMBER	No	
failure_score_natnl_ percentile	IN	NUMBER	No	
failure_score_overri de_code	IN	VARCHAR 2	No	
global_failure_score	IN	VARCHAR 2	No	
debarment_ind	IN	VARCHAR 2	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_cu rrency_code	IN	VARCHAR 2	No	Validation: Foreign key to fnd_currencies. currency_code
maximum_credit_rc md	IN	NUMBER	No	
paydex_norm	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
paydex_score	IN	VARCHAR 2	No	
paydex_three_mont hs_ago	IN	VARCHAR 2	No	
credit_score_overrid e_code	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE
cr_scr_clas_expl	IN	VARCHAR 2	No	
low_rng_delq_scr	IN	NUMBER	No	
high_rng_delq_scr	IN	NUMBER	No	
delq_pmt_rng_prnt	IN	NUMBER	No	
delq_pmt_pctg_for_ all_firms	IN	NUMBER	No	
num_trade_experien ces	IN	NUMBER	No	
paydex_firm_days	IN	VARCHAR 2	No	
paydex_firm_comme nt	IN	VARCHAR 2	No	
paydex_industry_da ys	IN	VARCHAR 2	No	
paydex_industry_co mment	IN	VARCHAR 2	No	
paydex_comment	IN	VARCHAR 2	No	
suit_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
lien_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
judgement_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
bankruptcy_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
no_trade_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
prnt_hq_bkcy_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
num_prnt_bkcy_filing	IN	NUMBER	No	
prnt_bkcy_filg_type	IN	VARCHAR 2	No	
prnt_bkcy_filg_chapter	IN	NUMBER	No	
prnt_bkcy_filg_date	IN	DATE	No	
num_prnt_bkcy_cons	IN	NUMBER	No	
prnt_bkcy_conv_date	IN	DATE	No	
prnt_bkcy_chapter_conv	IN	VARCHAR 2	No	
slow_trade_expl	IN	VARCHAR 2	No	
negv_pmt_expl	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
pub_rec_expl	IN	VARCHAR 2	No	
business_discontinu ed	IN	VARCHAR 2	No	
spcl_event_comment	IN	VARCHAR 2	No	
num_spcl_event	IN	NUMBER	No	
spcl_event_update_ date	IN	DATE	No	
spcl_evnt_txt	IN	VARCHAR 2	No	
actual_content_sourc e	IN	VARCHAR 2	No	Validate: HZ_MIXNM.Utility Validate Content Source
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	VARCHAR 2	No	
party_id	IN	NUMBER	No	Validation: Must a valid party from hz_parties. Non-updateable
rating	IN	VARCHAR 2	No	
rated_as_of_date	IN	DATE	No	
rating_organization	IN	VARCHAR 2	No	
comments	IN	VARCHAR 2	No	
det_history_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
fincl_embt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
criminal_proceeding _ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
claims_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
secured_flng_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
fincl_lgl_event_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
disaster_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
oprg_spec_evnt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
other_spec_evnt_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
status	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS
avg_high_credit	IN	NUMBER	No	
credit_score	IN	VARCHAR 2	No	
credit_score_age	IN	NUMBER	No	
credit_score_class	IN	NUMBER	No	
credit_score_commentary	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary2	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary3	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary4	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary5	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY
credit_score_commentary6	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
failure_score_commentary6	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary7	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary8	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary9	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY
failure_score_commentary10	IN	VARCHAR 2	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	VARCHAR 2	No	
global_failure_score	IN	VARCHAR 2	No	
debarment_ind	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Foreign key to fnd_currencies. currency_code

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
maximum_credit_rc_md	IN	NUMBER	No	
paydex_norm	IN	VARCHAR 2	No	
paydex_score	IN	VARCHAR 2	No	
paydex_three_mont_hs_ago	IN	VARCHAR 2	No	
credit_score_overrid_e_code	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE
cr_scr_clas_expl	IN	VARCHAR 2	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	VARCHAR 2	No	
paydex_firm_comm_ent	IN	VARCHAR 2	No	
paydex_industry_da ys	IN	VARCHAR 2	No	
paydex_industry_com mment	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
paydex_comment	IN	VARCHAR 2	No	
suit_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
lien_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
judgement_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
bankruptcy_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
no_trade_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
prnt_hq_bkcy_ind	IN	VARCHAR 2	No	Validation: Must be a valid lookup value from lookup YES/NO
num_prnt_bkcy_filing	IN	NUMBER	No	
prnt_bkcy_filg_type	IN	VARCHAR 2	No	
prnt_bkcy_filg_chapter	IN	NUMBER	No	
prnt_bkcy_filg_date	IN	DATE	No	
num_prnt_bkcy_conv	IN	NUMBER	No	
prnt_bkcy_conv_date	IN	DATE	No	
prnt_bkcy_chapter_conv	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
slow_trade_expl	IN	VARCHAR 2	No	
negv_pmt_expl	IN	VARCHAR 2	No	
pub_rec_expl	IN	VARCHAR 2	No	
business_discontinu ed	IN	VARCHAR 2	No	
spcl_event_comment	IN	VARCHAR 2	No	
num_spcl_event	IN	NUMBER	No	
spcl_event_update_ date	IN	DATE	No	
spcl_evnt_txt	IN	VARCHAR 2	No	
actual_content_sourc e	IN	VARCHAR 2	No	Validate: HZ_MIXNM.utility ValidateContentSource
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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 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.
relationship_rec_type:=HZ_RELATIONSHIP_V2PUB.G_MISS_REL_REC
)
```

## PL/SQL Record Structure for Org Contact Role

```
TYPE org_contact_role_rec_type IS RECORD(
    org_contact_role_id                       NUMBER,
    role_type                                 VARCHAR2(30),
    primary_flag                             VARCHAR2(1),
    org_contact_id                           NUMBER,
    orig_system_reference                   VARCHAR2(240),
    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 String application_id;  
  
    public String 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_type;  
    public String status;  
    public String created_by_module;  
    public BigDecimal application_id;  
  
    public OrgContactRoleRec();  
    public OrgContactRoleRec(boolean __RosettaUseGMISSValues);  
}
```

## Create Org Contact API

### Description

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

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	Comment: This parameter is no longer used. Use hz_parties.person_pre_name_adjunct instead.
job_title	IN	VARCHAR2	No	
decision_maker_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
job_title_code	IN	VARCHAR2	No	Validation: Validated against AR lookup type RESPONSIBILITY
reference_use_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
rank	IN	VARCHAR2	No	
party_site_id	IN	NUMBER	No	Validation: Foreign key to HZ_PARTY_SITES. PARTY_SITE_ID. If a value is passed, then the party_id of the party site should be same as the object_id of the relationship to be created for this org contact.
orig_system_reference	IN	VARCHAR2	No	Default: org_contact_id 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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to HZ_PARTIES.PARTY_ID</li> </ul> <p>Comment: Pass the party_id of the contact person here.</p>
subject_type	IN	VARCHAR2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to FND_OBJECT_INSTANCE_SETS.INSTANCESET_NAME</li> </ul> <p>Comment: Pass the party_type of the subject person, which is PERSON here.</p>
subject_table_name	IN	VARCHAR2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to FND_OBJECTS.OBJ.NAME</li> </ul> <p>Comment: Pass HZ_PARTIES for the table name of the source of the subject.</p>
object_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to HZ_PARTIES.PARTY_ID</li> </ul> <p>Comment: Pass the party_id of the organization or person for which you are creating the contact.</p>

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
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***

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
attribute21	IN	VARCHAR2	No	
attribute22	IN	VARCHAR2	No	
attribute23	IN	VARCHAR2	No	
attribute24	IN	VARCHAR2	No	
x_org_contact_id	OUT	NUMBER	No	Comment: org_contact_id of the org contact record created
x_party_rel_id	OUT	NUMBER	No	Comment: relationship_id of the relationship record created
x_party_id	OUT	NUMBER	No	Comment: party_id of the party record created
x_party_number	OUT	VARCHAR2	No	Comment: party_number of the party record created

## Update Org Contact API

### Description

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

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_FALSE,
    p_org_contact_rec                     IN      ORG_CONTACT_REC_TYPE,
    p_cont_object_version_number          IN OUT   NUMBER,
    p_rel_object_version_number          IN OUT   NUMBER,
    p_party_object_version_number        IN OUT   NUMBER,
    x_return_status                       OUT     VARCHAR2,
    x_msg_count                           OUT     NUMBER,
    x_msg_data                            OUT     VARCHAR2
)
```

## Java Method

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

## Parameter Description and Validation

The following 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	VARCHAR 2	No	
contact_number	IN	VARCHAR 2	No	
department_code	IN	VARCHAR 2	No	Validation: Validated against AR lookup type DEPARTMENT_TYPE

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
department	IN	VARCHAR 2	No	
title	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use hz_parties.person_pre_name_adjunct instead.
job_title	IN	VARCHAR 2	No	
decision_maker_flag	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
job_title_code	IN	VARCHAR 2	No	Validation: Validated against AR lookup type RESPONSIBILITY
reference_use_flag	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
rank	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system.
attribute_category	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
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	<p>Validation: Valid relationship_id should be passed in</p> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the relationship_id of the relationship record for this org contact.</li> <li>• Pass if you want to update the relationship record.</li> </ul>
subject_id	IN	NUMBER	No	Validation: Non updateable
subject_type	IN	VARCHAR 2	No	Validation: Non updateable
subject_table_name	IN	VARCHAR 2	No	Validation: Non updateable
object_id	IN	NUMBER	No	Validation: Non updateable
object_type	IN	VARCHAR 2	No	Validation: Non updateable
object_table_name	IN	VARCHAR 2	No	Validation: Non updateable
relationship_code	IN	VARCHAR 2	No	Validation: Non updateable
relationship_type	IN	VARCHAR 2	No	Validation: Non updateable
comments	IN	VARCHAR 2	No	
start_date	IN	DATE	No	Validation: Cannot be updated to null
end_date	IN	DATE	No	Validation: Cannot be less than start_date

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	IN	VARCHAR 2	No	Validation: Validate against AR lookup type REGISTRY_STATUS  Cannot be updated to null
content_source_type	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use actual_content_source.  Validation: Not updateable
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
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	<p>Validation: Valid party id in HZ_PARTIES.</p> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass party_id of the relationship's party record.</li> <li>• Pass if you want to update the party record.</li> </ul>
party_number	IN	VARCHAR 2	No	Validation: Not updateable
validated_flag	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validate against AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>
category_code	IN	VARCHAR 2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
p_cont_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against the value in the database for the existing org contact record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number of the record from HZ_ORG_CONTACTS</li> <li>• Return a new value after update</li> </ul>

---

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

## Create Org Contact Role API

### Description

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

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_ROLE_REC_TYPE,
    x_org_contact_role_id                OUT         NUMBER,
    x_return_status                     OUT         VARCHAR2,
    x_msg_count                         OUT         NUMBER,
    x_msg_data                          OUT         VARCHAR2
)
```

## Java Method

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

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
org_contact_role_id	IN	NUMBER	No	Validation: Unique when passed in, else generated from sequence
role_type	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against AR lookup type CONTACT_ROLE_TYPE</li></ul>
primary_flag	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
org_contact_id	IN	NUMBER	Yes	Validation: Mandatory attribute  Foreign key to HZ_ORG_CONTACTS. ORG_CONTACT_ID
orig_system_reference	IN	VARCHAR 2	No	Default: org_contact_role_id  Validation: If orig_system is passed in, then orig_system_reference is required.
orig_system	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system.
role_level	IN	VARCHAR 2	No	
primary_contact_per_role_type	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type REGISTRY_STATUS  Default: A
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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_ROLE_REC_TYPE,
    p_object_version_number              IN OUT NUMBER,
    x_return_status                      OUT     VARCHAR2,
    x_msg_count                          OUT     NUMBER,
    x_msg_data                           OUT     VARCHAR2
)
```

### Java Method

```
public static void updateOrgContactRole(
    OracleConnection connection,
    String 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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
role_type	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot be set to null during update</li> <li>• Validated against AR lookup type CONTACT_ROLE_TYPE</li> </ul>
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
org_contact_id	IN	NUMBER	No	Validation: Not updateable
orig_system_reference	IN	VARCHAR2	No	Validation: 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	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists.

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

### Other Validations

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



# 5

---

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

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 Integer               party_id;
    public String                native_language;
    public Boolean               primary_language_indicator;
    public String                reads_level;
    public String                speaks_level;
    public String                writes_level;
    public String                status;
    public Integer               created_by_module;
    public Integer               application_id;
    public PersonLanguageRec();
    public PersonLanguageRec(boolean_RosettaUseGMISSValues);
}
```

## Create Person Language API

### Description

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

## PL/SQL Procedure

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

## Java Method

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

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
language_use_refere nce_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
language_name	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Foreign key to fnd_languages.language_code</li></ul>
party_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Foreign key to hz_parties.party_id</li></ul>
native_language	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
primary_language_indicator	IN	VARCHAR 2	No	
reads_level	IN	VARCHAR 2	No	
speaks_level	IN	VARCHAR 2	No	
writes_level	IN	VARCHAR 2	No	
spoken_comprehension_level				Validation: Validated against the HZ_LANGUAGE_PROFICIENCY lookup type.
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type REGISTRY_STATUS  Default: A
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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_referenc_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

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

## Other Validations

- There can be only one record for a given party and language.

- A party can have only one native language.
- A party can have only one primary language, 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_in_d	IN	VARCHAR 2	No	Validation: Validated against the HZ_CITIZENSHIP_ACQUISITION lookup type.
country_code	IN	VARCHAR 2	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	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
document_reference	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
birth_or_selected_in_d	IN	VARCHAR 2	No	Validation: validated against AR lookup type HZ_CITIZENSHIP_ACQUISITION
country_code	IN	VARCHAR 2	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	VARCHAR 2	No	
document_reference	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	VARCHAR 2	No	
degree_received	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	IN	VARCHAR 2	No	Validation: If the value is modified, then the value is validated against the HZ_TYPE_OF_SCHOOL lookup type.
Status	IN	VARCHAR 2	No	Validation: Validated against the REGISTRY_STATUS lookup type.
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	VARCHAR 2	No	
degree_received	IN	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
school_attended_name	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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_EMPLOYMENT_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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
military_rank	IN	VARCHAR 2	No	
Served	IN	VARCHAR 2	No	
Station	IN	VARCHAR 2	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	VARCHAR 2	No	
faculty_position_ind	IN	VARCHAR 2	Yes	Default: N  Validation: validated against AR lookup type YES/NO.
tenure_code	IN	VARCHAR 2	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	VARCHAR 2	No	
status	IN	VARCHAR 2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	
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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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_divisi on_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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
work_class_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence.  Not updateable.
level_of_experience	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
level_of_experience	IN	VARCHAR 2	No	
work_class_name	IN	VARCHAR 2	Yes	
employment_history_id	IN	NUMBER	Yes	Validation: Must exist in HZ_EMPLOYMENT_HISTORY.Not updateable.
reason_for_leaving	IN	VARCHAR 2	No	
Status	IN	VARCHAR 2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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
) ;
```

### Parameter Description and Validation

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
person_interest_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence.  Not updateable.
level_of_interest	IN	VARCHAR2	No	
party_id	IN	NUMBER	Yes	Validation: Must exist in HZ_PARTIES.
level_of_participation	IN	VARCHAR2	No	
interest_type_code	IN	VARCHAR2	No	
comments	IN	VARCHAR2	No	
sport_indicator	IN	VARCHAR2	No	Validation: validated against AR lookup type YES/NO.
sub_interest_type_code	IN	VARCHAR2	No	
interest_name	IN	VARCHAR2	Yes	Validation: Mandatory attribute.
team	IN	VARCHAR2	No	
since	IN	DATE	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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
);
```

### Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
person_interest_id	IN	NUMBER	No	Validation: From hz_person_interests. Not updateable.
level_of_interest	IN	VARCHAR2	No	
party_id	IN	NUMBER	Yes	Validation: Not updateable.
level_of_participation	IN	VARCHAR2	No	
interest_type_code	IN	VARCHAR2	No	
comments	IN	VARCHAR2	No	
sport_indicator	IN	VARCHAR2	No	Validation: validated against AR lookup type YES/NO.
sub_interest_type_code	IN	VARCHAR2	No	
interest_name	IN	VARCHAR2	Yes	Validation: Mandatory attribute.
team	IN	VARCHAR2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
since	IN	DATE	No	
status	IN	VARCHAR2	No	Validation: validated against AR lookup type REGISTRY_STATUS.
created_by_module	IN	VARCHAR2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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.
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comments
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_principal_s_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	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	Validation: Not updateable.
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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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_principal_s_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	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
projected_actual_flag	IN	VARCHAR2	No	
status	IN	VARCHAR2	Yes	Validation: validated against AR lookup type REGISTRY_STATUS
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comments</b>
financial_number_name	IN	VARCHAR 2	No	Validation: Must be a valid lookup under FIN_NUM_NAME lookup type.
financial_units_applied	IN	NUMBER	No	
financial_number_currency	IN	VARCHAR 2	No	
projected_actual_flag	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	Yes	Validation: validated against AR lookup type REGISTRY_STATUS
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

## 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_SOURCE_TYPE
CONSTANT VARCHAR2(30) :=USER_ENTERED;
HZ_GEOOMETRY_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_geometry_default                      NUMBER,
    loc_hierarchy_id                         VARCHAR2(30),
    sales_tax_geocode                        VARCHAR2(30),
    sales_tax_inside_city_limits            NUMBER,
    fa_location_id                           VARCHAR2(30):=
    G_MISS_CONTENT_SOURCE_TYPE,
    attribute_category                       VARCHAR2(30),
    attribute1                                VARCHAR2(150),
    attribute2                                VARCHAR2(150),
    attribute3                                VARCHAR2(150),
    attribute4                                VARCHAR2(150),
    attribute5                                VARCHAR2(150),
    attribute6                                VARCHAR2(150),
    attribute7                                VARCHAR2(150),
    attribute8                                VARCHAR2(150),
    attribute9                                VARCHAR2(150),
    attribute10                               VARCHAR2(150),
    attribute11                               VARCHAR2(150),
    attribute12                               VARCHAR2(150),
    attribute13                               VARCHAR2(150),
    attribute14                               VARCHAR2(150),
    attribute15                               VARCHAR2(150),
    attribute16                               VARCHAR2(150),
    attribute17                               VARCHAR2(150),
    attribute18                               VARCHAR2(150),
```

```
attribute19          VARCHAR2(150),  
attribute20          VARCHAR2(150),  
timezone_id           NUMBER,  
created_by_module     VARCHAR2(150),  
application_id        NUMBER  
)
```

## Java Inner Class for Location

```
public static class LocationRec {  
    public BigDecimal location_id;  
    public String orig_system_reference;  
    public String 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 String address_effective_date;  
    public String 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 String 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.

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
validated_flag	IN	VARCHAR2	No	
address_lines_phone_tic	IN	VARCHAR2	No	
po_box_number	IN	VARCHAR2	No	Comment: This parameter is no longer used.
house_number	IN	VARCHAR2	No	Comment: This parameter is no longer used.
street_suffix	IN	VARCHAR2	No	Comment: This parameter is no longer used.
street	IN	VARCHAR2	No	Comment: This parameter is no longer used.
street_number	IN	VARCHAR2	No	Comment: This parameter is no longer used.
floor	IN	VARCHAR2	No	Comment: This parameter is no longer used.
suite	IN	VARCHAR2	No	Comment: This parameter is no longer used.
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
description	IN	VARCHAR2	No	
geometry	IN	OBJECT	No	
loc_hierarchy_id	IN	NUMBER	No	
sales_tax_geocode	IN	VARCHAR2	No	
sales_tax_inside_city	IN	VARCHAR2	No	
				_limits
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
location_id	IN	NUMBER	Yes	Validation: Valid location_id should be passed in.
orig_system_reference	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
country	IN	VARCHAR 2	No	Validation: Foreign key to fnd_territories.territory_code
address1	IN	VARCHAR 2	No	Validation: Cannot be set to null during update
address2	IN	VARCHAR 2	No	
address3	IN	VARCHAR 2	No	
address4	IN	VARCHAR 2	No	
city	IN	VARCHAR 2	No	
postal_code	IN	VARCHAR 2	No	
state	IN	VARCHAR 2	No	
province	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
county	IN	VARCHAR 2	No	
address_key	IN	VARCHAR 2	No	
address_style	IN	VARCHAR 2	No	
validated_flag	IN	VARCHAR 2	No	
address_lines_phone_tic	IN	VARCHAR 2	No	
po_box_number	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
house_number	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
street_suffix	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
street	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
street_number	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
floor	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
suite	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
postal_plus4_code	IN	VARCHAR 2	No	
position	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
delivery_point_code	IN	VARCHAR 2	No	
location_directions	IN	VARCHAR 2	No	
address_effective_da te	IN	DATE	No	
address_expiration_‑ date	IN	DATE	No	
clli_code	IN	VARCHAR 2	No	
language	IN	VARCHAR 2	No	Validation: Foreign key to fnd_languages. language_code (installed)
short_description	IN	VARCHAR 2	No	
description	IN	VARCHAR 2	No	
geometry	IN	OBJECT	No	
loc_hierarchy_id	IN	NUMBER	No	
sales_tax_geocode	IN	VARCHAR 2	No	
sales_tax_inside_city _limits	IN	VARCHAR 2	No	
fa_location_id	IN	NUMBER	No	
content_source_type	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use actual_content_source.
				Validation: Not updateable

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezones. timezone_id
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_source	IN	VARCHAR 2	No	Validation: Cannot be updated.

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

## Party Site APIs

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

**Java Class Name:** HzPartySiteV2Pub

## PL/SQL Record Structure for Party Site

```
TYPE party_site_rec_type
      party_site_id
      party_id
      location_id
      party_site_number
      orig_system_reference
      orig_system
      mailstop
      identifying_address_flag
      status
      party_site_name
      attribute_category
      attribute1
      attribute2
      attribute3
      attribute4
      attribute5
      attribute6
      attribute7
      attribute8
      attribute9
      attribute10
      attribute11
      attribute12
      attribute13
      attribute14
      attribute15
      attribute16
      attribute17
      attribute18
      attribute19
      attribute20
      language
      addressee
      created_by_module
      application_id
      global_location_number
      duns_number_c
)
IS RECORD(
      NUMBER,
      NUMBER,
      NUMBER,
      VARCHAR2(30),
      VARCHAR2(240),
      VARCHAR2(30),
      VARCHAR2(60),
      VARCHAR2(1),
      VARCHAR2(1),
      VARCHAR2(240),
      VARCHAR2(30),
      VARCHAR2(150),
      NUMBER
      VARCHAR2(40)
      VARCHAR2(30)
```

## PL/SQL Record Structure for Party Site Use

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign Key to hz_parties.party_id</li> </ul>
location_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign Key to hz_locations.location_id</li> </ul>
party_site_number	IN	VARCHAR 2	Yes/No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Required when profile HZ_GENERATE_PARTY_SITE_NUMBER = N, else generated from sequence.</li> <li>• Unique when passed in</li> </ul>
orig_system_reference	IN	VARCHAR 2	No	<p>Default: party_site_id</p> <p>Validation: If orig_system is passed in, then orig_system_reference is required.</p>
orig_system	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
mailstop	IN	VARCHAR 2	No	
identifying_address_flag	IN	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type REGISTRY STATUS
party_site_name	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
language	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use hz_locations.language instead.
addressee	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_location_number	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot have nonnumeric characters.</li> <li>• Must be 13 digits long.</li> <li>• Must satisfy the check-digit algorithm.</li> </ul> <p>The check-digit algorithm is:</p> <ol style="list-style-type: none"> <li>1. Add the digits in even position. Multiply this sum by three.</li> <li>2. Add the digits in odd position, excluding the thirteenth digit.</li> <li>3. Add the values obtained in the previous two steps.</li> </ol> <p>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>
duns_number_c	IN	VARCHAR 2	No	Comment: For internal use only.
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 deactivate 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_SITEUSES 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: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY STATUS</li><li>• Cannot be set to null during update</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_site_name	IN	VARCHAR2	No	
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
language	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use hz_locations.language instead.
addressee	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
global_location_number	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot have nonnumeric characters.</li> <li>• Must be 13 digits long.</li> <li>• Must satisfy the check-digit algorithm.</li> </ul> <p>The check-digit algorithm is:</p> <ol style="list-style-type: none"> <li>1. Add the digits in even position. Multiply this sum by three.</li> <li>2. Add the digits in odd position, excluding the thirteenth digit.</li> <li>3. Add the values obtained in the previous two steps.</li> </ol> <p>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>
duns_number_c	IN	VARCHAR2	No	Comment: For internal use only.

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

## Other Validations

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

## Create Party Site Use API

### Description

This routine is used to create a Party Site Use for a Party Site. The API creates a record in the HZ\_PARTY\_SITEUSES 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: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against AR lookup type PARTY_SITE_USE_CODE</li></ul>
party_site_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Foreign key to hz_party_sites.party_site_id</li></ul>
primary_per_type	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO. 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

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

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

# 6

---

## Contact Point API Use

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_CONTENT_SOURCE_TYPE,
    attribute_category VARCHAR2(30),
    attribute1 VARCHAR2(150),
    attribute2 VARCHAR2(150),
    attribute3 VARCHAR2(150),
    attribute4 VARCHAR2(150),
    attribute5 VARCHAR2(150),
    attribute6 VARCHAR2(150),
    attribute7 VARCHAR2(150),
    attribute8 VARCHAR2(150),
    attribute9 VARCHAR2(150),
    attribute10 VARCHAR2(150),
    attribute11 VARCHAR2(150),
    attribute12 VARCHAR2(150),
    attribute13 VARCHAR2(150),
    attribute14 VARCHAR2(150),
    attribute15 VARCHAR2(150),
    attribute16 VARCHAR2(150),
    attribute17 VARCHAR2(150),
    attribute18 VARCHAR2(150),
    attribute19 VARCHAR2(150),
    attribute20 VARCHAR2(150),
    contact_point_purpose VARCHAR2(30),
    primary_by_purpose VARCHAR2(1),
    created_by_module VARCHAR2(150),
    application_id NUMBER,
    actual_content_source VARCHAR2(30)
)
```

## PL/SQL Record Structure for EDI

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

## PL/SQL Record Structure for EMAIL

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

## PL/SQL Record Structure for PHONE

```
TYPE phone_rec_type
    phone_calling_calendar
    last_contact_dt_time
    timezone_id
    phone_area_code
    phone_country_code
    phone_number
    phone_extension
    phone_line_type
    raw_phone_numberVARCHAR2(60)
)
```

```
IS RECORD (
    VARCHAR2(30),
    DATE,
    NUMBER,
    VARCHAR2(10),
    VARCHAR2(10),
    VARCHAR2(40),
    VARCHAR2(20),
    VARCHAR2(30),
```

## PL/SQL Record Structure for TELEX

```
TYPE telex_rec_type
    telex_number
)
```

```
IS RECORD (
    VARCHAR2(50)
```

## PL/SQL Record Structure for WEB

```
TYPE web_rec_type IS RECORD (
    web_type
    url
)

```

```
    VARCHAR2(60),
    VARCHAR2(2000)
```

## PL/SQL Record Structure for EFT

```
TYPE eft_rec_type
    eft_transmission_program_id
    eft_printing_program_id
    eft_user_number
    eft_swift_code
)
```

```
IS RECORD (
    NUMBER,
    NUMBER,
    VARCHAR2(30),
    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 String 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 String 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 String 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_id	IN	NUMBER	No	Validation: Unique if passed in, else generated from sequence
contact_point_type	IN	VARCHAR2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against AR lookup type COMMUNICATION_TYPE</li> <li>• EDI and EFT contact points must be Organization-type parties.</li> </ul>
status	IN	VARCHAR2	No	<p>Validation: Validated against AR lookup type REGISTRY_STATUS</p> <p>Default: A</p>
owner_table_name	IN	VARCHAR2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against AR lookup type OWNER_TABLE_NAME</li> </ul> <p>Comment: If you are creating contact point for a party, pass HZ_PARTIES, if you are creating contact point for a party site, pass HZ_PARTY_SITES</p>
owner_table_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign Key hz_parties.party_id when owner_table_name = HZ_PARTIES.</li> <li>• Foreign Key to hz_party_sites.party_site_id when owner_table_name = HZ_PARTY_SITES</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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.
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_purpo se	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type=WEB</li> </ul>
primary_by_purpose	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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***

---

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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**

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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**

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default Comment</b>
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	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default Comment</b>
phone_line_type	IN	VARCHAR2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against AR lookup type PHONE_LINE_TYPE</li> </ul>
raw_phone_number	IN	VARCHAR2	Yes/No	Validation: Mandatory if phone_number is not passed in

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

#### ***p\_telex\_rec record type***

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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***

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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**

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_REC,
    p_phone_rec               IN      PHONE_REC_TYPE:= G_MISS_PHONE_REC,
    p_telex_rec               IN      TELEX_REC_TYPE:= G_MISS_TELEX_REC,
    p_web_rec                 IN      WEB_REC_TYPE:= G_MISS_WEB_REC,
    p_object_version_number   IN OUT NUMBER,
    x_return_status            OUT     VARCHAR2,
    x_msg_count                OUT    NUMBER,
    x_msg_data                  OUT    VARCHAR2
)
```

## Java Method for EDI Contact Points

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

## Java Method for EMAIL Contact Points

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

## Java Method for PHONE Contact Points

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

### **Java Method for WEB Contact Points**

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

### **Java Method for TELEX Contact Points**

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

### **Java Method for EFT Contact Points**

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

### **Generic Java Method**

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

### **Parameter Description and Validation**

The following 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_id	IN	NUMBER	Yes	Validation: Valid contact_point_id should be passed in  Comment: Pass the contact_point_id from hz_contact_points table
contact_point_type	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS</li><li>• Cannot be set to null during update</li></ul>
owner_table_name	IN	VARCHAR2	No	Validation: Not updateable
owner_table_id	IN	NUMBER	No	Validation: Not updateable
primary_flag	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
orig_system_reference	IN	VARCHAR2	No	Validation: 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	

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_purpo se	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type=WEB</li> </ul>
primary_by_purpos e	IN	VARCHAR2	No	Validation: Validated against AR lookup type YES/NO
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_sour ce	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*

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
edi_transaction_han dling	IN	VARCHAR2	No	
edi_id_number	IN	VARCHAR2	No	
edi_payment_metho d	IN	VARCHAR2	No	
edi_payment_format	IN	VARCHAR2	No	
edi_remittance_meth od	IN	VARCHAR2	No	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
edi_remittance_instr uction	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*

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
email_format	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type EMAIL_FORMAT.</li> <li>• Cannot set to null during update</li> </ul>
email_address	IN	VARCHAR2	No	Validation: Cannot be set to null during update

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

#### *p\_phone\_rec record type*

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
phone_calling_calen dar	IN	VARCHAR2	No	
last_contact_dt_time	IN	DATE	No	
timezone_id	IN	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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: <ul style="list-style-type: none"> <li>• Validated against AR lookup type PHONE_LINE_TYPE.</li> <li>• Cannot be updated to null</li> </ul>
raw_phone_number	IN	VARCHAR2	No	Validation: Mandatory if phone_number is not passed in

If you use generic validations, validations only apply when contact\_point\_type = TELEX.

#### ***p\_telex\_rec record type***

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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***

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
web_type	IN	VARCHAR2	No	Validation: Cannot be set to null during update

---

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
url	IN	VARCHAR2	No	Validation: Cannot be set to null during update

---

#### **p\_eft\_rec\_record\_type**

---

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

---

#### **Other Validations**

- The Primary\_flag is a lookup code of lookup type YES/NO. The API automatically marks the first 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.



---

# Relationship and Hierarchy API Use

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_REC_TYPE,  
    x_relationship_type_id    OUT     NUMBER,  
    x_return_status           OUT     VARCHAR2,  
    x_msg_count               OUT     NUMBER,  
    x_msg_data                OUT     VARCHAR2,  
)
```

## Java Method

```
public static void createRelationshipType(
    OracleConnection_connection,
    String 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	VARCHAR 2	Yes	Validation: Mandatory attribute
forward_rel_code	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against AR lookup type PARTY_RELATIONS_TYPE</li></ul>
backward_rel_code	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against AR lookup type PARTY_RELATIONS_TYPE</li></ul>
direction_code	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against AR lookup type DIRECTION_CODE</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
hierarchical_flag	IN	VARCHAR 2	No	<p>Validation: Validated against AR lookup type YES/NO.</p> <ul style="list-style-type: none"> <li>• If hierarchical_flag = Y, then allow_circular_relationships must be N.</li> <li>• If hierarchical_flag = Y, then direction_code must be P or C</li> <li>• All relationship type records with same relationship_type value must have same value for hierarchical_flag.</li> </ul> <p>Default : N</p> <p>Comment : Indicates whether the relationship type is hierarchical.</p>
create_party_flag	IN	VARCHAR 2	No	<p>Validation: Validated against AR lookup type YES/NO</p> <p>Default: N</p> <p>Comment: Indicates whether a denormalized party will be created for a relationship having this relationship type</p>
allow_relate_to_self_flag	IN	VARCHAR 2	No	<p>Validation: Validated against AR lookup type YES/NO</p> <p>Default: N</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
allow_circular_relationships	IN	VARCHAR 2	No	<p>Validation: Validated against AR lookup type YES/NO</p> <ul style="list-style-type: none"> <li>If hierarchical_flag = Y, then allow_circular_relationships must be N.</li> <li>If direction_code = N, the allow_circular_relationships must be Y.</li> <li>All relationship type records with same relationship_type value must have same value for allow_circular_relationships.</li> </ul> <p>Default: Y</p> <p>7</p>
subject_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <p>Mandatory attribute</p> <p>Foreign key to fnd_object_instance_sets. instance_set_name</p>
object_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <p>Mandatory attribute</p> <p>Foreign key to fnd_object_instance_sets. instance_set_name</p>
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type CODE_STATUS
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
multiple_parent_allowed	IN	VARCHAR 2	No	<p>Validation : Validated against AR lookup type YES/NO</p> <ul style="list-style-type: none"> <li>• All relationship type records with same relationship_type value must have same value for multiple_parent_allowed.</li> </ul> <p>Comment : Indicates whether a child can have multiple parents. This is applicable when hierarchical_flag is Y.</p> <p>Default : Y</p>
incl_unrelated_entities	IN	VARCHAR 2	No	<p>Validation :Validated against AR lookup type YES/NO</p> <p>Comment : Indicates whether entities not having a relationship will be included in a hierarchy or not. This is applicable when hierarchical_flag is Y.</p> <p>Default : N</p>
forward_role	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type.</li> <li>• Forward_role should be unique.</li> </ul> <p>Default : Defaulted to 'USER_ROLE_'  to_char(relationship_type_id)</p> <p>Comment : This describes the role a subject party plays in a relationship.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
backward_role	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup values for the HZ_RELATIONSHIP_ROLE lookup type.</li> <li>• Backward_role should be unique.</li> </ul> <p>Comment: This describes the role an object party plays in a relationship.</p> <p>Default: Defaulted to 'USER_ROLE_''  to_char(relationship_type_id)</p>
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
create_party_flag	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Can be updated. No relationship record, created with the current setup of create_party_flag, can exist with this relationship type.</li> <li>• Validate against AR lookup type YES/NO.</li> </ul>
allow_relate_to_self_flag	IN	VARCHAR2	No	Validation: Not updateable
allow_circular_relationships	IN	VARCHAR2	No	Validation: Not updateable
subject_type	IN	VARCHAR2	No	Validation: Not updateable
object_type	IN	VARCHAR2	No	Validation: Not updateable
status	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CODE_STATUS</li> <li>• Cannot set to null during update</li> </ul>
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
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

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

### Other Validations

- The combination of relationship type, forward relationship code, backward relationship code, subject\_type, and object\_type should be unique.
- The combination of 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_CONTENT_SOURCE_TYPE,
    attribute_category              VARCHAR2(30),
    attribute1                      VARCHAR2(150),
    attribute2                      VARCHAR2(150),
    attribute3                      VARCHAR2(150),
    attribute4                      VARCHAR2(150),
    attribute5                      VARCHAR2(150),
    attribute6                      VARCHAR2(150),
    attribute7                      VARCHAR2(150),
    attribute8                      VARCHAR2(150),
    attribute9                      VARCHAR2(150),
    attribute10                     VARCHAR2(150),
    attribute11                     VARCHAR2(150),
    attribute12                     VARCHAR2(150),
    attribute13                     VARCHAR2(150),
    attribute14                     VARCHAR2(150),
    attribute15                     VARCHAR2(150),
    attribute16                     VARCHAR2(150),
    attribute17                     VARCHAR2(150),
    attribute18                     VARCHAR2(150),
    attribute19                     VARCHAR2(150),
    attribute20                     VARCHAR2(150),
    created_by_module               VARCHAR2(150),
    application_id                 NUMBER,
    party_rec                      HZ_PARTY_V2PUB.PARTY_REC_TYPE:=
HZ_PARTY_V2PUB.G_MISS_PARTY_REC
    additional_information1        VARCHAR2(150),
    additional_information2        VARCHAR2(150),
    additional_information3        VARCHAR2(150),
    additional_information4        VARCHAR2(150),
    additional_information5        VARCHAR2(150),
    additional_information6        VARCHAR2(150),
    additional_information7        VARCHAR2(150),
    additional_information8        VARCHAR2(150),
    additional_information9        VARCHAR2(150),
    additional_information10       VARCHAR2(150),
    additional_information11       VARCHAR2(150),
    additional_information12       VARCHAR2(150),
    additional_information13       VARCHAR2(150),
    additional_information14       VARCHAR2(150),
    additional_information15       VARCHAR2(150),
    additional_information16       VARCHAR2(150),
    additional_information17       VARCHAR2(150),
    additional_information18       VARCHAR2(150),
    additional_information19       VARCHAR2(150),
    additional_information20       VARCHAR2(150),
    additional_information21       VARCHAR2(150),
```

```
additional_information22      VARCHAR2(150),
additional_information23      VARCHAR2(150),
additional_information24      VARCHAR2(150),
additional_information25      VARCHAR2(150),
additional_information26      VARCHAR2(150),
additional_information27      VARCHAR2(150),
additional_information28      VARCHAR2(150),
additional_information29      VARCHAR2(150),
additional_information30      VARCHAR2(150),
percentage_ownership        NUMBER
actual_content_source        VARCHAR2(30)
)
```

## Java Inner Class for Relationship

```
public static class RelationshipRec {
    public BigDecimal relationship_id;
    public BigDecimal subject_id;
    public String subject_type;
    public String subject_table_name;
    public String 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 String application_id;
    public HzPartyV2Pub.PartyRec party_rec;

    HzPartyV2Pub.PartyRec additional_information1;
    HzPartyV2Pub.PartyRec additional_information2;
    HzPartyV2Pub.PartyRec additional_information3;
    HzPartyV2Pub.PartyRec additional_information4;
    HzPartyV2Pub.PartyRec additional_information5;
    HzPartyV2Pub.PartyRec additional_information6;
    HzPartyV2Pub.PartyRec additional_information7;
    HzPartyV2Pub.PartyRec additional_information8;
    HzPartyV2Pub.PartyRec additional_information9;
    HzPartyV2Pub.PartyRec additional_information10;
    HzPartyV2Pub.PartyRec additional_information11;
    HzPartyV2Pub.PartyRec additional_information12;
    HzPartyV2Pub.PartyRec additional_information13;
    HzPartyV2Pub.PartyRec additional_information14;
    HzPartyV2Pub.PartyRec additional_information15;
    HzPartyV2Pub.PartyRec additional_information16;
    HzPartyV2Pub.PartyRec additional_information17;
    HzPartyV2Pub.PartyRec additional_information18;
    HzPartyV2Pub.PartyRec additional_information19;
    HzPartyV2Pub.PartyRec additional_information20;
    HzPartyV2Pub.PartyRec additional_information21;
    HzPartyV2Pub.PartyRec 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;
}

```

## 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	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Validated against Primary Key in <code>fnd_objects.obj_name</code> where <code>fnd_objects.object_id = fnd_object_instance_sets.object_id</code> and <code>fnd_object_instance_sets.instance_set_name= subject_type</code></li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
subject_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to fnd_object_instance_sets. instance_set_name</li> </ul>
subject_table_name	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to fnd_objects.obj_name.</li> </ul>
object_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• 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</li> </ul>
object_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to fnd_object_instance_sets. instance_set_name</li> </ul>
object_table_name	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign Key to fnd_objects.obj_name</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
relationship_code	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against AR lookup type PARTY_RELATIONS_TYPE</li> <li>• Required to be a valid relationship code for the particular relationship type requested.</li> </ul>
relationship_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to hz_relationship_types. relationship_type</li> </ul>
comments	IN	VARCHAR 2	No	
start_date	IN	DATE	Yes	<p>Validation: Must be less than end_date if end_date is passed</p> <p>Default: sysdate</p>
end_date	IN	DATE	No	<p>Default: 31-DEC-4712</p> <p>Validation: Must be greater than start_date</p>
status	IN	VARCHAR 2	No	<p>Validation: Validated against AR lookup type REGISTRY_STATUS Default: A</p>
content_source_type	IN	VARCHAR 2	No	<p>Comment: This parameter is no longer used. Use actual_content_source.</p> <p>Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.</p> <p>Default: USER_ENTERED</p>
attribute_category	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
additional_information1	IN	VARCHAR 2	No	
additional_information2	IN	VARCHAR 2	No	
additional_information3	IN	VARCHAR 2	No	
additional_information4	IN	VARCHAR 2	No	
additional_information5	IN	VARCHAR 2	No	
additional_information6	IN	VARCHAR 2	No	
additional_information7	IN	VARCHAR 2	No	
additional_information8	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_informati on9	IN	VARCHAR 2	No	
additional_informati on10	IN	VARCHAR 2	No	
additional_informati on11	IN	VARCHAR 2	No	
additional_informati on12	IN	VARCHAR 2	No	
additional_informati on13	IN	VARCHAR 2	No	
additional_informati on14	IN	VARCHAR 2	No	
additional_informati on15	IN	VARCHAR 2	No	
additional_informati on16	IN	VARCHAR 2	No	
additional_informati on17	IN	VARCHAR 2	No	
additional_informati on18	IN	VARCHAR 2	No	
additional_informati on19	IN	VARCHAR 2	No	
additional_informati on20	IN	VARCHAR 2	No	
additional_informati on21	IN	VARCHAR 2	No	
additional_informati on22	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_informati on23	IN	VARCHAR 2	No	
additional_informati on24	IN	VARCHAR 2	No	
additional_informati on25	IN	VARCHAR 2	No	
additional_informati on26	IN	VARCHAR 2	No	
additional_informati on27	IN	VARCHAR 2	No	
additional_informati on28	IN	VARCHAR 2	No	
additional_informati on29	IN	VARCHAR 2	No	
additional_informati on30	IN	VARCHAR 2	No	
percentage_ownersh ip	IN	NUMBER	No	
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
actual_content_sour ce	IN	VARCHAR 2	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	VARCHAR 2	Yes/No	Validation: Mandatory if HZ_GENERATE_PARTY_NUMBER=N, else generated by sequence
validated_flag	IN	VARCHAR 2	No	Default: N
orig_system_reference	IN	VARCHAR 2	No	Default: party_id
orig_system	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
status	IN	VARCHAR 2	No	Validation: Validated against AR lookup type REGISTRY_STATUS Default: A
category_code	IN	VARCHAR 2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	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	VARCHAR 2	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 2<sup>nd</sup> relationship code from the HZ\_RELATIONSHIP\_TYPES table, and use it to create a second relationship. The 2<sup>nd</sup> relationship will have the same relationship\_id and party\_id as the first relationship. The requested relationship will be created with directional\_code value *F* (meaning forward) and the reciprocal relationship will be created with directional\_code value *B* (meaning backward).
- If a hierarchical relationship type is used, the API ensures no circular relationship is created during the creation of a relationship. Usually the API also ensures that a child has only one parent at a time. However, if the attribute multiple\_parents\_allowed = *Y*, then you can create multiple parents for a child. This is a very special case and you must cautiously use this attribute setting.
- If a hierarchical relationship type is used, API denormalizes the relationship information into HZ\_HIERARCHY\_NODES table. This denormalized table contains relationship information along with level, date effectiveness of relationships etc. Simple query can be written against that table to query hierarchy information.

## Update Relationship API

### Description

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

### PL/SQL Procedure

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

### Java Method

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

### Parameter Description and Validation

The following 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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
subject_id	IN	NUMBER	No	Validation: Not updateable
subject_type	IN	VARCHAR 2	No	Validation: Not updateable
subject_table_name	IN	VARCHAR 2	No	Validation: Not updateable
object_id	IN	NUMBER	No	Validation: Not updateable
object_type	IN	VARCHAR 2	No	Validation: Not updateable
object_table_name	IN	VARCHAR 2	No	Validation: Not updateable
relationship_code	IN	VARCHAR 2	No	Validation: Not updateable
relationship_type	IN	VARCHAR 2	No	Validation: Not updateable
comments	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS</li><li>• Cannot be updated to null</li></ul>
content_source_type	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use actual_content_source. Validation: Cannot be updated.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
additional_informati on1	IN	VARCHAR 2	No	
additional_informati on2	IN	VARCHAR 2	No	
additional_informati on3	IN	VARCHAR 2	No	
additional_informati on4	IN	VARCHAR 2	No	
additional_informati on5	IN	VARCHAR 2	No	
additional_informati on6	IN	VARCHAR 2	No	
additional_informati on7	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_informati on8	IN	VARCHAR 2	No	
additional_informati on9	IN	VARCHAR 2	No	
additional_informati on10	IN	VARCHAR 2	No	
additional_informati on11	IN	VARCHAR 2	No	
additional_informati on12	IN	VARCHAR 2	No	
additional_informati on13	IN	VARCHAR 2	No	
additional_informati on14	IN	VARCHAR 2	No	
additional_informati on15	IN	VARCHAR 2	No	
additional_informati on16	IN	VARCHAR 2	No	
additional_informati on17	IN	VARCHAR 2	No	
additional_informati on18	IN	VARCHAR 2	No	
additional_informati on19	IN	VARCHAR 2	No	
additional_informati on20	IN	VARCHAR 2	No	
additional_informati on21	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
additional_informati on22	IN	VARCHAR 2	No	
additional_informati on23	IN	VARCHAR 2	No	
additional_informati on24	IN	VARCHAR 2	No	
additional_informati on25	IN	VARCHAR 2	No	
additional_informati on26	IN	VARCHAR 2	No	
additional_informati on27	IN	VARCHAR 2	No	
additional_informati on28	IN	VARCHAR 2	No	
additional_informati on29	IN	VARCHAR 2	No	
additional_informati on30	IN	VARCHAR 2	No	
percentage_ownersh ip	IN	NUMBER	No	
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
actual_content_sour ce	IN	VARCHAR 2	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	<p>Validation: Valid party_id should be passed in to update party sequence</p> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the party_id from the hz_parties record</li> <li>• Pass only if you want to update the party record</li> </ul>
party_number	IN	VARCHAR 2	No	Validation: Not updateable
validated_flag	IN	VARCHAR 2	No	Validation: Not updateable
orig_system_reference	IN	VARCHAR 2	No	Validation: Not updateable
orig_system	IN	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
status	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type REGISTRY_STATUS</li> <li>• Cannot be updated to null</li> </ul>
category_code	IN	VARCHAR 2	No	Validation: Validated against lookup type CUSTOMER_CATEGORY
salutation	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
p_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute for relationship record</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number from hz_relationships table</li> <li>• Return new value after update</li> </ul>

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

## Other Validations

- You are required to specify a subject id, an object id, a subject type, an object type, a subject table name (where the subject belongs to e.g. 'HZ\_PARTIES' for subject type of 'PERSON', 'ORGANIZATION'), an object table name, a relationship type, a relationship code, and start date to create a relationship.
- The relationship code must be either a forward or backward relationship code defined in active records in the HZ\_RELATIONSHIP\_TYPES table for your relationship type.
- The end date will be default to '31-DEC-4712' if not specified. The end date must be greater than the start date.
- For a given subject\_id, object\_id, relationship\_code, there can be no overlap of the start\_date and the end\_date.
- The subject type and the object type will be used to do the foreign key check for the subject id and the object id column. The subject id should be a valid object instance defined in the object instance set which is specified as the subject type. The object id should be a valid object instance defined in the object instance set which is specified as the object type.
- The subject id and the object id cannot be the same value unless the self related flag of the relationship type is set to yes.
- It will check the "denormalized to party" flag in the HZ\_RELATIONSHIP\_TYPES

table to see if a denormalized party is necessary. If the flag is set to yes, a party will be created for the relationship. Only the relationship with both subject\_table\_name and object\_table\_name as 'HZ\_PARTIES' can be denormalized to HZ\_PARTIES table.

- A record will be created in the HZ\_RELATIONSHIPS table with the relationship code passed as input parameter.
- It will find the 2<sup>nd</sup> relationship code from the HZ\_RELATIONSHIP\_TYPES table, and use it to create a second relationship. The 2<sup>nd</sup> relationship will have the same relationship\_id and party\_id as the first relationship. The requested relationship will be created with directional\_code value 'F' (meaning forward) and the reciprocal relationship will be created with directional\_code value 'B' (meaning backward).
- Information in the Create Relationship API section about hierarchical types is also valid for the Update Relationship API section.

## 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 INDEX 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	VARCHAR 2	No	Comment : Indicates whether message stack should initialized Default : FND_API.G_FALSE
p_hierarchy_type	IN	VARCHAR 2	Yes	Validation : Must be a hierarchical relationship type from hz_relationship_types table
p_parent_id	IN	NUMBER	Yes	
p_parent_table_name	IN	VARCHAR 2	No	Default : 'HZ_PARTIES'
p_parent_object_type	IN	VARCHAR 2	No	Default : 'ORGANIZATION'
p_child_id	IN	NUMBER	Yes	
p_child_table_name	IN	VARCHAR 2	No	Default : 'HZ_PARTIES'
p_child_object_type	IN	VARCHAR 2	No	Default : 'ORGANIZATION'
p_effective_date	IN	DATE	No	Default : SYSDATE
p_include_node	IN	VARCHAR 2	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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
x_result	OUT	VARCHAR 2	Yes	Comment : Y/N
x_return_status	OUT	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	Yes	Comment : Message text if x_msg_count = 1

#### ***Record related\_nodes\_list\_rec***

<b>Parameter</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
related_node_id		NUMBER		Id of the related entity
related_node_table_name		VARCHAR 2		Table name for the related entity
related_node_object_type		VARCHAR 2		Object type of the related entity
level_number		NUMBER		Level at which the entity is related to
top_parent_flag		VARCHAR 2		Indicates whether related entity is top parent
leaf_child_flag		VARCHAR 2		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 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
  code_assignment_id
  owner_table_name
  owner_table_id
  owner_table_key_1
  owner_table_key_2
  owner_table_key_3
  owner_table_key_4
  owner_table_key_5
  class_category
  class_code
  primary_flag
  content_source_type
G_MISS_
  start_date_active
  end_date_active
  status
  created_by_module
  application_id
  rank
)
IS RECORD (
  NUMBER,
  VARCHAR2(30),
  NUMBER,
  VARCHAR2(255)
  VARCHAR2(255)
  VARCHAR2(255)
  VARCHAR2(255)
  VARCHAR2(255)
  VARCHAR2(30),
  VARCHAR2(30),
  VARCHAR2(1),
  VARCHAR2(30) := HZ_PARTY_V2PUB.

CONTENT_SOURCE_TYPE,
DATE,
DATE,
VARCHAR2(1),
VARCHAR2(150),
NUMBER
NUMBER
```

## PL/SQL Record Structure for Class Category Use

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

## Java Inner Class for Class Category

```
public static class ClassCategoryRec {
  public String
  public String
  public String
  public String
  public String
  public BigDecimal
  public String

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

class\_category;
allow\_multi\_parent\_flag;
allow\_multi\_assign\_flag;
allow\_leaf\_node\_only\_flag;
created\_by\_module;
application\_id;
delimiter

## Java Inner Class for Class Code Relation

```
public static class ClassCodeRelationRec {
  public String
  public String
  public String
  public java.sql.Timestamp
  public java.sql.Timestamp
  public String
  public BigDecimal

  class_category;
  class_code;
  sub_class_code;
  start_date_active;
  end_date_active;
  created_by_module;
  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	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Unique</li><li>• A Class Category correspond to a LOOKUP_TYPE</li></ul>
allow_multi_parent_flag	IN	VARCHAR 2	No	Validation: Validated against FND lookup type YES/NO
allow_multi_assign_flag	IN	VARCHAR 2	No	Validation: Validated against FND lookup type YES/NO
allow_leaf_node_only_flag	IN	VARCHAR 2	No	Validation: Validated against FND lookup type YES/NO
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.

Parameter Name	Type	Data Type	Required	ValidationDefaultComment
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
delimiter	IN	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
class_category	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Unique</li> <li>• A Class Category correspond to a LOOKUP_TYPE</li> </ul>
allow_multi_parent_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup type YES/NO</li> <li>• This flag cannot be updated from Y to N, if there are any class codes related to more than one parent code</li> </ul>
allow_multi_assign_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup type YES/NO</li> <li>• This flag cannot be updated from Y to N, if there are any class codes assigned to more than one instance of HZ_PARTIES or HZ_RELATIONSHIP_TYPES or any entities that uses the classification model.</li> </ul>
allow_leaf_node_only_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup type YES/NO</li> <li>• This flag cannot be updated from Y to N, if there are any class codes assigned to more than one instance of HZ_PARTIES or HZ_RELATIONSHIP_TYPES or any entities that uses the classification model.</li> </ul>
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
delimiter	IN	VARCHAR 2	No	Comment: Used for concatenated class code in the HZ_CLASS_CODE_DENORM table.  Default value is / .
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against value in the database for the existing record</li></ul> Comment: <ul style="list-style-type: none"><li>• Pass the current object_version_number from HZ_CLASS_CATEGORIES</li><li>• Return new value after update</li></ul>

## Create Class Code Relation API

### Description

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

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

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

## PL/SQL Procedure

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

## Java Method

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

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
class_category	IN	VARCHAR 2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES.CLASS_CATEGORY
class_code	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column</li><li>• sub_class_code cannot be an ancestor code of class_code. This validation avoids recursive relationship</li></ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
sub_class_code	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup values where the LOOKUP_TYPE equals to the value in the CLASS_CATEGORY column</li> <li>• class_code cannot be a descendant of sub_class_code. This validation avoid recursive relationship</li> <li>• If allow_multi_parent_flag = N then the sub_class_code must not have any class_code related to it for the period range from start_date_active to end_date_active.</li> </ul>
start_date_active	IN	DATE	No	
end_date_active	IN	DATE	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• A Class Code Relation must be unique for a period of time range from start_date_active to end_date_active.</li> <li>• The end_date_active must be null or greater than the start_date_active.</li> </ul>
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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	VARCHAR 2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES
class_code	IN	VARCHAR 2	Yes	Validation: Validated the existence of the relation (class_category, class_code, sub_class_code, start_date_active)
sub_class_code	IN	VARCHAR 2	Yes	Validation: Part of the existence validation
start_date_active	IN	DATE	No	Validation: Part of the existence validation

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
end_date_active	IN	DATE	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>A Class Code Relation (class_category, class_code, sub_class_Code) must be unique for a period of time range from Start_date_Active to End_Date_Active.</li> <li>End_date_Active must be null or greater than Start_Date_Active.</li> </ul>
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>Mandatory attribute</li> <li>Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>Pass the current object_version_number of the record from hz_class_code_relations</li> <li>Returns new value after update</li> </ul>

## Create Code Assignment API

### Description

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

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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
owner_table_key_1	IN	VARCHAR 2	No	<p>Validation:</p> <p>One or the other of owner_table_id or owner_table_key_1 must be supplied. Both cannot be supplied.</p> <p>If supplied the value must correspond to the primary key value of the entity which is being classified.</p> <p>Not updateable.</p>
owner_table_key_2	IN	VARCHAR 2	No	<p>Validation:</p> <p>If owner_table_key_1 is not supplied, then this parameter cannot be supplied.</p> <p>If supplied the value must correspond to the primary key value of the entity which is being classified.</p> <p>Not updateable.</p>
owner_table_key_3	IN	VARCHAR 2	No	<p>Validation:</p> <p>If owner_table_key_2 is not supplied, then this parameter cannot be supplied.</p> <p>If supplied the value must correspond to the primary key value of the entity which is being classified.</p> <p>Not updateable.</p>
owner_table_key_4	IN	VARCHAR 2	No	<p>Validation:</p> <p>If owner_table_key_3 is not supplied, then this parameter cannot be supplied.</p> <p>If supplied the value must correspond to the primary key value of the entity which is being classified.</p> <p>Not updateable.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
owner_table_key_5	IN	VARCHAR 2	No	<p>Validation:</p> <p>If owner_table_key_4 is not supplied, then this parameter cannot be supplied.</p> <p>If supplied the value must correspond to the primary key value of the entity which is being classified.</p> <p>Not updateable.</p>
class_category	IN	VARCHAR 2	No	Validation: Validated against HZ_CLASS_CATEGORIES
class_code	IN	VARCHAR 2	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	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup values where lookup type YES/NO</li> <li>• An owner_table_id can only have one primary (PRIMARY_FLAG = Y) assignment to a class_code of one class_category for one actual_content_source at one time.</li> </ul>
content_source_type	IN	VARCHAR 2	No	<p>Comment: This parameter is no longer used. Use actual_content_source.</p> <p>Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of Y.</p> <p>Default: USER_ENTERED</p>
start_date_active	IN	DATE	No	
end_date_active	IN	DATE	No	Validation: must be null or greater than start_date_active
Status	IN	VARCHAR 2	No	Validation: Validated against AR_LOOKUP type CODE_STATUS

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate module from which creation of record is initiated
actual_content_source	IN	VARCHAR 2	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_ASSIGNMENT_REC_TYPE,
    p_object_version_number               IN OUT     NUMBER,
    x_return_status                       OUT         VARCHAR2,
    x_msg_count                           OUT         NUMBER,
    x_msg_data                            OUT         VARCHAR2
)

```

## Java Method

```
public static void updateCodeAssignment(
    OracleConnection_connection,
    String 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_ASSIGNMENTS. CODE_ASSIGNMENT_ID
owner_table_name	IN	VARCHAR 2	No	
owner_table_id	IN	NUMBER	No	
class_category	IN	VARCHAR 2	Yes	
class_code	IN	VARCHAR 2	No	
primary_flag	IN	VARCHAR 2	No	
content_source_type	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Use actual_content_source.
start_date_active	IN	DATE	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	VARCHAR 2	No	Validation: Validated against AR_LOOKUP type CODE_STATUS
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Validated against value in the database for the existing records</li></ul> Comment: <ul style="list-style-type: none"><li>• Pass the current object_version_number of the record from hz_code_assignments</li><li>• Returns new value after update</li></ul>
rank	IN	NUMBER	No	Comment: Provides the ability to rank classes if multiple classes have been assigned to a party.

## Create Class Category Use API

### Description

This routine is used to create a Class Category Use. The API creates a record in the HZ\_CLASS\_CATEGORYUSES table. The classification model is an open structure, the HZ\_CLASS\_CATEGORYUSES table indicates which tables or subsets of tables, use which classifications. The HZ\_CLASS\_CATEGORYUSES 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	VARCHAR 2	Yes	Validation: Validated against HZ_CLASS_CATEGORIES
owner_table	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"> <li>• Validated against FND lookup values where Lookup type = CODE_ASSIGN_OWNER_TABLE</li> <li>• The combination (class_category, owner_table) must be unique.</li> </ul>
column_name	IN	VARCHAR 2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
additional_where_clause	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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\_CATEGORYUSES table.

### PL/SQL Procedure

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

### Java Method

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

### Parameter Description and Validation

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

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

## Is Valid Category API

### Description

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

the function returns  $F$ .

## PL/SQL Function

```
FUNCTION is_valid_category(
    p_owner_table                      IN          VARCHAR2,
    p_class_category                     IN          VARCHAR2,
    p_id                                IN          NUMBER,
    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	VARCHAR 2	Yes	
p_owner_table	IN	VARCHAR 2	Yes	
p_id	IN	VARCHAR 2	Yes	Default: NULL
p_key_1	IN	VARCHAR 2		Default: NULL
p_key_2	IN	VARCHAR 2		Default: NULL



---

## Contact Preference API Use

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 String preference_topic_type_id;  
    public String preference_topic_type_code;  
    public String preference_start_date;  
    public String 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 String 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	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Non updateable</li><li>• Validated against AR lookup type SUBJECT_TABLE</li></ul>
contact_level_table_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Not updateable</li><li>• Is the foreign key of contact_level_table</li></ul>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_type	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory Attribute.</li> <li>• Non-updateable</li> <li>• contact_type is lookup code in lookup type CONTACT_TYPE</li> </ul>
preference_code	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Should be validated against the PREFERENCE_CODE lookup type</li> </ul>
preference_topic_type	IN	VARCHAR 2	N	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Should be validated against the PREFERENCE_TOPIC_TYPE lookup type.</li> <li>• The lookup contains the following lookup_codes against which the PREFERENCE_TOPIC_TYPE will be validated:</li> </ul> <p>TABLES:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES,</li> <li>• AS_INTEREST_TYPES_B,</li> <li>• AS_INTEREST_CODES_B</li> <li>• LOOKUP_TYPE: CONTACT_USAGE</li> </ul>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_topic_type_id	IN	NUMBER	N	<p>Validation :</p> <p>Valid if the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES</li> <li>• AS_INTEREST_TYPES_B</li> <li>• AS_INTEREST_CODES_B</li> </ul> <p>The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected PREFERENCE_TOPIC_TYPE.</p>
preference_topic_type_code	IN	VARCHAR 2	N	<p>Validation:</p> <p>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.</p> <p>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.</p>
preference_start_date	IN	DATE	Y	<p>Validation:</p> <p>Mandatory attribute.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
preference_end_date	IN	DATE	N	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot be updated to a day before the sysdate</li> <li>• PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE</li> <li>• If the STATUS column in the HZ_CONTACT_POINTS table is set to a value other than A for Active (such as I for Inactive, M for Merged, or D for deleted), then PREFERENCE_END_DATE is not passed in these situations, it should default to the system date. If a value other than the system date is passed, it should fail.</li> </ul>
preference_start_time_hr	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_end_time_hr	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_start_time_mi	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 59 minute format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_end_time_mi	IN	NUMBER	N	<p>Validation:</p> <p>In 0 to 59 minute format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
max_no_of_interactions	IN	NUMBER	N	Validation: none
max_no_of_interact uom_code	IN	VARCHAR 2	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	VARCHAR 2	Y	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Should be validated against the REQUESTED_BY lookup type</li> </ul>
reason_code	IN	VARCHAR 2	N	Validation : Validated against the REASON_CODE lookup type.
status	IN	VARCHAR 2	N	<p>Validation :</p> <p>Should be validated against the CODE_STATUS lookup type.</p> <p>The PREFERENCE_END_DATE attribute should be set to the sysdate when STATUS has a value other than A.</p>
created_by_module	IN	VARCHAR 2	Y	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	N	Comment: Text to indicate application from which creation of record is initiated.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_contact_preference_id	OUT	NUMBER	N	Comment: Returns contact_preference_id of the record created.

## Other Validations

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

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

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

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

## Update Contact Preference API

### Description

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

### PL/SQL Procedure

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

### Java Method

```
public static void updateContactPreference(
    OracleConnection_connection,
    String
    ContactPreferenceRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) 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.

---

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

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_topic_typ e_id	IN	NUMBER	No	<p>Validation:</p> <p>If the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES</li> <li>• AS_INTEREST_TYPES_B</li> <li>• AS_INTEREST_CODES_B</li> </ul> <p>The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected in the attribute PREFERENCE_TOPIC_TYPE.</p>
preference_topic_typ e_code	IN	VARCHAR2	No	<p>Validation:</p> <p>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.</p> <p>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.</p>
preference_start_dat e	IN	DATE	No	<p>Validation:</p> <p>The value of the PREFERENCE_END_DATE attribute should be greater than or equal to the value of the PREFERENCE_START_DATE attribute.</p>
preference_end_date	IN	DATE	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• PREFERENCE_END_DATE can not update to a day before sysdate</li> <li>• PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE,</li> </ul>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_start_time_hr	IN	NUMBER	No	<p>Validation:</p> <p>In 0 to 24 hour format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_end_time_hr	IN	NUMBER	No	<p>Validation:</p> <p>In 0 to 24 hour format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_start_time_mi	IN	NUMBER	No	<p>Validation:</p> <p>In 0 to 59 minute format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_end_time_mi	IN	NUMBER	No	<p>Validation:</p> <p>In 0 to 59 minute format</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
max_no_of_interactions	IN	NUMBER	No	Validation: none

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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: <ul style="list-style-type: none"><li>• Should be validated against the CODE_STATUS lookup type</li><li>• The PREFERENCE_END_DATE attribute should be set to sysdate when the STATUS column has a value other than A</li><li>• Can not be set to NULL during update</li></ul>
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists

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

## Other Validations

- When a contact point is created, PREFERENCE\_START\_DATE is defaulted to the system date if a value is not passed and the PREFERENCE\_CODE should be set to "Do" in HZ\_CONTACT\_PREFERENCES unless the party explicitly opts-out (in that case the PREFERENCE\_CODE should be set to "Do Not" and the PREFERENCE\_START\_DATE should be set to the system date if no date has been specified by the party)
- If the STATUS column in HZ\_CONTACT\_POINTS is set to a value other than "A" for "Active" such as "I" for Inactive, "M" for Merged, or "D" for Deleted, then PREFERENCE\_END\_DATE should be set. If a value for PREFERENCE\_END\_DATE is not passed in these situations, it should default to the system date. If a value other than the system date is passed, it should fail.
- If a value is passed for MAX\_NO\_OF\_INTERACT\_UOM\_CODE then the PREFERENCE\_START\_DATE should have a value. If a value is not passed for PREFERENCE\_START\_DATE it should default to the system date.
- Duplication check. uniquely identify a contact preference to prevent duplication record to be created in HZ\_CONTACT\_PREFERENCES by checking the follow columns:
  - contact\_level\_table
  - contact\_level\_table\_id
  - contact\_type

- preference\_topic\_type
- preference\_topic\_type\_id
- preference\_topic\_type\_code
- When the CONTACT\_LEVEL\_TABLE attribute is HZ\_CONTACT\_POINTS, the CONTACT\_TYPE cannot be MAIL or VISIT.
- CONTACT\_TYPE lookup should be validated against the HZ\_CONTACT\_POINTS.CONTACT\_POINT\_TYPE column based on the value passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE. So if the value passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE is HZ\_CONTACT\_POINTS and the HZ\_CONTACT\_LEVEL\_TABLE\_ID signifies that the contact point is an email address, then the values that will be allowed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE will be 'EMAIL' only.
- The following table gives the values that can be passed in HZ\_CONTACT\_PREFERENCES.CONTACT\_TYPE based on the HZ\_CONTACT\_POINT.CONTACT\_POINT\_TYPE, HZ\_CONTACT\_PREFERENCES.CONTACT\_LEVEL\_TABLE and HZ\_CONTACT\_PREFERENCES/CONTACT\_LEVEL\_TABLE\_ID

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

---

<b>CONTACT_TYPE allowed</b>	<b>when CONTACT_POINT_TYPE is</b>	<b>and CONTACT_LEVEL_TABLE is</b>
	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 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),
    ceterminate_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,
federal_entity_type            VARCHAR2(30),
trading_partner_agency_id     VARCHAR2(3),
duns_extension                 VARCHAR2(4),
advance_payment_indicator     VARCHAR2(30),
cancel_unshipped_lines_flag   VARCHAR2(1)
)

```

## 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 String               primary_salesrep_id;
    public String               sales_channel_code;
    public String               order_type_id;
    public String               price_list_id;
    public String               tax_code;
    public String               fob_point;
    public String               freight_term;
    public String               ship_via;
    public String               warehouse_id;
    public String               tax_header_level_flag;
    public String               tax_rounding_rule;
    public String               ceterminate_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 boolean hold_bill_flag;
    public boolean realtime_rate_flag;
    public String acct_life_cycle_status;
    public String account_name;
    public String deposit_refund_method;
    public boolean dormant_account_flag;
    public String npa_number;
    public String 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 boolean ship_sets_include_lines_flag;
    public boolean arrivalsets_include_lines_flag;
    public boolean sched_date_push_flag;
    public String invoice_quantity_rule;
    public java.sql.Timestamp status_update_date;
    public boolean autopay_flag;
    public boolean notify_flag;
    public String last_batch_id;
    public String selling_party_id;
    public String created_by_module;
    public String application_id;
    public String federal_entity_type;
    public String trading_partner_agency_id;
    public String duns_extension;
    public boolean advance_payment_indicator;
    public boolean cancel_unshipped_lines_flag;
    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 String 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_TYPE,
    p_customer_profile_rec IN      HZ_CUSTOMER_PROFILE_V2PUB.
CUSTOMER_PROFILE_REC_TYPE,
    p_create_profile_amt IN      VARCHAR2 := FND_API.G_TRUE,
    x_cust_account_id    OUT     NUMBER,
    x_account_number     OUT     VARCHAR2,
    x_party_id           OUT     NUMBER,
    x_party_number        OUT     VARCHAR2,
    x_profile_id         OUT     NUMBER,
    x_return_status       OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data            OUT     VARCHAR2
)
)
```

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

## PL/SQL Procedure for Organization Account

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

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

### Java Method for Person Account

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

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

### Java Method for Organization Account

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

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

### Parameter Description and Validation

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

about the parameter such as validation, defaults, and other comments.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
cust_account_id	IN	NUMBER	N	Validation: unique if passed in, else generated from sequence
account_number	IN	VARCHAR 2	Y	Validation: <ul style="list-style-type: none"> <li>• Mandatory Attribute. If GENERATE_CUSTOMER_NUMBER of AR_SYSTEM_PARAMETERS is on, if user has passed in an account_number, error out</li> <li>• account_number will be generated from sequence. If autonumbering is off, if user has not passed in value, error out</li> </ul>
attribute_category	IN	VARCHAR 2	N	
attribute1	IN	VARCHAR 2	N	
attribute2	IN	VARCHAR 2	N	
attribute3	IN	VARCHAR 2	N	
attribute4	IN	VARCHAR 2	N	
attribute5	IN	VARCHAR 2	N	
attribute6	IN	VARCHAR 2	N	
attribute7	IN	VARCHAR 2	N	
attribute8	IN	VARCHAR 2	N	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute9	IN	VARCHAR 2	N	
attribute10	IN	VARCHAR 2	N	
attribute11	IN	VARCHAR 2	N	
attribute12	IN	VARCHAR 2	N	
attribute13	IN	VARCHAR 2	N	
attribute14	IN	VARCHAR 2	N	
attribute15	IN	VARCHAR 2	N	
attribute16	IN	VARCHAR 2	N	
attribute17	IN	VARCHAR 2	N	
attribute18	IN	VARCHAR 2	N	
attribute19	IN	VARCHAR 2	N	
attribute20	IN	VARCHAR 2	N	
global_attribute_cate gory	IN	VARCHAR 2	N	
global_attribute1	IN	VARCHAR 2	N	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute2	IN	VARCHAR 2	N	
global_attribute3	IN	VARCHAR 2	N	
global_attribute4	IN	VARCHAR 2	N	
global_attribute5	IN	VARCHAR 2	N	
global_attribute6	IN	VARCHAR 2	N	
global_attribute7	IN	VARCHAR 2	N	
global_attribute8	IN	VARCHAR 2	N	
global_attribute9	IN	VARCHAR 2	N	
global_attribute10	IN	VARCHAR 2	N	
global_attribute11	IN	VARCHAR 2	N	
global_attribute12	IN	VARCHAR 2	N	
global_attribute13	IN	VARCHAR 2	N	
global_attribute14	IN	VARCHAR 2	N	
global_attribute15	IN	VARCHAR 2	N	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute16	IN	VARCHAR 2	N	
global_attribute17	IN	VARCHAR 2	N	
global_attribute18	IN	VARCHAR 2	N	
global_attribute19	IN	VARCHAR 2	N	
global_attribute20	IN	VARCHAR 2	N	
orig_system_reference	IN	VARCHAR 2	N	Validation: unique if passed in Default: cust_account_id
orig_system	IN	VARCHAR 2	N	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.  Default: UNKNOWN if an orig_system_reference is passed in.
status	IN	VARCHAR 2	N	Validation: <ul style="list-style-type: none"><li>• status is lookup code in lookup type CODE_STATUS</li><li>• status cannot be set to null during update. It is defaulted to 'A' if user does not pass any value</li></ul>
customer_type	IN	VARCHAR 2	N	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE
customer_class_code	IN	VARCHAR 2	N	Validation: Validated against AR lookup type CUSTOMER CLASS
primary_salesrep_id	IN	NUMBER	N	Comment: This attribute is no longer used. Use hz_cust_site_uses.primary_salesrep_id instead.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
sales_channel_code	IN	VARCHAR 2	N	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups
order_type_id	IN	NUMBER	N	Comment: This attribute is no longer used. Use hz_cust_site_uses.order_type_id instead.
price_list_id	IN	NUMBER	N	Validation: Must be a valid price_list_id from SO_PRICE_LISTS table.
tax_code	IN	VARCHAR 2	N	Comment: This attribute is no longer used. Use hz_cust_site_uses.tax_code instead.
fob_point	IN	VARCHAR 2	N	Validation: Validated against AR lookup type FOB.
freight_term	IN	VARCHAR 2	N	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups
ship_partial	IN	VARCHAR 2	N	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	N	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
coterminate_day_month	IN	VARCHAR 2	N	
primary_specialist_id	IN	NUMBER	N	Validation: primary_specialist_id is foreign key to per_all_people_f

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
secondary_specialist_id	IN	NUMBER	N	Validation: secondary_specialist_id is foreign key to per_all_people_f
account_liable_flag	IN	VARCHAR 2		Comment: This parameter is no longer used.  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	Comment: This parameter is no longer used.
account_established_date	IN	DATE	N	
account_termination_date	IN	DATE	N	Comment: This parameter is no longer used.  Validation: <ul style="list-style-type: none"><li>• account_termination_date should be greater than account_established_date.</li><li>• account_termination_date should be greater than account_activation_date.</li></ul>
account_activation_date	IN	DATE	N	Comment: This parameter is no longer used.  Validation: account_activation_date should be greater than account_established_date.
department	IN	VARCHAR 2		Comment: This parameter is no longer used.
held_bill_expiration_date	IN	DATE	N	
hold_bill_flag	IN	VARCHAR 2		Validation: hold_bill_flag is lookup code in lookup type YES/NO  Default: N.
realtime_rate_flag	IN	VARCHAR 2		Comment: This parameter is no longer used.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
acct_life_cycle_status	IN	VARCHAR 2	N	Comment: This parameter is no longer used.
account_name	IN	VARCHAR 2	N	
deposit_refund_method	IN	VARCHAR 2	N	
dormant_account_flag	IN	VARCHAR 2	N	Comment: This parameter is no longer used. Validation: dormant_account_flag is lookup code in lookup type YES/NO. Default: N.
npa_number	IN	VARCHAR 2	N	
suspension_date	IN	DATE	N	Comment: This parameter is no longer used.
source_code	IN	VARCHAR 2	N	
comments	IN	VARCHAR 2	N	
dates_negative_tolerance	IN	NUMBER	N	
dates_positive_tolerance	IN	NUMBER	N	
date_type_preference	IN	VARCHAR 2	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
under_return_tolerance	IN	NUMBER	N	
item_cross_ref_pref	IN	VARCHAR 2	N	Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYPES
ship_sets_include_lines_flag	IN	VARCHAR 2	N	<p>Validation:</p> <ul style="list-style-type: none"> <li>• ship_sets_include_lines_flag is lookup code in lookup type YES/NO</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	IN	VARCHAR 2	N	<p>Default: N</p> <p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	IN	VARCHAR 2	N	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR 2	N	Validated against OE lookup type INVOICE_BASIS.
pricing_event	IN	VARCHAR 2	N	Comment: This attribute is no longer used.
status_update_date	IN	DATE	N	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
autopay_flag	IN	VARCHAR 2	N	Validation: autopay_flag is lookup code in lookup type YES/NO
notify_flag	IN	VARCHAR 2	N	Comment: This parameter is no longer used. 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	VARCHAR 2	Y	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	N	Comment: Text to indicate application from which creation of record is initiated
x_cust_account_id	OUT	NUMBER	N	Comment: Returns cust_account_id of the account record create
x_account_number	OUT	NUMBER	N	Comment: Returns account_number of the account record created
x_party_id	OUT	NUMBER	N	Comment: Returns party_id of the organization or person party created
x_party_number	OUT	NUMBER	N	Comment: Returns party_number of the organization or person created
x_profile_id	OUT	NUMBER	N	Comment: Returns profile_id of the organization or person profile created
federal_entity_type	IN	VARCHAR 2	N	Validation: Select lookup_code, description from fv_lookup_codes where lookup_type = 'FV_FED_NON_FED_CODE'; Comment: Identifies the type of entity involved in transactions with the reporting entity.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
trading_partner_agency_id	IN	VARCHAR 2	N	<p>Validation:</p> <ol style="list-style-type: none"> <li>1. If 'federal_entity_type' is NULL then 'trading_partner_agency_id' must not contain any value.</li> <li>2. If 'federal_entity_type' is NOT NULL and equal to 'F' (Federal) then Trading Partner Agency ID must have a value.</li> <li>3. Value must be from fv_tp_treasury_symbols.agency_id</li> </ol> <p>SELECT agency_id FROM fv_tp_treasury_symbols ORDER BY agency_id;</p> <p>Comment: ID of the federal agency involved in transactions with the reporting entity. This is required if Federal Entity Type is "F".</p>
duns_extension	IN	VARCHAR 2	N	Validation: Must contain four alphanumeric characters in uppercase, without spaces and underscores.
advance_payment_indicator	IN	VARCHAR 2	N	Validation: Lookup is Select lookup_code, description from fv_lookup_codes where lookup_type = 'FV_ADV_PMT_INDICATOR';
cancel_unshipped_lines_flag	IN	VARCHAR 2	N	<p>Validation: cancel_unshipped_lines_flag is lookup code in lookup type YES/NO</p> <p>Comment: Indicates whether to cancel all unshipped orders/lines.</p> <p>Default: Null</p>

## 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	VARCHAR 2	No	Validation: Not updateable if automatic numbering is on. If not, then account_number can be updated.
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	
global_attribute_cate gory	IN	VARCHAR 2	No	
global_attribute1	IN	VARCHAR 2	No	
global_attribute2	IN	VARCHAR 2	No	
global_attribute3	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute4	IN	VARCHAR 2	No	
global_attribute5	IN	VARCHAR 2	No	
global_attribute6	IN	VARCHAR 2	No	
global_attribute7	IN	VARCHAR 2	No	
global_attribute8	IN	VARCHAR 2	No	
global_attribute9	IN	VARCHAR 2	No	
global_attribute10	IN	VARCHAR 2	No	
global_attribute11	IN	VARCHAR 2	No	
global_attribute12	IN	VARCHAR 2	No	
global_attribute13	IN	VARCHAR 2	No	
global_attribute14	IN	VARCHAR 2	No	
global_attribute15	IN	VARCHAR 2	No	
global_attribute16	IN	VARCHAR 2	No	
global_attribute17	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute18	IN	VARCHAR 2	No	
global_attribute19	IN	VARCHAR 2	No	
global_attribute20	IN	VARCHAR 2	No	
orig_system_reference	IN	VARCHAR 2	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	VARCHAR 2	N	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
status	IN	VARCHAR 2	No	Validation: Cannot be set to null during update
customer_type	IN	VARCHAR 2	No	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE
customer_class_code	IN	VARCHAR 2	No	Validation: Validated against AR lookup type CUSTOMER CLASS
primary_salesrep_id	IN	NUMBER	No	Comment: This attribute is no longer used. Use hz_cust_site_uses.primary_salesrep_id instead.
sales_channel_code	IN	VARCHAR 2	No	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups
order_type_id	IN	NUMBER	No	Comment: This attribute is no longer used. Use hz_cust_site_uses.order_type_id instead.
price_list_id	IN	NUMBER	No	Validation: Must be valid price_list_id from SO_PRICE_LISTS.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
tax_code	IN	VARCHAR 2	No	Comment: This attribute is no longer used. Use hz_cust_site_uses.tax_code instead.
fob_point	IN	VARCHAR 2	No	Validated against AR lookup type FOB.
freight_term	IN	VARCHAR 2	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups
ship_partial	IN	VARCHAR 2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: tax_header_level_flag is lookup code in lookup type YES/NO
tax_rounding_rule	IN	VARCHAR 2	No	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
coterminate_day_month	IN	VARCHAR 2	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	VARCHAR 2	No	Comment: This parameter is no longer used. 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	No	Comment: This parameter is no longer used.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
account_established_date	IN	DATE	No	
account_termination_date	IN	DATE	No	Comment: This parameter is no longer used. Validation: <ul style="list-style-type: none"><li>• account_termination_date should be greater than account_established_date.</li><li>• account_termination_date should be greater than account_activation_date.</li></ul>
account_activation_date	IN	DATE	No	Comment: This parameter is no longer used. Validation: account_activation_date should be greater than account_established_date.
department	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
held_bill_expiration_date	IN	DATE	No	
hold_bill_flag	IN	VARCHAR 2	No	Validation: hold_bill_flag is lookup code in lookup type YES/NO
realtime_rate_flag	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
acct_life_cycle_status	IN	VARCHAR 2	No	Comment: This parameter is no longer used.
account_name	IN	VARCHAR 2	No	
deposit_refund_met_hod	IN	VARCHAR 2	No	
dormant_account_flag	IN	VARCHAR 2	No	Comment: This parameter is no longer used. Validation: dormant_account_flag is lookup code in lookup type YES/NO. Default: N.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
npa_number	IN	VARCHAR 2	No	
suspension_date	IN	DATE	No	Comment: This parameter is no longer used.
source_code	IN	VARCHAR 2	No	
comments	IN	VARCHAR 2	No	
dates_negative_tolerance	IN	NUMBER	No	
dates_positive_tolerance	IN	NUMBER	No	
date_type_preference	IN	VARCHAR 2	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	
item_cross_ref_pref	IN	VARCHAR 2	No	Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYPES.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_sets_include_lines_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	IN	VARCHAR 2	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR 2	No	Validation: Validated against OE lookup type INVOICE_BASIS.
pricing_event	IN	VARCHAR 2	No	Comment: This attribute is no longer used.
status_update_date	IN	DATE	No	
autopay_flag	IN	VARCHAR 2	No	Validation: autopay_flag is lookup code in lookup type YES/NO
notify_flag	IN	VARCHAR 2	No	<p>Comment: This parameter is no longer used.</p> <p>Validation: notify_flag is lookup code in lookup type YES/NO.</p>
last_batch_id	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
selling_party_id	IN	NUMBER	No	Validation: selling_party_id should point to a organization party
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists.
p_object_version_number	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record.</li> </ul> <p>Comment</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number of the customer account record</li> <li>• Return new value after update.</li> </ul>
federal_entity_type	IN	VARCHAR 2	No	<p>Validation: Select lookup_code, description from fv_lookup_codes where lookup_type = 'FV_FED_NON_FED_CODE';</p> <p>Comment: Identifies the type of entity involved in transactions with the reporting entity.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
trading_partner_age ncy_id	IN	VARCHAR 2	No	<p>Validation:</p> <ol style="list-style-type: none"> <li>1. If 'federal_entity_type' is NULL then 'trading_partner_agency_id' must not contain any value.</li> <li>2. If 'federal_entity_type' is NOT NULL and equal to 'F' (Federal) then Trading Partner Agency ID must have a value.</li> <li>3. Value must be from fv_tp_treasury_symbols. agency_id</li> </ol> <p>Comment: ID of the federal agency involved in transactions with the reporting entity. This is required if Federal Entity Type is "F".</p>
duns_extension	IN	VARCHAR 2	No	Validation: Must contain four alphanumeric characters in uppercase, without spaces and underscores.
advance_payment_i ndicator	IN	VARCHAR 2	No	Validation: Lookup is Select lookup_code, description from fv_lookup_codes where lookup_type = 'FV_ADV_PMT_INDICATOR';
cancel_unshipped_li nes_flag	IN	VARCHAR 2	No	<p>Validation: cancel_unshipped_lines_flag is lookup code in lookup type YES/NO</p> <p>Comment: Indicates whether to cancel all unshipped orders/lines.</p> <p>Default: Null</p>

## Create Customer Account Relationship API

### Description

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

## PL/SQL Procedure

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

## Java Method

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

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_id	IN	NUMBER	Yes	Validation: Mandatory attribute Comment: Pass cust_account_id hz_cust_accounts
related_cust_account_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• related_cust_account_id is foreign key of hz_cust_accounts</li></ul>
relationship_type	IN	VARCHAR 2	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE
comments	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
customer_reciprocal_flag	IN	VARCHAR 2	No	Validation: customer_reciprocal_flag is lookup code in lookup type YES/NO  Default: N
status	IN	VARCHAR 2	No	Validation: Status is lookup code in lookup type CODE_STATUS  Default: A

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
bill_to_flag	IN	VARCHAR 2	No	
ship_to_flag	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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_RELATE_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
relationship_type	IN	VARCHAR2	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE
comments	IN	VARCHAR2	No	

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute15	IN	VARCHAR2	No	
bill_to_flag	IN	VARCHAR2	No	
ship_to_flag	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists.
p_rowid	IN	ROWID	No	Comment: This is only in the overloaded procedure of this API.
p_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number of the cust account relate record</li> <li>• Return new value after update</li> </ul>

## Other Validations

The combination of cust\_account\_id and related\_cust\_account\_id should be unique for all active customer accounts in an organization.

# 11

---

## Customer Account Site API Use

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,
    org_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 String tp_header_id;  
    public String ece_tp_location_code;  
    public String primary_specialist_id;  
    public String secondary_specialist_id;  
    public String territory_id;  
    public String territory;  
    public String translated_customer_name;  
    public String created_by_module;  
    public String 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 String 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 String 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 String           primary_salesrep_id;
        public String           finchrg_receivables_trx_id;
        public BigDecimal       dates_negative_tolerance;
        public BigDecimal       dates_positive_tolerance;
        public String           date_type_preference;
        public String           over_shipment_tolerance;
        public String           under_shipment_tolerance;
        public String           item_cross_ref_pref;
        public String           over_return_tolerance;
        public String           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 String           gl_id_rec;
        public String           gl_id_rev;
        public String           gl_id_tax;
        public String           gl_id_freight;
        public String           gl_id_clearing;
        public String           gl_id_unbilled;
        public String           gl_id_unearned;
        public String           gl_id_unpaid_rec;
        public String           gl_id_remittance;
        public String           gl_id_factor;
        public String           tax_classification;
        public String           created_by_module;
        public String           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
    CustAcctSiteRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) 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	<p>Validation:</p> <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• cust_account_id is foreign key of hz_cust_accounts</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_site_id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• party_site_id is foreign key of hz_party_sites<sup>2</sup></li> <li>• party_site_id must link to a location of content source type USER_ENTERED</li> </ul>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
global_attribute_cate gory	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_referen ce	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_c ode	IN	VARCHAR2	No	Validation: customer_category_code is lookup code in lookup type ADDRESS_CATEGORY
language	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use hz_locations.language instead. 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_cod e	IN	VARCHAR2	No	Validation: The ece_tp_location_code should be unique for a customer within the organization.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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
    CustAcctSiteRec
    BigDecimal [ ]
    String [ ]
    BigDecimal [ ]
    String [ ]
) throws SQLException;
    p_init_msg_list,
    p_cust_acct_site_rec,
    p_object_version_number,
    x_return_status,
    x_msg_count,
    x_msg_data
```

### **Parameter Description and Validation**

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute16	IN	VARCHAR2	No	
attribute17	IN	VARCHAR2	No	
attribute18	IN	VARCHAR2	No	
attribute19	IN	VARCHAR2	No	
attribute20	IN	VARCHAR2	No	
global_attribute_cate gory	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_referen ce	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: <ul style="list-style-type: none"> <li>• Status cannot be set to null during update.</li> <li>• Status is lookup code in lookup type CODE_STATUS.</li> </ul>
customer_category_ code	IN	VARCHAR2	No	Validation: customer_category_code is lookup code in lookup type ADDRESS_CATEGORY
language	IN	VARCHAR2	No	Comment: This parameter is no longer used. Use hz_locations.language instead. Validation: language is foreign key of fnd installed languages.
key_account_flag	IN	VARCHAR2	No	
tp_header_id	IN	NUMBER	No	Validation: tp_header_id must be unique
ece_tp_location_cod e	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass current object_version_number of the record from hz_cust_acct_sites</li> <li>• Return new value after update</li> </ul>

## Other Validations

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

## Create Customer Account Site Use API

### Description

This routine is used to create a Customer Account Site Use. The API creates a record in the HZ\_CUST\_SITEUSES 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.
    CUSTOMER_PROFILE_REC_TYPE,
    p_create_profile     IN      VARCHAR2 := FND_API.G_TRUE,
    p_create_profile_amt IN      VARCHAR2 := FND_API.G_TRUE,
    x_site_use_id        OUT     NUMBER,
    x_return_status       OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2
)
```

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

## Java Method

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

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

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
site_use_id	IN	NUMBER	Yes	Validation: unique if passed in, else generated from sequence
cust_acct_site_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• cust_account_id is foreign key to hz_cust_acct_sites</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
site_use_code	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• site_use_code is lookup type YES/NO</li> </ul>
primary_flag	IN	VARCHAR2	No	<p>Validation: Primary_flag is lookup code in lookup type YES/NO</p> <p>Default: N</p>
status	IN	VARCHAR2	No	<p>Validation: status is lookup code in lookup type CODE_STATUS</p> <p>Default: A</p>
location	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• It will be generated from sequence if user does not pass in and AUTO_SITE_NUMBERING in AR_SYSTEM_PARAMETERS is on</li> <li>• location must be unique within a customer account/ site_use_type</li> </ul>
bill_to_site_use_id	IN	NUMBER	No	
orig_system_reference	IN	VARCHAR2	No	Validation: site_use_id
orig_system	IN	VARCHAR2	No	<p>Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system .</p> <p>Default: UNKNOWN if an orig_system_reference is passed in.</p>
sic_code	IN	VARCHAR2	No	
payment_term_id	IN	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gsa_indicator	IN	VARCHAR2	No	Validation: gsa_indicator is lookup code in lookup type YES/NO  Default: N
ship_partial	IN	VARCHAR2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR2	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be used in both single and multi org case.
fob_point	IN	VARCHAR2	No	Validation: Validated against AR lookup type FOB.
order_type_id	IN	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	IN	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_categ	IN	VARCHAR2	No	
primary_salesrep_id	IN	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_	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.

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gl_id_rec	IN	NUMBER	No	Validation: gl_id_rec is valid gl field
gl_id_rev	IN	NUMBER	No	Validation: gl_id_rev is valid gl field
gl_id_tax	IN	NUMBER	No	Validation: gl_id_tax is valid gl field
gl_id_freight	IN	NUMBER	No	Validation: gl_id_freight is valid gl field
gl_id_clearing	IN	NUMBER	No	Validation: gl_id_clearing is valid gl field
gl_id_unbilled	IN	NUMBER	No	Validation: gl_id_unbilled is valid gl field
gl_id_unearned	IN	NUMBER	No	Validation: gl_id_unearned is valid gl field
gl_id_unpaid_rec	IN	NUMBER	No	Validation: gl_id_unpaid_rec is valid gl field
gl_id_remittance	IN	NUMBER	No	Validation: gl_id_remittance is valid gl field
gl_id_factor	IN	NUMBER	No	Validation: gl_id_factor is valid gl field
tax_classification	IN	VARCHAR2	No	
created_by_module	IN	VARCHAR2	No	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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\_SITEUSES 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	Yes	Validation: Not updateable
site_use_code	IN	VARCHAR 2	No	Validation: Not updateable
primary_flag	IN	VARCHAR 2	No	Validation: <ul style="list-style-type: none"><li>• primary_flag cannot be set to null during update</li><li>• primary_flag is lookup code in lookup type YES/NO</li></ul>
status	IN	VARCHAR 2	No	Validation: <ul style="list-style-type: none"><li>• status cannot be set to null during update</li><li>• status is lookup code in lookup type CODE_STATUS</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
location	IN	VARCHAR 2	No	Validation: Can only be updated if the AUTO_SITE_NUMBERING profile option in AR_SYSTEMS_PARAMETERS is unchecked.
contact_id	IN	NUMBER	No	
bill_to_site_use_id	IN	NUMBER	No	
orig_system_reference	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system.
sic_code	IN	VARCHAR 2	No	
payment_term_id	IN	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS.
gsa_indicator	IN	VARCHAR 2	No	Validation: gsa_indicator is lookup code in lookup type YES/NO
ship_partial	IN	VARCHAR 2	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Validated against AR lookup type FOB.
order_type_id	IN	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
price_list_id	IN	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.
freight_term	IN	VARCHAR 2	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	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute10	IN	VARCHAR 2	No	
tax_reference	IN	VARCHAR 2	No	
sort_priority	IN	NUMBER	No	
tax_code	IN	VARCHAR 2	No	Validation: Must be a valid tax_code from AR_VAT_TAX.
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
attribute25	IN	VARCHAR 2	No	
demand_class_code	IN	VARCHAR 2	No	Validation: Validated against AR lookup type DEMAND_CLASS.
tax_header_level_flag	IN	VARCHAR 2	No	
tax_rounding_rule	IN	VARCHAR 2	No	
global_attribute1	IN	VARCHAR 2	No	
global_attribute2	IN	VARCHAR 2	No	
global_attribute3	IN	VARCHAR 2	No	
global_attribute4	IN	VARCHAR 2	No	
global_attribute5	IN	VARCHAR 2	No	
global_attribute6	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute7	IN	VARCHAR 2	No	
global_attribute8	IN	VARCHAR 2	No	
global_attribute9	IN	VARCHAR 2	No	
global_attribute10	IN	VARCHAR 2	No	
global_attribute11	IN	VARCHAR 2	No	
global_attribute12	IN	VARCHAR 2	No	
global_attribute13	IN	VARCHAR 2	No	
global_attribute14	IN	VARCHAR 2	No	
global_attribute15	IN	VARCHAR 2	No	
global_attribute16	IN	VARCHAR 2	No	
global_attribute17	IN	VARCHAR 2	No	
global_attribute18	IN	VARCHAR 2	No	
global_attribute19	IN	VARCHAR 2	No	
global_attribute20	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute_categ	IN	VARCHAR 2	No	
primary_salesrep_id	IN	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_tx	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	VARCHAR 2	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	VARCHAR 2	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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_sets_include_lines_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	IN	VARCHAR 2	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO
invoice_quantity_rule	IN	VARCHAR 2	No	
pricing_event	IN	VARCHAR 2	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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN/OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record.</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass current object_version_number of the record from hz_cust_site_uses</li> <li>• Return new value after update</li> </ul>

## Other Validations

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

# 12

---

## Customer Profile and Customer Account Role API Use

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 String collector_id;
    public String 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 String profile_class_id;
    public String site_use_id;
    public String credit_rating;
    public String risk_code;
    public String standard_terms;
    public String override_terms;
    public String dunning_letter_set_id;
    public String interest_period_days;
    public String payment_grace_days;
    public String discount_grace_days;
    public String statement_cycle_id;
    public String account_status;
    public String percent_collectable;
    public String autocash_hierarchy_id;
    public String attribute_category;
    public String attribute1;
    public String attribute2;
    public String attribute3;
    public String attribute4;
    public String attribute5;
    public String attribute6;
    public String attribute7;
    public String attribute8;
    public String attribute9;
    public String attribute10;
    public String attribute11;
    public String attribute12;
    public String attribute13;
    public String attribute14;
    public String attribute15;
    public String auto_rec_incl_disputed_flag;
    public String tax_printing_option;
    public String charge_on_finance_charge_flag;
    public String grouping_rule_id;
    public String 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 String autocash_hierarchy_id_for_adr;
        public String lockbox_matching_option;
        public String created_by_module;
        public String 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 String cust_account_id;
    public String site_use_id;
    public 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
cust_account_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Foreign key to hz_cust_accounts.cust_account_id</li> <li>• One of the following 2 columns need to be not null : cust_account_id and party_id</li> <li>• If cust_account_id is not null and party_id is null then:           <ul style="list-style-type: none"> <li>• The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.</li> <li>• The party_id in the hz_parties table associated with this cust_account_id must be active.</li> </ul> </li> <li>• If cust_account_id is null and party_id is not null then:           <ul style="list-style-type: none"> <li>• The party_id is a foreign key to an active hz_parties.party_id.</li> <li>• The cust_account_id will be defaulted to (-1).</li> </ul> </li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
				<ul style="list-style-type: none"> <li>• If cust_account_id is not null and party_id is not null then:           <ul style="list-style-type: none"> <li>• The party_id is a foreign key to an active hz_parties.party_id.</li> <li>• The party_id and the cust_account_id must be associated in hz_cust_accounts.</li> </ul> </li> <li>• You can only have one customer profile for one account In this case you can find the party_id in the hz_customer_profiles.party_id column.</li> <li>• You can only have one customer profile at party level for a party. In this case the cust_account_id will be defaulted to -1.</li> </ul>
status	IN	VARCHAR 2	No	<p>Validation: It is a lookup code in lookup type CODE_STATUS</p> <p>Default: A</p>
collector_id	IN	NUMBER	No	<p>Validation: foreign key to ar_collectors</p> <p>Default: it is defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
credit_analyst_id	IN	NUMBER	No	
credit_checking	IN	VARCHAR 2	No	<p>Validation:It is lookup code in lookup type YES/NO</p> <p>Default: It is defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
next_credit_review_date	IN	DATE	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
tolerance	IN	NUMBER	No	<p>Validation: Must be between -100 and 100.</p> <p>Default: Defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
discount_terms	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If discount_terms = Y, then discount_grace_days should be greater than or equal to 0.</li> <li>• If discount_terms = N, then discount_grace_days should be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
dunning_letters	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Dunning letters is lookup code in lookup type YES/NO.</li> <li>• If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
interest_charges	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in lookup type YES/NO.</li> <li>• If interest_charges = Y, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>• If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
send_statements	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in lookup type YES/NO.</li> <li>• If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have value.</li> <li>• If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
credit_balance_state ments	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in lookup type YES/NO.</li> <li>• Can be populated if send_statements is Y.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
credit_hold	IN	VARCHAR 2	No	<p>Validation: It is a lookup code in lookup type YES/NO</p> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
profile_class_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If passed in, profile_class_id should be positive.</li> <li>• Foreign key to hz_cust_profile_classes and the corresponding profile class should be active</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID=0)</p>
site_use_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Foreign key to hz_cust_site_uses</li> <li>• One site use can only have one profile.</li> <li>• The customer which the site_use_id belongs to should have the same id as cust_account_id in this profile. In this case cust_account_id is mandatory.</li> </ul>
credit_rating	IN	VARCHAR 2	No	Validation: Credit Rating is lookup code in AR lookup type CREDIT_RATING
risk_code	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Validated against AR lookup type YES/NO.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
dunning_letter_set_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>If dunning_letters = Y, the dunning_letter_set_id must have a value.</li> <li>If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.</li> </ul>
interest_period_days	IN	NUMBER	No	Validation: Can be entered when interest_charges is Y and it is mandatory.
payment_grace_days	IN	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	IN	NUMBER	No	Validation: Can be populated only if discount_terms is Y.
statement_cycle_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES.</li> <li>Can be populated if send_statements is Y.</li> </ul>
account_status	IN	VARCHAR 2	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	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
auto_rec_incl_disputed_flag	IN	VARCHAR 2	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	VARCHAR 2	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	IN	VARCHAR 2	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	VARCHAR 2	No	
jgzz_attribute1	IN	VARCHAR 2	No	
jgzz_attribute2	IN	VARCHAR 2	No	
jgzz_attribute3	IN	VARCHAR 2	No	
jgzz_attribute4	IN	VARCHAR 2	No	
jgzz_attribute5	IN	VARCHAR 2	No	
jgzz_attribute6	IN	VARCHAR 2	No	
jgzz_attribute7	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
jgzz_attribute8	IN	VARCHAR 2	No	
jgzz_attribute9	IN	VARCHAR 2	No	
jgzz_attribute10	IN	VARCHAR 2	No	
jgzz_attribute11	IN	VARCHAR 2	No	
jgzz_attribute12	IN	VARCHAR 2	No	
jgzz_attribute13	IN	VARCHAR 2	No	
jgzz_attribute14	IN	VARCHAR 2	No	
jgzz_attribute15	IN	VARCHAR 2	No	
global_attribute1	IN	VARCHAR 2	No	
global_attribute2	IN	VARCHAR 2	No	
global_attribute3	IN	VARCHAR 2	No	
global_attribute4	IN	VARCHAR 2	No	
global_attribute5	IN	VARCHAR 2	No	
global_attribute6	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute7	IN	VARCHAR 2	No	
global_attribute8	IN	VARCHAR 2	No	
global_attribute9	IN	VARCHAR 2	No	
global_attribute10	IN	VARCHAR 2	No	
global_attribute11	IN	VARCHAR 2	No	
global_attribute12	IN	VARCHAR 2	No	
global_attribute13	IN	VARCHAR 2	No	
global_attribute14	IN	VARCHAR 2	No	
global_attribute15	IN	VARCHAR 2	No	
global_attribute16	IN	VARCHAR 2	No	
global_attribute17	IN	VARCHAR 2	No	
global_attribute18	IN	VARCHAR 2	No	
global_attribute19	IN	VARCHAR 2	No	
global_attribute20	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute_category	IN	VARCHAR 2	No	
cons_inv_flag	IN	VARCHAR 2	No	Validation: cons_inv_flag is lookup code in lookup type YES/NO
cons_inv_type	IN	VARCHAR 2	No	
autocash_hierarchy_id_for_addr	IN	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
lockbox_matching_option	IN	VARCHAR 2	No	Validation: Validated against AR lookup type ARLPLB_MATCHING_OPTION.
created_by_module	IN	VARCHAR 2	Y	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated
review_cycle	IN	VARCHAR 2	No	Validation : Validated against AR lookup type PERIODIC_REVIEW_CYCLE.
last_review_date	IN	DATE	No	

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• One of the following columns cannot be null: cust_account_id and party_id.</li> <li>• If cust_account_id is not null and party_id is null, then this is mandatory field.           <ul style="list-style-type: none"> <li>• The cust_account_id is a foreign key to hz_cust_accounts.cust_account_id.</li> <li>• The party_id in the hz_parties table associated with this cust_account_id must be active.</li> </ul> </li> <li>• If cust_account_id is null and party_id is not null then           <ul style="list-style-type: none"> <li>• The party_id is a foreign key to an active hz_parties.party_id.</li> </ul> </li> </ul> <p>The cust_account_id will be defaulted to (-1).</p> <ul style="list-style-type: none"> <li>• If cust_account_id is not null and party_id is not null:           <ul style="list-style-type: none"> <li>• The party_id is a foreign key to an active hz_parties.party_id.</li> <li>• Party_id and cust_account_id must be associated in hz_cust_accounts.</li> </ul> </li> <li>• You can only have one customer profile for one account. You can find the party_id in the hz_customer_profiles.party_id column.</li> <li>• You can only have one customer profile at party level for a party. The cust_account_id defaults to (-1).</li> </ul>

---

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_create_profile_amt	IN	VARCHAR 2	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_prof ile_id	OUT	NUMBER	No	Comment: Returns cust_account_profile_id for the record created
credit_classification	IN	VARCHAR 2	No	Validation: This is a lookup code in lookup_type 'CREDIT_CLASSIFICATION'.  Default: NULL
AUTOMATCH_SET _ID	IN	NUMBER	No	

## 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_PROFILE_REC_TYPE,
    p_object_version_number              IN OUT NUMBER,
    x_return_status                      OUT     VARCHAR2,
    x_msg_count                          OUT     NUMBER,
    x_msg_data                           OUT     VARCHAR2
)
```

## Java Method

```
public static void updateCustomerProfile(
    OracleConnection_connection,
    String p_init_msg_list,
    CustomerProfileRec p_customer_profile_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_profile_id	IN	NUMBER	Yes	Validation: valid cust_account_profile_id has to be passed in  Comment: Pass cust_account_profile_id from table hz_customer_profiles
cust_account_id	IN	NUMBER	No	Validation: Not updateable
status	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type CODE_STATUS  Default: A
collector_id	IN	NUMBER	No	Validation: Foreign key to ar_collectors
credit_analyst_id	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Tolerance cannot be updated to null.</li> <li>• Must be between -100 and 100.</li> </ul>
discount_terms	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If discount_terms = Y, then discount_grace_days should be greater than or equal to 0.</li> <li>• If discount_terms = N, then discount_grace_days should be null.</li> </ul>
dunning_letters	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> </ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
interest_charges	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup type YES/NO.</li> <li>• If interest_charges = Y, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>• If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul>
send_statements	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup type YES/NO.</li> <li>• If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have a value.</li> <li>• If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.</li> </ul>
credit_balance_state ments	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in lookup type YES/NO.</li> <li>• Can be populated if send_statements is Y.</li> </ul>
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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If dunning_letters = Y, then dunning_letters_set_id must have a value.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>• Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.</li> </ul>
interest_period_days	IN	NUMBER	No	Validation: Must be entered when interest_charges is Y.
payment_grace_days	IN	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	IN	NUMBER	No	Validation: Can only be populated if the value of discount_terms is Y.
statement_cycle_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES.</li> <li>• Can be populated if send_statements is Y.</li> </ul>
account_status	IN	VARCHAR2	No	Validation: Validated against AR lookup type ACCOUNT_STATUS.
percent_collectable	IN	NUMBER	No	Validation: Must be between 0 and 100.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	
attribute15	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
auto_rec_incl_disputed_flag	IN	VARCHAR2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot be set to null</li> <li>• It is a lookup code in lookup type YES/NO</li> </ul>
tax_printing_option	IN	VARCHAR2	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	IN	VARCHAR2	No	Validation: It is lookup code in lookup type YES/NO.
grouping_rule_id	IN	NUMBER	No	Validation: Must be a valid grouping_rule_id from RA_GROUPING_RULES.
clearing_days	IN	NUMBER	No	Validation: Must be greater than zero.
jgzz_attribute_category	IN	VARCHAR2	No	
jgzz_attribute1	IN	VARCHAR2	No	
jgzz_attribute2	IN	VARCHAR2	No	
jgzz_attribute3	IN	VARCHAR2	No	
jgzz_attribute4	IN	VARCHAR2	No	
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute17	IN	VARCHAR2	No	
global_attribute18	IN	VARCHAR2	No	
global_attribute19	IN	VARCHAR2	No	
global_attribute20	IN	VARCHAR2	No	
global_attribute_cate_gory	IN	VARCHAR2	No	
cons_inv_flag	IN	VARCHAR2	No	Validation: It is a lookup code in lookup type YES/NO
cons_inv_type	IN	VARCHAR2	No	
autocash_hierarchy_id_for_addr	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: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Not updateable if value exists
p_object_version_number	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass the current object_version_number of the record</li> <li>• Return new value after update</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
credit_classification	IN	VARCHAR2	No	Validation: Validated against AR lookup type 'CREDIT_CLASSIFICATION'
AUTOMATCH_SET_ID	IN	NUMBER	No	

## Create Customer Profile Amount API

### Description

This routine is used to create Customer Profile Amount. The API creates record in the HZ\_CUST\_PROFILE\_AMTS table for a profile. You have to create a customer profile before you can create profile amount record.

### PL/SQL Procedure

```
PROCEDURE create_cust_profile_amt (
    p_init_msg_list                      IN      VARCHAR2 := FND_API.G_FALSE,
    p_check_foreign_key                   IN      VARCHAR2 := FND_API.G_TRUE,
    p_cust_profile_amt_rec              IN      CUST_PROFILE_AMT_REC_TYPE,
    x_cust_acct_profile_amt_id          OUT     NUMBER,
    x_return_status                      OUT     VARCHAR2,
    x_msg_count                          OUT     NUMBER,
    x_msg_data                           OUT     VARCHAR2
)

```

**Note:** p\_check\_foreign\_key indicates whether to do foreign key checking for the profile amount being created. If value equals to FND\_API.G\_TRUE, API will do foreign key checking on cust\_account\_id, cust\_account\_profile\_id and site\_use\_id. This was added for backward compatibility with customer form only. You should always set the parameter to FND\_API.G\_TRUE when you call API.

## Java Method

```
public static void createCustProfileAmt(
    OracleConnection_connection,
    String                                p_init_msg_list,
    String                                p_check_foreign_key,
    CustProfileAmtRec                    p_cust_profile_amt_rec,
    BigDecimal [ ]                      x_cust_acct_profile_amt_id,
    String [ ]                           x_return_status,
    BigDecimal [ ]                      x_msg_count,
    String [ ]                           x_msg_data
) throws SQLException;
```

**Note:** p\_check\_foreign\_key indicates whether to do foreign key checking for the profile amount being created. If value equals to HzConstant.getGTrue(), we will do foreign key checking on cust\_account\_id, cust\_account\_profile\_id and site\_use\_id. This was added for backward compatibility with customer form only. You should always set the parameter to HzConstant.getGTrue() when you call API.

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_profile_am t_id	IN	NUMBER	Yes/No	Validation: unique if passed in, else generated from sequence
cust_account_profile _id	IN	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"><li>Mandatory attribute</li><li>When p_check_foreign_key is FND_API_G_TRUE, cust_account_profile_id must be a foreign key to hz_customer_profiles</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
currency_code	IN	VARCHAR 2	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Foreign key to fnd_currencies</li> <li>• For a given cust_account_profile_id and currency_code, only one record of the profile amount is allowed.</li> </ul>
trx_credit_limit	IN	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
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_invoiced_amount	IN	NUMBER	No	
max_interest_charge	IN	NUMBER	No	
min_statement_amount	IN	NUMBER	No	
auto_rec_min_receipt_amount	IN	NUMBER	No	
interest_rate	IN	NUMBER	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
min_fc_balance_amo unt	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
min_fc_invoice_amo unt	IN	NUMBER	No	
cust_account_id	IN	NUMBER	Yes	<p>Validation:</p> <p>Mandatory attribute</p> <p>When p_check_foreign_key is FND_API. G_TRUE, cust_account_id must be a foreign key to hz_cust_accounts.</p> <p>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.</p>
site_use_id	IN	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• When p_check_foreign_key is FND_API. G_TRUE, site_use_id must be a foreign key to hz_cust_site_uses</li> <li>• When p_check_foreign_key is FND_API. G_TRUE, site_use_id should be the same as cust_account_id in corresponding customer profile record.</li> </ul>
expiration_date	IN	DATE	No	
jgzz_attribute_categ ory	IN	VARCHAR 2	No	
jgzz_attribute1	IN	VARCHAR 2	No	
jgzz_attribute2	IN	VARCHAR 2	No	
jgzz_attribute3	IN	VARCHAR 2	No	
jgzz_attribute4	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
jgzz_attribute5	IN	VARCHAR 2	No	
jgzz_attribute6	IN	VARCHAR 2	No	
jgzz_attribute7	IN	VARCHAR 2	No	
jgzz_attribute8	IN	VARCHAR 2	No	
jgzz_attribute9	IN	VARCHAR 2	No	
jgzz_attribute10	IN	VARCHAR 2	No	
jgzz_attribute11	IN	VARCHAR 2	No	
jgzz_attribute12	IN	VARCHAR 2	No	
jgzz_attribute13	IN	VARCHAR 2	No	
jgzz_attribute14	IN	VARCHAR 2	No	
jgzz_attribute15	IN	VARCHAR 2	No	
global_attribute1	IN	VARCHAR 2	No	
global_attribute2	IN	VARCHAR 2	No	
global_attribute3	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute4	IN	VARCHAR 2	No	
global_attribute5	IN	VARCHAR 2	No	
global_attribute6	IN	VARCHAR 2	No	
global_attribute7	IN	VARCHAR 2	No	
global_attribute8	IN	VARCHAR 2	No	
global_attribute9	IN	VARCHAR 2	No	
global_attribute10	IN	VARCHAR 2	No	
global_attribute11	IN	VARCHAR 2	No	
global_attribute12	IN	VARCHAR 2	No	
global_attribute13	IN	VARCHAR 2	No	
global_attribute14	IN	VARCHAR 2	No	
global_attribute15	IN	VARCHAR 2,	No	
global_attribute16	IN	VARCHAR 2	No	
global_attribute17	IN	VARCHAR 2	No	

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
global_attribute18	IN	VARCHAR 2	No	
global_attribute19	IN	VARCHAR 2	No	
global_attribute20	IN	VARCHAR 2	No	
global_attribute_categ	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated.
p_check_foreign_ke	IN	VARCHAR 2	No	Comment: Indicates whether foreign key checks will be done wherever possible.
x_cust_acct_profile_	OUT	NUMBER	No	Comment: Returns cust_acct_profile_amt_id for the record created.
amt_id				

## 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_REC_TYPE,
    p_object_version_number   IN OUT NUMBER,
    x_return_status            OUT     VARCHAR2,
    x_msg_count                OUT     NUMBER,
    x_msg_data                 OUT     VARCHAR2
)
```

## Java Method

```
public static void updateCustProfileAmt(
    OracleConnection_connection,
    String p_init_msg_list,
    CustProfileAmtRec p_cust_profile_amt_rec,
    BigDecimal [ ] p_object_version_number,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_acct_profile_am t_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	VARCHAR 2	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_amou nt	IN	NUMBER	No	
min_dunning_invoic e_amount	IN	NUMBER	No	
max_interest_charge	IN	NUMBER	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
min_fc_balance_amo unt	IN	NUMBER	No	
min_fc_invoice_amo unt	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_categ ory	IN	VARCHAR 2	No	
jgzz_attribute1	IN	VARCHAR 2	No	
jgzz_attribute2	IN	VARCHAR 2	No	
jgzz_attribute3	IN	VARCHAR 2	No	
jgzz_attribute4	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
jgzz_attribute5	IN	VARCHAR 2	No	
jgzz_attribute6	IN	VARCHAR 2	No	
jgzz_attribute7	IN	VARCHAR 2	No	
jgzz_attribute8	IN	VARCHAR 2	No	
jgzz_attribute9	IN	VARCHAR 2	No	
jgzz_attribute10	IN	VARCHAR 2	No	
jgzz_attribute11	IN	VARCHAR 2	No	
jgzz_attribute12	IN	VARCHAR 2	No	
jgzz_attribute13	IN	VARCHAR 2	No	
jgzz_attribute14	IN	VARCHAR 2	No	
jgzz_attribute15	IN	VARCHAR 2	No	
global_attribute1	IN	VARCHAR 2	No	
global_attribute2	IN	VARCHAR 2	No	
global_attribute3	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute4	IN	VARCHAR 2	No	
global_attribute5	IN	VARCHAR 2	No	
global_attribute6	IN	VARCHAR 2	No	
global_attribute7	IN	VARCHAR 2	No	
global_attribute8	IN	VARCHAR 2	No	
global_attribute9	IN	VARCHAR 2	No	
global_attribute10	IN	VARCHAR 2	No	
global_attribute11	IN	VARCHAR 2	No	
global_attribute12	IN	VARCHAR 2	No	
global_attribute13	IN	VARCHAR 2	No	
global_attribute14	IN	VARCHAR 2	No	
global_attribute15	IN	VARCHAR 2	No	
global_attribute16	IN	VARCHAR 2	No	
global_attribute17	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute18	IN	VARCHAR 2	No	
global_attribute19	IN	VARCHAR 2	No	
global_attribute20	IN	VARCHAR 2	No	
global_attribute_cate gory	IN	VARCHAR 2	No	
created_by_module	IN	VARCHAR 2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
p_object_version_nu mber	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass current object_version_number of the record from hz_cust_profile_amts table</li> <li>• Return new value after update</li> </ul>

## Customer Account Role APIs

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

**Java Class Name:** HzCustAccountRoleV2Pub

## PL/SQL Record Structure for Customer Account Role

```
TYPE cust_account_role_rec_type IS RECORD (
    cust_account_role_id NUMBER,
    party_id NUMBER,
    cust_account_id NUMBER,
    cust_acct_site_id NUMBER,
    primary_flag VARCHAR2(1),
    role_type VARCHAR2(30),
    source_code VARCHAR2(150),
    attribute_category VARCHAR2(30),
    attribute1 VARCHAR2(150),
    attribute2 VARCHAR2(150),
    attribute3 VARCHAR2(150),
    attribute4 VARCHAR2(150),
    attribute5 VARCHAR2(150),
    attribute6 VARCHAR2(150),
    attribute7 VARCHAR2(150),
    attribute8 VARCHAR2(150),
    attribute9 VARCHAR2(150),
    attribute10 VARCHAR2(150),
    attribute11 VARCHAR2(150),
    attribute12 VARCHAR2(150),
    attribute13 VARCHAR2(150),
    attribute14 VARCHAR2(150),
    attribute15 VARCHAR2(150),
    attribute16 VARCHAR2(150),
    attribute17 VARCHAR2(150),
    attribute18 VARCHAR2(150),
    attribute19 VARCHAR2(150),
    attribute20 VARCHAR2(150),
    attribute21 VARCHAR2(150),
    attribute22 VARCHAR2(150),
    attribute23 VARCHAR2(150),
    attribute24 VARCHAR2(150),
    orig_system_reference VARCHAR2(240),
    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_REC_TYPE,  
    x_cust_account_role_id   OUT     NUMBER,  
    x_return_status          OUT     VARCHAR2,  
    x_msg_count              OUT     NUMBER,  
    x_msg_data               OUT     VARCHAR2  
)
```

## Java Method

```
public static void createCustAccountRole(
    OracleConnection_connection,
    String p_init_msg_list,
    CustAccountRoleRec p_cust_account_role_rec,
    BigDecimal [ ] x_cust_account_role_id,
    String [ ] x_return_status,
    BigDecimal [ ] x_msg_count,
    String [ ] x_msg_data
) throws SQLException;
```

## Parameter Description and Validation

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
cust_account_role_id	IN	NUMBER	Yes	Validation: Unique if passed in, else generated from sequence
party_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute</li><li>• Foreign key to hz_parties</li></ul>
cust_account_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory field</li><li>• Foreign key to hz_cust_accounts</li></ul>
cust_acct_site_id	IN	NUMBER	No	Validation: <ul style="list-style-type: none"><li>• Foreign key to hz_cust_acct_sites</li><li>• The cust_account_id in hz_cust_acct_sites which cust_acct_site_id points to should be same as the cust_account_id put in the hz_cust_account_roles</li></ul>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
primary_flag	IN	VARCHAR 2	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Primary flag is lookup code in lookup type YES/NO</li> <li>• It is unique per cust_account_id or cust_acct_site_id</li> </ul> <p>Default: N</p>
role_type	IN	VARCHAR 2	Y	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• It is a lookup code in lookup type ACCT_ROLE_TYPE<sup>2</sup></li> <li>• The combination of CUST_ACCOUNT_ID, PARTY_ID, ROLE_TYPE should be unique.I.</li> <li>• Or the combination of CUST_ACCT_SITE_ID, PARTY_ID, ROLE_TYPE should be unique</li> </ul>
source_code	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
orig_system_referen ce	IN	VARCHAR 2	Yes	<p>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.</p> <p>Default: cust_account_role_id</p>
orig_system	IN	VARCHAR 2	Yes	<p>Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system.</p> <p>Default: UNKNOWN if an orig_system_reference is passed in.</p>
attribute25	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	Yes	<p>Validation: It is a lookup code in AR lookup type REGISTRY_STATUS Default: 'A'</p>
created_by_module	IN	VARCHAR 2	Yes	<p>Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.</p>
application_id	IN	NUMBER	No	<p>Comment: Text to indicate application from which creation of record is initiated</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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_REC_TYPE,
    p_object_version_number              IN OUT      NUMBER,
    x_return_status                      OUT         VARCHAR2,
    x_msg_count                          OUT         NUMBER,
    x_msg_data                           OUT         VARCHAR2
)
```

### Java Method

```
public static void updateCustAccountRole(
    OracleConnection_connection,
    String                                     p_init_msg_list,
    CustAccountRoleRec                    p_cust_account_role_rec,
    BigDecimal [ ]                         p_object_version_number,
    String [ ]                             x_return_status,
    BigDecimal [ ]                         x_msg_count,
    String [ ]                            x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Update Customer Account Role API. The table includes the parameter names, the type of each parameter,

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	VARCHAR 2	No	Validation: Primary flag is lookup code in lookup type YES/NO
role_type	IN	VARCHAR 2	Yes	Validation: Not updateable
source_code	IN	VARCHAR 2	No	
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute20	IN	VARCHAR 2	No	
attribute21	IN	VARCHAR 2	No	
attribute22	IN	VARCHAR 2	No	
attribute23	IN	VARCHAR 2	No	
attribute24	IN	VARCHAR 2	No	
orig_system_referen ce	IN	VARCHAR 2	Yes	Validation: Not updateable
orig_system	IN	VARCHAR 2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
attribute25	IN	VARCHAR 2	No	
status	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"> <li>• Status cannot be set to null during update</li> <li>• It is a lookup code in lookup type REGISTRY_STATUS</li> </ul>
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists

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

## Create Role Responsibility API

### Description

This routine is used to create a Role Responsibility. The API creates a record in the HZ\_ROLE\_RESPONSIBILITY table.

### PL/SQL Procedure

```
PROCEDURE create_role_responsibility (
    p_init_msg_list                      IN          VARCHAR2 := FND_API.
    G_FALSE,
    p_role_responsibility_rec            IN          ROLE_RESPONSIBILITY_REC_TYPE,
    x_responsibility_id                  OUT         NUMBER,
    x_return_status                      OUT         VARCHAR2,
    x_msg_count                          OUT         NUMBER,
    x_msg_data                           OUT         VARCHAR2
)
```

### Java Method

```
public static void createRoleResponsibility(
    OracleConnection connection,
    String p_init_msg_list,
    RoleResponsibilityRec p_role_responsibility_rec,
    BigDecimal x_responsibility_id,
    String x_return_status,
    BigDecimal x_msg_count,
    String x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Role

Responsibility API. The table includes the parameter names, the type of each parameter, the data type of each parameter, the necessity of the parameter, and other information about the parameter such as validation, defaults, and other comments.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
responsibility_id	IN	NUMBER	Yes/No	Validation: unique if passed in, else generated from sequence
cust_account_role_id	IN	NUMBER	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory field</li> <li>• Foreign key to hz_cust_account_roles</li> </ul>
responsibility_type	IN	VARCHAR 2	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory field</li> <li>• It is a lookup code in lookup type SITE_USE_CODE</li> </ul>
primary_flag	IN	VARCHAR 2	No	Validation: <ul style="list-style-type: none"> <li>• Primary Flag is lookup code in lookup type YES/NO.</li> <li>• It is unique per cust_account_role_id Default: N</li> </ul>
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
orig_system_referen ce	IN	VARCHAR 2	No	Default: responsibility_id
created_by_module	IN	VARCHAR 2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment: Text to indicate application from which creation of record is initiated

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_responsibility_id	OUT	NUMBER	No	Comment: Returns responsibility_id for the record created

## Other Validations

The combination of cust\_account\_role\_id and responsibility\_type should be unique.

## Update Role Responsibility API

### Description

This routine is used to update a Role Responsibility. The API updates a record in the HZ\_ROLE\_RESPONSIBILITY table.

### PL/SQL Procedure

```
PROCEDURE update_role_responsibility (
    p_init_msg_list                      IN      VARCHAR2 := FND_API .
    G_FALSE,
    p_role_responsibility_rec            IN      ROLE_RESPONSIBILITY_REC_TYPE,
    p_object_version_number              IN OUT   NUMBER ,
    x_return_status                      OUT     VARCHAR2 ,
    x_msg_count                          OUT     NUMBER ,
    x_msg_data                           OUT     VARCHAR2
)
```

### Java Method

```
public static void updateRoleResponsibility(
    OracleConnection_connection,
    String                                     p_init_msg_list,
    RoleResponsibilityRec                    p_role_responsibility_rec,
    BigDecimal [ ]                            p_object_version_number,
    String [ ]                                x_return_status,
    BigDecimal [ ]                            x_msg_count,
    String [ ]                                x_msg_data
) throws SQLException;
```

### Parameter Description and Validation

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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
responsibility_id	IN	NUMBER	Yes	Validation: Valid responsibility_id should be passed in  Comment: Pass responsibility_id from the hz_role_responsibility table
cust_account_role_id	IN	NUMBER	No	Validation: Non updateable
responsibility_type	IN	VARCHAR2	No	Validation: Non updateable
primary_flag	IN	VARCHAR2	No	Validation: <ul style="list-style-type: none"><li>• Primary Flag is lookup code in lookup type YES/NO</li><li>• It is unique per cust_account_role_id</li></ul>
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
orig_system_referen ce	IN	VARCHAR2	No	Validation: Not updateable
created_by_module	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Validation: Cannot be updated if value exists
p_object_version_nu mber	IN OUT	NUMBER	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• Validated against value in the database for the existing record</li> </ul> <p>Comment:</p> <ul style="list-style-type: none"> <li>• Pass current object_version_number of the record from hz_role_responsibility<sup>2</sup></li> <li>• Return new value after update</li> </ul>

## Other Validations

The combination of cust\_account\_role\_id and responsibility\_type should be unique.

# 13

---

## Source System Management API Use

This chapter covers the following topics:

- Source System Management APIs

### Source System Management 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_REFERENCE 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_FALSE,  
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,  
);
```

## 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_referen ce	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
owner_table_name	IN	VARCHAR2	Yes	<p>Validations:</p> <p>Valid lookup code under TCA_OWNER_TABLE lookup type.</p> <p>If multiple_flag = N, then the combination of orig_system, orig_system_reference, and owner_table_name must be unique.</p> <p>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.</p>
owner_table_id	IN	NUMBER	Yes	Validation: The combination of owner_table_name and owner_table_id must be valid.
status	IN	VARCHAR	No	<p>Validation: Validated against the MOSR_STATUS lookup type. Default: A</p> <p>Sync up status and start/end_date_active.</p>
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_ref erence	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. Validated against AR lookup type HZ_CREATED_BY_MODULE.
application_id	IN	NUMBER	No	Comment : Text to indicate application that initiates creation of record.
attribute_category	IN	VARCHAR2	No	
attribute1	IN	VARCHAR2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
attribute2	IN	VARCHAR2	No	
attribute3	IN	VARCHAR2	No	
attribute4	IN	VARCHAR2	No	
attribute5	IN	VARCHAR2	No	
attribute6	IN	VARCHAR2	No	
attribute7	IN	VARCHAR2	No	
attribute8	IN	VARCHAR2	No	
attribute9	IN	VARCHAR2	No	
attribute10	IN	VARCHAR2	No	
attribute11	IN	VARCHAR2	No	
attribute12	IN	VARCHAR2	No	
attribute13	IN	VARCHAR2	No	
attribute14	IN	VARCHAR2	No	
attribute15	IN	VARCHAR2	No	
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 := FND_API .
G_FALSE,
    p_orig_sys_reference_rec              IN          ORIG_SYS_REFERENCE_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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
orig_system_reference	IN	VARCHAR 2	Yes	<p>Validations:</p> <p>The combination of orig_system, orig_system_reference, and owner_table_name must exist in the HZ_ORIG_SYSTEM_REFERENCE table.</p> <p>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.</p>
owner_table_name	IN	VARCHAR 2	Yes	<p>Validations:</p> <p>Valid lookup code under TCA_OWNER_TABLE lookup type.</p> <p>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.</p>
owner_table_id	IN	NUMBER	Yes	Validation: The combination of owner_table_name and owner_table_id must be valid.
status	IN	VARCHAR	No	<p>Validation: Validated against the MOSR_STATUS lookup type.</p> <p>Sync up status and start/end_date_active</p>
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	VARCHAR 2	Yes	Validation : Validated against the MOSR_STATUS lookup type.
old_orig_system_ref	IN	VARCHAR 2	Yes	Validation : The combination of orig_system, old_orig_system_reference, and owner_table_name must exist.
created_by_module	IN	VARCHAR 2	Yes	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
application_id	IN	NUMBER	No	Validation : Cannot be updated if value exists.
attribute_category	IN	VARCHAR 2	No	
attribute1	IN	VARCHAR 2	No	
attribute2	IN	VARCHAR 2	No	
attribute3	IN	VARCHAR 2	No	
attribute4	IN	VARCHAR 2	No	
attribute5	IN	VARCHAR 2	No	
attribute6	IN	VARCHAR 2	No	
attribute7	IN	VARCHAR 2	No	
attribute8	IN	VARCHAR 2	No	
attribute9	IN	VARCHAR 2	No	
attribute10	IN	VARCHAR 2	No	
attribute11	IN	VARCHAR 2	No	
attribute12	IN	VARCHAR 2	No	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
attribute13	IN	VARCHAR 2	No	
attribute14	IN	VARCHAR 2	No	
attribute15	IN	VARCHAR 2	No	
attribute16	IN	VARCHAR 2	No	
attribute17	IN	VARCHAR 2	No	
attribute18	IN	VARCHAR 2	No	
attribute19	IN	VARCHAR 2	No	
attribute20	IN	VARCHAR 2	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 deactivate existing mapping with reason code.

## PL/SQL Procedure

```
PROCEDURE remap_internal_identifier(
    p_init_msg_list           IN          VARCHAR2 := FND_API.
    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,
    x_msg_count                OUT         NOCOPY,
    x_msg_data                 OUT         NOCOPY
)
;
```

## 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.

<b>Parameter Names</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation Default Comment</b>
orig_system	IN	VARCHAR2	No	<p>Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system where status is active.</p> <p>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.</p>
orig_system_reference	IN	VARCHAR2	No	<p>Validation:</p> <p>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.</p>
reason_code	IN	VARCHAR2	Yes	Validation : Validated against the MOSR_STATUS lookup type.

## Bulk Import API Use

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.

To turn off events for D&B bulk import, you must set the HZ\_Execute\_API\_Callouts profile option to All Events Disabled or Only Business Object Events Enabled. After import is complete you must run the Synchronize WF Local Tables concurrent program for the HZ\_Party original system to gather all events.

## 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	VARCHAR 2	No	Default : FND_API.G_FALSE
p_batch_name	IN	VARCHAR 2	Yes	
p_description	IN	VARCHAR 2	No	
p_original_system	IN	VARCHAR 2	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system.
p_load_type	IN	VARCHAR 2	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

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	IN	VARCHAR 2	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	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	Yes	Comment : Message text if x_msg_count = 1

---

## Address and Tax Validation API Use

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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	VARCHAR 2	No	Comment : Country code used to find adapter information for address validation, if p_adapter_id is not passed.
p_module	IN	VARCHAR 2	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	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
username	IN	VARCHAR 2	No	
encrypted_password	IN	VARCHAR 2	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	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	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	VARCHAR 2	Yes	Validation : Mandatory attribute, Validated against FND_TERRITORIES table
enabled_flag	IN	VARCHAR 2	Yes	Validation : Mandatory attribute; Y or N Default: Y
default_flag	IN	VARCHAR 2	No	Validation : Y or N
x_return_status	OUT	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	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	VARCHAR 2	Yes	Validation : Mandatory attribute, Validated against FND_TERRITORIES table
enabled_flag	IN	VARCHAR 2	Yes	Validation : Mandatory attribute; Y or N Default: Y
default_flag	IN	VARCHAR 2	No	Validation : Y or N
x_return_status	OUT	VARCHAR 2	Yes	Comment : API return status
x_msg_count	OUT	NUMBER	Yes	Comment : Number of messages in stack
x_msg_data	OUT	VARCHAR 2	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: <ul style="list-style-type: none"> <li>• Mandatory attribute</li> <li>• valid location_id from the HZ_LOCATIONS table should be passed in.</li> </ul>
p_lock_flag	IN	NUMBER	No	

Parameter Name	Type	Data Type	Required	Validation Default Comment
p_created_by_modul e	IN	VARCHAR2	Yes	Validation: Mandatory attribute. Validated against AR lookup type HZ_CREATED_BY_MODULE.
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	<p>Validation:</p> <ul style="list-style-type: none"><li>Mandatory attribute</li><li>Valid location_id from the HZ_LOCATIONS table should be passed in.</li></ul>
p_lock_flag	IN	NUMBER	No	
p_created_by_modul e	IN	VARCHAR2	No	Validation: Non updateable if value exists, else validated against AR lookup type HZ_CREATED_BY_MODULE.
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 covers the following topics:

- Data Quality Management Search and Duplicate Identification APIs
- Data Quality Management Availability APIs
- Data Quality Management Transformation APIs
- Find Parties Object API
- Data Quality Management Merge 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	OUT	VARCHAR 2	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_searc h_rec_type	No	Comments: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	party_site_li st	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	contact_poi nt_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_poi nt_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
x_num_matches	OUT	NUMBER	No	Comment: Number of parties that matched.
x_return_status	OUT	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_party_search_rec	IN	party_searc h_rec_type	No	Comments: The PL/SQL record structure that has the party search record information.
p_party_site_list	IN	party_site_l ist	No	Comments: The PL/SQL table of records structure that has the party site search record information.
p_contact_list	IN	contact_poi nt_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_poi nt_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	No	Comment: A code indicating whether any errors occurred during processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_msg_count	OUT	NUMBER	No	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_restrict_sql	IN	VARCHAR2	No	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre>SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;;</pre>
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>;

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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\_ctxt\_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	VARCHAR 2	No	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_rule_id	IN	NUMBER	Yes	Comment: Match rule ID.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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_poi_nt_list	No	Comment: The PL/SQL table of records structure that has the contact search record information.
p_contact_point_list	IN	contact_poi_nt_list	No	Comment: The PL/SQL table of records structure that has the contact point search record information.
p_restrict_sql	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_msg_data	OUT	VARCHAR 2	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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_restrict_sql	IN	VARCHAR 2	No	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre>SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;;</pre>
p_match_type	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	OUT	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	VARCHAR 2	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	VARCHAR 2	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\_ctxt\_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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre>SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;;</pre>
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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	VARCHAR 2	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_restrict_sql	IN	VARCHAR 2	No	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre>SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;;</pre>
p_match_type	IN	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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 parameter, 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	OUT	VARCHAR 2	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	VARCHAR 2	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	VARCHAR 2	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          IN  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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre>SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;;</pre>
p_api_name	IN	VARCHAR2	No	<p>Comment: The name of the API to call.</p> <p>Validation: Value can be: FIND_PARTIES, FIND_PARTY_DETAILS, FIND_PERSONS, GET_MATCHING_PARTY_SITES, GET_MATCHING_CONTACTS, or GET_MATCHING_CONTACT_POINTS.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_match_type	IN	VARCHAR2	No	<p>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> <p><b>Note:</b> Thresholds are still applied to filter results.</p> <p>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.</p> <p>Default: As specified in the match rule.</p>
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.
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	<p>Comment: Additional SQL clause to specify a subset of parties to search in. The format of this SQL clause should be as follows.</p> <p>The SQL clause will be substituted into the following SELECT statement:</p> <pre> SELECT party_id FROM hz_staged_parties stage WHERE contains (concat_col,'&lt;intermedia query_string&gt;') AND &lt;p_restrict_sql&gt;; </pre>
p_api_name	IN	VARCHAR2	No	<p>Comment: The name of the API to call.</p> <p>Validation: Value can be: FIND_PARTIES, FIND_PARTY_DETAILS, FIND_PERSONS, GET_MATCHING_PARTY_SITES, GET_MATCHING_CONTACTS, or GET_MATCHING_CONTACT_POINTS.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_match_type	IN	VARCHAR2	No	<p>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> <p><b>Note:</b> Thresholds are still applied to filter results.</p> <p>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.</p> <p>Default: As specified in the match rule.</p>
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.
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 *interMedia* 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	VARCHARA R2	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	VARCHARA R2	No	Comment: Not currently used.
p_has_spc	IN	BOOLEA N	No	Comment: Not currently used. Only in one version of this API.

# Find Parties Object API

## Description

This new DQM object API is a wrapper on the existing public DQM Find Parties API. New features from the standard DQM API facilitate DQM being called as a service from integration software such as Oracle BPEL.

Search criteria can be passed as a database object, match results are returned as an out parameter, and source system IDs are returned along with the party IDs in the results set. Additionally, this API provides the capability to search for parties within a specific source system.

## PL/SQL Function

```
PROCEDURE
find_parties (
  p_init_msg_list    IN      VARCHAR2 := FND_API.G_FALSE,
  p_within_os        IN      VARCHAR2,
  p_rule_id          IN      NUMBER,
  p_search_attr_obj  IN      HZ_SEARCH_ATTR_OBJ_TBL,
  p_party_status     IN      VARCHAR2,
  p_restrict_sql     IN      VARCHAR2,
  p_match_type       IN      VARCHAR2,
  x_search_results_obj OUT NOCOPY HZ_MATCHED_PARTY_OBJ_TBL,
  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 Find Parties Object 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	Yes	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_within_os	IN	VARCHAR2	Yes	Comment: Indicates the name of the source-system to search within.  When the value is null, records from all source systems will be searched.
p_rule_id	IN	NUMBER	Yes	Comment: Indicates the match rule numbers.
p_search_attr_obj	IN	HZ_SEARC H_ATTR_O BJ_TBL	Yes	Comment: The PL/SQL table of records structure that has the search attributes.
p_party_status	IN	NUMBER	Yes	Comment: Indicates the status of parties to be included in the DQM search.
p_restrict_sql	IN	NUMBER	Yes	Comment: Indicates the SQL where clause is used to restrict the search results.
p_match_type	IN	NUMBER	Yes	Comment: Indicates the match type.
x_search_results_obj	OUT NOCOPY	HZ_MATC HED_PART Y_OBJ_TBL	Yes	Comment: The PL/SQL table of records structure that has the search results.
x_return_status	OUT NOCOPY	VARCHAR2	Yes	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT NOCOPY	NUMBER	Yes	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT NOCOPY	VARCHAR2	Yes	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

## Data Quality Management Merge APIs

**PL/SQL Package Name:** HZ\_DUP\_MERGE\_PUB (Create Merge Request API) and HZ\_EXTRACT\_MERGE\_EVENT\_PKG (Get Party Merge and Get Account Merge APIs).

## Create Merge Request API

### Description

Creates a merge request in Oracle Data Librarian. A merge request contains the details of a party or parties that are considered similar and potential candidates for merge. To successfully call this API, pass a list of duplicate party IDs or source system management mappings, and optionally some note text. The merge request will create data in the following tables: HZ\_PARTIES, HZ\_DUP\_BATCH, HZ\_DUP\_SETS, HZ\_DUP\_SET\_PARTIES, HZ\_MERGE\_BATCH, HZ\_MERGE\_PARTIES, HZ\_MERGE\_PARTY\_DETAILS, HZ\_MERGE\_ENTITY\_ATTRIBUTES, and JTF\_NOTES\_B and JTF\_NOTES\_TL.

### PL/SQL Function

```
PROCEDURE
create_dup_merge_request (
  p_init_msg_list  IN      VARCHAR2 := FND_API.G_FALSE,
  p_dup_id_objs   IN      HZ_DUP_ID_OBJ_TBL,
  p_note_text     IN      VARCHAR2,
  x_return_status OUT NOCOPY VARCHAR2,
  x_msg_count     OUT NOCOPY NUMBER,
  x_msg_data      OUT NOCOPY VARCHAR2,
  x_merge_request_id OUT NOCOPY NUMBER
);
```

### Parameter Description and Validation

The following table lists information about the parameters in the Create Merge Request 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	VARCHAR 2	Yes	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_dup_id_objs	IN	HZ_DUP_ID_OBJ_TBL	Yes	Comment: The PL/SQL table of records structure that has the duplicate party ID information.
p_note_text	IN	VARCHAR 2	Yes	Comment: Note text for the merge request.
x_return_status	OUT NOCOPY	VARCHAR 2	Yes	Comment: A code indicating whether any errors occurred during processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_msg_count	OUT NOCOPY	NUMBER	Yes	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT NOCOPY	VARCHAR 2	Yes	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.
x_merge_request_id	OUT NOCOPY	NUMBER	Yes	Comment: Indicates the merge request.

## Other Validations

- Ensure all Party ID's and OS/OSR combinations are valid in TCA.

Here valid means that they exist, and that their status is either *A* or *I*. Since the user can pass either all the Party ID's, or all the OS/OSR's, or a combination of both, there are three separate validations:

- Checks to ensure that all the Party ID's that were passed are valid. If at least one Party ID is invalid, the following message is returned:
  - Message Name: HZ\_DL\_MR\_INV\_PARTYNUM
  - Message Text: "The merge request could not be submitted because the following party ID is invalid: &PARTY\_ID."
- Checks to ensure that all the OS/OSR combinations that were passed are valid. If at least one OS/OSR combination is invalid, the following message is returned:
  - Message Name: HZ\_DL\_MR\_INV\_OSOSR
  - Message Text: "The merge request could not be submitted because the combination of source system and source system reference is invalid: &OSOSR."

**Note:** The token &OSOSR passes a concatenation of &OS and &OSR, joined by a hyphen.

- Checks to ensure that for all records where the Party ID and OS/OSR combination were passed, that they both map to the same TCA party. If at least

one combination of Party ID and OS/OSR combination is invalid, the following message is returned:

- Message Name: HZ\_DL\_MR\_INVALID
- Message Text: "The merge request could not be submitted because this combination of source system and source system reference does not match its related party ID: &OSOSR."

**Note:** The token &OSOSR passes a concatenation of &OS and &OSR, joined by a hyphen.

- Ensure all records intended to be in the merge request are of the same TCA Party Type.
  - If all of the records are not of the same TCA Party Type, that is, Organization or Person, then the following message is returned:
    - Message Name: HZ\_DL\_INVAL\_MR
    - Message Text: "A merge request must contain parties of the same type."
- Ensure none of the records exist in an active merge request.

An active merge request is defined as a merge request with a status of *New*, *Preprocessing*, *Mapping*, *Submitted for Merge*, *Error*, or *AM Queue*.

- For each record found to be in violation of this validation, the following message is returned:
  - Message Name: HZ\_PM\_MR\_EXIST\_DL
  - Message Text: "You cannot select party &ID because it is already in the Data Librarian merge request &REQUEST\_ID."

**Note:** The token &ID passes a concatenation of PARTY\_NAME and PARTY\_ID, joined by a hyphen.

- Ensure the party uniqueness of the records passed to the API.

This is checked because the calling application could pass a series of OS/OSR combinations that might point to the same party record in TCA.

  - If all the records passed point to a single party in TCA, the following message is returned:

- Message Name: HZ\_DL\_ALREADY\_MERGED
- Message Text: "The selected records have already been merged."
- If some the records passed point to a single party in TCA, then create the merge request using only the unique TCA records.

## Get Party Merge Details API

### Description

Finds the details of a particular party merge by passing in the merge batch ID and the master party ID from the merge event.

The party merge object is then extracted, containing the batch name, merge type, autmerge flag value, master party ID and the party numbers, names, and types and source system management mappings for all parties involved in the merge. The merge object is extracted from the following tables: HZ\_PARTIES, HZ\_DUP\_BATCH, HZ\_DUP\_SETS, HZ\_DUP\_SET\_PARTIES, HZ\_MERGE\_BATCH, and HZ\_MERGE\_PARTIES. If a value of Y is passed into the parameter p\_get\_merge\_detail\_flag, historical Merge To and Merge From party details from the HZ\_Merge\_Party\_History table and their associated Source System Mapping details will be retrieved as part of the object. Use this API when to synchronize the merge activities within the Customer Data Hub with external source systems.

### PL/SQL Function

```
PROCEDURE
get_party_merge_event_data (
  p_init_msg_list  IN      VARCHAR2 := FND_API.G_FALSE,
  p_batch_id       IN      NUMBER,
  p_merge_to_party_id IN      NUMBER,
  x_party_merge_obj OUT NOCOPY HZ_PARTY_MERGE_OBJ,
  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 Get Party Merge 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	IN	VARCHAR2	Yes	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_batch_id	IN	NUMBER	Yes	Comment: Indicates the party merge batch numbers.
p_merge_to_party_id	IN	NUMBER	Yes	Comment: Indicates the master party numbers.
p_get_merge_event_data	IN	HZ_PARTY _MERGE_D ETAIL_OBJ ECT	Yes	Comment: Used to control the inclusion of merge history in the result. Default: Value is N.
x_party_merge_obj	OUT NOCOPY	HZ_PARTY _MERGE_O BJ	Yes	Comment: The PL/SQL table of records structure that has the party merge result information.
x_return_status	OUT NOCOPY	VARCHAR2	Yes	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT NOCOPY	NUMBER	Yes	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT NOCOPY	VARCHAR2	Yes	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

## Create Account Merge Request API

### Description

Creates an account merge request in Oracle Data Librarian. A merge request contains the details of a customer or customers that are considered similar and potential candidates for a merge. To successfully call this API, pass account IDs or source system management mappings, and optionally submit a request parameter that indicates to submit the merge.

## PL/SQL Function

```

PROCEDURE CREATE_ACCOUNT_MERGE_REQUEST(
    P_INIT_MSG_LIST IN      VARCHAR2 := FND_API.G_FALSE,
    P_SUBMIT_REQUEST IN      VARCHAR2 := 'N',
    P_ACCOUNT_MERGE_REQUEST_OBJ IN      HZ_ACCOUNT_MERGE_REQUEST_OBJ,
    X_CUSTOMER_MERGE_HEADER_ID OUT     NUMBER,
    X_ACCOUNT_MERGE_REQUEST_ID 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 Create Account Merge Request 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	Yes	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_submit_request	IN	VARCHAR2	N	Comment: This is a request to submit the merge. Default: N.
p_account_merge_re quest_obj	IN	HZ_AC CO UNT_MER GE_QUE ST_OBJ	-	Comment: The PL/SQL table of records structure that has the duplicate Account ID and Account Site information.
x_customer_merge_ header_id	OUT NOCOPY	NUMBER	-	Comment: Indicates Customer Merge Header.
x_account_merge_re quest_id	OUT NOCOPY	NUMBER	-	Comment: Indicates the Customer Merge Request.
x_return_status	OUT NOCOPY	VARCHAR2	-	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT NOCOPY	NUMBER	-	Comment: Indicates how many messages exist on the message stack upon completion of processing.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_msg_data	OUT NOCOPY	VARCHAR2	-	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.

## Get Account Merge Details API

### Description

Finds the details of a particular account merge by passing in the customer merge header ID, which is raised from the merge event.

The account merge object is then extracted, which contains customer merge ID, request ID and the associated party info, merge-to account ID, account number, account name, and source system management mappings for all accounts involved in the merge. The merge object is extracted from the following tables: HZ\_PARTIES, HZ\_CUST\_ACCOUNTS, and RA\_CUSTOMER\_MERGE\_HEADERS. Use this API when trying to synchronize the merge activities within the Customer Data Hub with external source systems.

### PL/SQL Function

```
PROCEDURE
get_account_merge_event_data (
  p_init_msg_list      IN      VARCHAR2 := FND_API.G_FALSE,
  p_customer_merge_header_id IN      NUMBER,
  p_get_merge_detail_flag   IN      VARCHAR2 := 'N'
  x_account_merge_obj     OUT NOCOPY HZ_ACCOUNT.Merge_OBJ,
  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 Get Account Merge 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_init_msg_list	IN	VARCHAR2	Yes	Comment: Indicates if the message stack is initialized. Default: FND_API.G_FALSE.
p_customer_merge_header_id	IN	NUMBER	Yes	Comment: Indicates the customer merge header numbers.
p_get_merge_detail_flag	IN	VARCHAR2	Yes	Comment: Indicates if details of the merged parties must be extracted. Default: No
x_account_merge_obj	OUT NOCOPY	HZ_ACCO UNT_MER GE_OBJ	Yes	Comment: The PL/SQL table of records structure that has the account merge result information.
x_return_status	OUT NOCOPY	VARCHAR2	Yes	Comment: A code indicating whether any errors occurred during processing.
x_msg_count	OUT NOCOPY	NUMBER	Yes	Comment: Indicates how many messages exist on the message stack upon completion of processing.
x_msg_data	OUT NOCOPY	VARCHAR2	Yes	Comment: If exactly one message exists on the message stack upon completion of processing, then this parameter contains that message.



## Extensions API Use

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 NULL,
    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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_row_table	IN	EGO_USER_ATTRS_ROW_TABLE	Yes	<p>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.</p> <p>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.</p> <p>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.)</p> <p>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).</p> <p>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>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	<p>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.</p> <p>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.</p> <p><b>Note:</b> The attribute value must be passed in exactly one of these four fields</p>
				<p>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).</p> <p>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.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
				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_TAB	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.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_commit	IN	VARCHAR 2	No	Comment: Indicates whether to commit work at the end of API processing.
x_failed_row_id_list	OUT	VARCHAR 2	No	Comment: Returns a comma-delimited list of ROW_IDENTIFIERs indicating which attribute group rows failed to be processed.
x_return_status	OUT	VARCHAR 2	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	VARCHAR 2	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 NULL,
    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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_row_table	IN	EGO_USER_ATTRS_RO_W_TABLE	Yes	<p>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.</p> <p>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.</p> <p>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.)</p> <p>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).</p> <p>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>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	<p>EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ.</p> <p>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.</p> <p>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.</p> <p><b>Note:</b> The attribute value must be passed in exactly one of these four fields</p> <p>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).</p> <p>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.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
				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_CHA NGE_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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.
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 NULL,
    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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_row_table	IN	EGO_USER_ATTRS_RO_W_TABLE	Yes	<p>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.</p> <p>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.</p> <p>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.)</p> <p>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).</p> <p>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>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	<p>EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ.</p> <p>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.</p> <p>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.</p> <p><b>Note:</b> The attribute value must be passed in exactly one of these four fields</p> <p>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).</p> <p>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.</p>

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
				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_CHA NGE_TBL E	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.
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 NULL,
    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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_row_table	IN	EGO_USER_ATTRS_RO_W_TABLE	Yes	<p>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.</p> <p>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.</p> <p>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.)</p> <p>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).</p> <p>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>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_attributes_data_table	IN	EGO_USER_ATTR_DATA_TABLE	Yes	<p>EGO_USER_ATTR_DATA_TABLE is a table of EGO_USER_ATTR_DATA_OBJ.</p> <p>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.</p> <p>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.</p> <p><b>Note:</b> The attribute value must be passed in exactly one of these four fields</p> <p>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).</p> <p>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.</p>

---

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
				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_CHANGESTABLE	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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_commit	IN	VARCHAR2	No	Comment: Indicates whether to commit work at the end of API processing.
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.



---

## Formatting API Use

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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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	<p>Comment: Territory code parameter should be passed if the user preferences are not set up.</p> <p>This is the territory code of the input Raw Phone Number.</p> <p>This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.</p>
p_phone_country_code	IN	VARCHAR2	No	<p>Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.</p>
p_phone_area_code	IN	VARCHAR2	No	<p>Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.</p>
p_phone_number	IN	VARCHAR2	No	<p>Comment: This parameter is only in the second procedure, which is used when the contact point ID of the phone number is unknown.</p>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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_	OUT	VARCHAR2	No	Comment : Returns the number after formatting number

## 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 it is 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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 <i>blank</i> valued delimiters. For example, &nbsp; for HTML output.
p_ref_language_code	In	Varchar2	No	Comment: Reference Language (context). Default: Profile option HZ: Reference Language or else the current session NLS Language setting.
p_ref_territory_code	In	Varchar2	No	Comment: Reference Territory (context). Default: Profile option HZ: Reference Territory or else the current session NLS Territory setting.
x_return_status	Out	Varchar2	Yes	Comment: Return status of API. <ul style="list-style-type: none"><li>• S=Success</li><li>• E=Error</li><li>• U=Unexpected Error</li></ul>
x_msg_count	Out	Number	Yes	Comment: Number of error messages returned.
x_msg_data	Out	Number	Yes	Comment: Text of messages returned.
x_formatted_name	Out	Varchar2	Yes	Comment: The formatted name returned as a single string with line breaks.

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
x_formatted_lines_c nt	Out	Number	Yes	Comment: The number of lines in the formatted name.
x_formatted_name_t bl	Out	String_Tbl_T ype	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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
p_person_suffix	In	Varchar2	No	Comment: name suffix. For example, in English, a common custom to distinguish offspring with same given names is to use a generation indicator. The suffix can be a Roman numeral such as <i>II</i> or <i>III</i> , or a string such as <i>Jr.</i> or <i>Sr.</i>
p_person_known_as	In	Varchar2	No	Comment: alternative or also-known-as (AKA) name
p_first_name_phonetic	In	Varchar2	No	Comment: phonetic representation of person's first name
p_first_middle_phonetic	In	Varchar2	No	Comment: phonetic representation of person's middle name
p_first_last_phonetic	In	Varchar2	No	Comment: phonetic representation of person's last name

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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
p_style_code	In	Varchar2	No	<p>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.</p> <p>This parameter is ignored if p_style_format_code is passed.</p>
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 <i>blank</i> valued delimiters. For example, &nbsp; for HTML output.
p_to_language_code	In	Varchar2	No	Comment: Language that is used at the destination location.
p_country_name_lan_g	In	Varchar2	No	<p>Comment: Language used to display the country name</p> <p>Default: Profile option HZ: Language for country name, or if not set then the current session NLS Language setting.</p>
p_from_territory_code	In	Varchar2	No	<p>Comment: Territory of the sender.</p> <p>Default: Profile option HZ: Reference Territory, or if not set then current session NLS Territory setting.</p>
x_return_status	Out	Varchar2	Yes	<p>Comment: Return status of API.</p> <ul style="list-style-type: none"> <li>• S=Success</li> <li>• E=Error</li> <li>• U=Unexpected Error</li> </ul>

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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

Parameter Name	Type	Data Type	Required	Validation, Default, Comment
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, &nbsp; 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.

<b>Parameter Name</b>	<b>Type</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
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.

## Sample Granular API Code

This chapter covers the following topics:

- Sample Code - Demonstrating the APIs

### Sample Code - Demonstrating the APIs

This section provides many sample codes to demonstrate how TCA API can be called to create and update entities. The sample code is provided for PL/SQL API as well as Java API. These are just sample code and do not attempt to cover any complete business case. These sample codes will help developers to quickly code required functionality using TCA API.

## Create an Organization

### PL/SQL Example

```
DECLARE
  p_organization_rec      HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE;
  x_return_status          VARCHAR2(2000);
  x_msg_count              NUMBER;
  x_msg_data               VARCHAR2(2000);
  x_party_id               NUMBER;
  x_party_number            VARCHAR2(2000);
  x_profile_id             NUMBER;

BEGIN
  p_organization_rec.organization_name := 'ABC Corporation';
  p_organization_rec.created_by_module := 'TCA_EXAMPLE';

  hz_party_v2pub.create_organization (
    'T',
    p_organization_rec,
    x_return_status,
    x_msg_count,
    x_msg_data,
    x_party_id,
    x_party_number,
    x_profile_id);

  dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
  1,255));
  dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
  dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

  IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
      LOOP
        dbms_output.put_line(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 HzPartyV2Pub.
        OrganizationRec();
        BigDecimal[] x_party_id = new BigDecimal[1];
        String[] x_party_number = new String[1];
        BigDecimal[] x_profile_id = new BigDecimal[1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartyV2Pub partyV2Pub = new HzPartyV2Pub();

            p_organization_rec.organization_name = "ABC Corporation";
            p_organization_rec.created_by_module = "TCA_EXAMPLE";

            partyV2Pub.createOrganization(
                conn
            , "T"
            , p_organization_rec
            , x_return_status
            , x_msg_count
            , x_msg_data
            , x_party_id
            , x_party_number
            , x_profile_id
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end; ");
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }

            conn.close();
        } catch (ClassNotFoundException e) {
```

```
System.out.println("Driver Not Found: " + e);
    } catch (SQLException e) {
        System.out.println("SQL Error." + e);
    }
}
```

## Out Parameters:

```
x_return_status = S  
x_msg_count = 0  
x_msg_data =  
x_party_id = 1272023  
x_party_number = 1268621  
x_profile_id = 99782
```

### Select Statement to Check Data:

```
select party_id, party_number, party_name, object_version_number,  
created_by_module  
from hz_parties  
where party_id = 1272023;
```

The following table provides information about this select statement.

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,  
object_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 HzLocationV2Pub.
        LocationRec();
        BigDecimal[] x_location_id = new BigDecimal[1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzLocationV2Pub locationV2Pub = new HzLocationV2Pub();

            p_location_rec.country = "US";
            p_location_rec.address1 = "300 Oracle Parkway";
            p_location_rec.address2 = "13th Floor";
            p_location_rec.city = "Redwood Shores";
            p_location_rec.postal_code = "94065";
            p_location_rec.state = "CA";
            p_location_rec.created_by_module = "TCA_EXAMPLE";

            locationV2Pub.createLocation(
                conn
                , "T"
                , p_location_rec
                , x_location_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ) ; end;" );
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }
            conn.close();
        }
    }
}
```

```

        } catch (ClassNotFoundException e) {
            System.out.println("Driver Not Found: " + e);
        } catch (SQLException e) {
            System.out.println("SQL Error." + e);
        }
    }
}

```

## Out Parameters

```

x_location_id = 359086
x_return_status = S
x_msg_count = 0
x_msg_data =

```

## Select Statement to Check Data

```

select location_id, address1, address2, city, state, postal_code,
country,
       object_version_number, created_by_module
from hz_locations
where location_id = 359086;

```

The following tables provide information about this select statement.

LOCATION_ID	ADDRESS1	ADDRESS2	CITY	STATE	POSTAL_CODE
359086	300 Oracle Parkway	13th Floor	Redwood Shores	CA	94065
<hr/>					
COUNTRY		OBJECT_VERSION_NUMBER	CREATED_BY_MODULE		
<hr/>		1	TCA_EXAMPLE		
<hr/>					

## Create a Party Site

The following examples use the organizations and locations created in the previous examples.

## PL/SQL Example

```
DECLARE
  p_party_site_rec          HZ_PARTY_SITE_V2PUB.
  PARTY_SITE_REC_TYPE;
  x_party_site_id            NUMBER;
  x_party_site_number         VARCHAR2(2000);
  x_return_status             VARCHAR2(2000);
  x_msg_count                NUMBER;
  x_msg_data                 VARCHAR2(2000);
BEGIN
  p_party_site_rec.party_id := 1272023;
  p_party_site_rec.location_id := 359086;
  p_party_site_rec.identifying_address_flag := 'Y';
  p_party_site_rec.created_by_module := 'TCA_EXAMPLE';

  hz_party_site_v2pub.create_party_site(
    'T',
    p_party_site_rec,
    x_party_site_id,
    x_party_site_number,
    x_return_status,
    x_msg_count,
    x_msg_data);

  dbms_output.put_line(SubStr('x_return_status = ' || x_return_status,
  1,255));
  dbms_output.put_line('x_msg_count = ' || TO_CHAR(x_msg_count));
  dbms_output.put_line(SubStr('x_msg_data = ' || x_msg_data,1,255));

  IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
      dbms_output.put_line(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
        HzPartySiteV2Pub.PartySiteRec();
        BigDecimal[] x_party_site_id = new BigDecimal[1];
        String[] x_party_site_number = new
        String[1];

        String[] x_return_status = new String
        [1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartySiteV2Pub partysiteV2Pub = new HzPartySiteV2Pub();

            p_party_site_rec.party_id = new BigDecimal("1272023");
            p_party_site_rec.location_id = new BigDecimal ("359086");
            p_party_site_rec.identifying_address_flag = "Y";
            p_party_site_rec.created_by_module = "TCA_EXAMPLE";

            partysiteV2Pub.createPartySite(
                conn
                , "T"
                , p_party_site_rec
                , x_party_site_id
                , x_party_site_number
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := 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_REC_TYPE;
  x_party_site_use_id            NUMBER;
  x_return_status                VARCHAR2(2000);
  x_msg_count                    NUMBER;
  x_msg_data                     VARCHAR2(2000);
  BEGIN
    p_party_site_use_rec.site_use_type := 'SHIP_TO';
    p_party_site_use_rec.party_site_id := 349327;
    p_party_site_use_rec.created_by_module := 'TCA_EXAMPLE';

    hz_party_site_v2pub.create_party_site_use(
      'T',
      p_party_site_use_rec,
      x_party_site_use_id,
      x_return_status,
      x_msg_count,
      x_msg_data);

    dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
      1,255));
    dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
    dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

    IF x_msg_count >1 THEN
      FOR I IN 1..x_msg_count
      LOOP
        dbms_output.put_line(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
HzPartySiteV2Pub.PartySiteUseRec();

        BigDecimal[] x_party_site_use_id = new BigDecimal
[1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartySiteV2Pub partysiteV2Pub = new HzPartySiteV2Pub();

            p_party_site_use_rec.site_use_type = "SHIP_TO";
            p_party_site_use_rec.party_site_id = new BigDecimal("349327");
            p_party_site_use_rec.created_by_module = "TCA_EXAMPLE";

            partysiteV2Pub.createPartySiteUse(
                conn
                , "T"
                , p_party_site_use_rec
                , x_party_site_use_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ) end;");
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }

            conn.close();
        } catch (ClassNotFoundException e) {
    }
```

```
System.out.println("Driver Not Found: " + e);
    } catch (SQLException e) {
        System.out.println("SQL Error." + e);
    }
}
}
```

## Out Parameters

```
x_party_site_use_id = 13523754  
x_return_status = S  
x_msg_count = 0  
x_msg_data =
```

## Select Statement to Check Data

```
select party_site_use_id, party_site_id, site_use_type, primary_per_type  
from hz_party_site_uses  
where party_site_use_id = 13523754;
```

The following table provides information about this select statement.

PARTY_SITE_USE_ID	PARTY_SITE_ID	USE_TYPE	PRIMARY_PER_TYPE
13523754	349327	SHIP_TO	Y

Because the example above is the first SHIP\_TO site use created for party ID 1272023, the primary\_per\_type attribute is set to Y.

## Create a Contact Point (Phone)

This procedure creates a phone number for the same organization created in the first example.

## PL/SQL Example

```
DECLARE
  p_contact_point_rec      HZ_CONTACT_POINT_V2PUB.CONTACT_POINT_REC_TYPE;
  p_edi_rec                 HZ_CONTACT_POINT_V2PUB.EDI_REC_TYPE;
  p_email_rec               HZ_CONTACT_POINT_V2PUB.EMAIL_REC_TYPE;
  p_phone_rec               HZ_CONTACT_POINT_V2PUB.PHONE_REC_TYPE;
  p_telex_rec               HZ_CONTACT_POINT_V2PUB.TELEX_REC_TYPE;
  p_web_rec                 HZ_CONTACT_POINT_V2PUB.WEB_REC_TYPE;
  x_return_status            VARCHAR2(2000);
  x_msg_count                NUMBER;
  x_msg_data                 VARCHAR2(2000);
  x_contact_point_id        NUMBER;

BEGIN
  p_contact_point_rec.contact_point_type := 'PHONE';
  p_contact_point_rec.owner_table_name := 'HZ_PARTIES';
  p_contact_point_rec.owner_table_id := '1272023';
  p_contact_point_rec.primary_flag := 'Y';
  p_contact_point_rec.contact_point_purpose := 'BUSINESS';
  p_phone_rec.phone_area_code := '650';
  p_phone_rec.phone_country_code := '1';
  p_phone_rec.phone_number := '506-7000';
  p_phone_rec.phone_line_type := 'GEN';
  p_contact_point_rec.created_by_module := 'TCA_EXAMPLE';

  hz_contact_point_v2pub.create_contact_point(
    'T',
    p_contact_point_rec,
    p_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
        HzContactPointV2Pub.EdiRec();
        HzContactPointV2Pub.EmailRec       p_email_rec = new
        HzContactPointV2Pub.EmailRec();
        HzContactPointV2Pub.PhoneRec       p_phone_rec = new
        HzContactPointV2Pub.PhoneRec();
        HzContactPointV2Pub.TelexRec       p_telex_rec = new
        HzContactPointV2Pub.TelexRec();
        HzContactPointV2Pub.WebRec         p_web_rec = new
        HzContactPointV2Pub.WebRec();
        BigDecimal[]                      x_contact_point_id = new BigDecimal[1];
        String[]                          x_return_status = new String[1];
        [1];
        BigDecimal[]                      x_msg_count = new BigDecimal[1];
        String[]                          x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzContactPointV2Pub contactpointV2Pub = new HzContactPointV2Pub();

            p_contact_point_rec.contact_point_type = "PHONE";
            p_contact_point_rec.owner_table_name = "HZ_PARTIES";
            p_contact_point_rec.owner_table_id = new BigDecimal("1272023");
            p_contact_point_rec.primary_flag = "Y";
            p_contact_point_rec.contact_point_purpose = "BUSINESS";
            p_phone_rec.phone_area_code = "650";
            p_phone_rec.phone_country_code = "1";
            p_phone_rec.phone_number = "506-7000";
            p_phone_rec.phone_line_type = "GEN";
            p_contact_point_rec.created_by_module = "TCA_EXAMPLE";

            contactpointV2Pub.createContactPoint(
                conn
                , "T"
                , p_contact_point_rec
                , p_edi_rec
                , p_email_rec
                , p_phone_rec
                , p_telex_rec
                , p_web_rec
                , x_contact_point_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );
        }
    }
}
```

## Out Parameters

```
x_contact_point_id = 429523  
x_return_status = S  
x_msg_count = 0  
x_msq_data =
```

## Select Statement to Check Data

```
select contact_point_id, contact_point_type, owner_table_name,  
owner_table_id, primary_flag,  
phone_area_code, phone_country_code, phone_number, phone_line_type,  
raw_phone_number  
from hz_contact_points  
where contact_point_id = 429523;
```

The following tables provide information about this select statement.

CONTACT_POINT_ID	CONTACT_POINT_TYPE	OWNER_TABLE_NAME
429523	PHONE	HZ_PARTIES
OWNER_TABLE_ID	PRIMARY_FLAG	PHONE_AREA_CODE
1272023	Y	650

PHONE_COUNTRY_CODE	PHONE_NUMBER	PHONE_LINE_TYPE
1	506-7000	GEN
<b>RAW_PHONE_NUMBER</b>		
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
        HzPartyContactV2Pub.OrgContactRec();
        BigDecimal[] x_org_contact_id = new BigDecimal[1];
        BigDecimal[] x_party_rel_id = new BigDecimal[1];
        BigDecimal[] x_party_id = new BigDecimal[1];
        String[] x_party_number = new String[1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzPartyContactV2Pub partycontactV2Pub = new HzPartyContactV2Pub();

            p_org_contact_rec.department_code = "ACCOUNTING";
            p_org_contact_rec.job_title = "ACCOUNTS OFFICER";
            p_org_contact_rec.decision_maker_flag = "Y";
            p_org_contact_rec.job_title_code = "APC";
            p_org_contact_rec.created_by_module = "TCA_EXAMPLE";
            p_org_contact_rec.party_rel_rec.subject_id = new BigDecimal
            ("16077");
            p_org_contact_rec.party_rel_rec.subject_type = "PERSON";
            p_org_contact_rec.party_rel_rec.subject_table_name = "HZ_PARTIES";
            p_org_contact_rec.party_rel_rec.object_id = new BigDecimal
            ("1272023");
            p_org_contact_rec.party_rel_rec.object_type = "ORGANIZATION";
            p_org_contact_rec.party_rel_rec.object_table_name = "HZ_PARTIES";
            p_org_contact_rec.party_rel_rec.relationship_code = "CONTACT_OF";
            p_org_contact_rec.party_rel_rec.relationship_type = "CONTACT";
            p_org_contact_rec.party_rel_rec.start_date = Timestamp.valueOf
            ("2001-09-26 00:00:00.0") ;

            partycontactV2Pub.createOrgContact(
                conn
            , "T"
            , p_org_contact_rec
            , x_org_contact_id
            , x_party_rel_id
            , x_party_id
            , x_party_number
            , x_return_status
            , x_msg_count
        }
    }
}
```

```

    , x_msg_data
);

System.out.println( "x_return_status = " + x_return_status[0] );
System.out.println( "x_msg_count = " + x_msg_count[0] );
System.out.println( "x_msg_data = " + x_msg_data[0] );

if (x_msg_count[0].intValue() > 1) {
    OracleCallableStatement ocs = null;
    for (int i=0; i<x_msg_count[0].intValue(); i++) {
        ocs = (OracleCallableStatement)conn.prepareCall(
            "begin ? = fnd_msg_pub.get( p_encoded => 'F' ) end;" );
        ocs.registerOutParameter(1, OracleTypes.VARCHAR);
        ocs.execute();
        System.out.println((i + 1) + ". " + ocs.getString(1));
    }
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}

```

## Out Parameters

```
x_org_contact_id = 411650  
x_party_rel_id = 10091074  
x_party_id = 1272029  
x_party_number = 1268627  
x_return_status = S  
x_msg_count = 0  
x_msq_data =
```

## Select Statement to Check Data

Organization contact record is created.

```
select org_contact_id, party_relationship_id, department_code, title,  
job_title, job_title_code  
from hz_org_contacts  
where org_contact_id = 411650;
```

The following tables provide information about this select statement.

ORG_CONTACT_ID	PARTY_RELATIONSHIP_ID	DEPARTMENT_CODE	TITLE
411650	10091074	ACCOUNTING	DR.

---

JOB_TITLE	JOB_TITLE_CODE
ACCOUNTS OFFICER	APC

---

Two relationship records are created.

```
select relationship_id, subject_id, subject_type, subject_table_name,
       object_id, object_type, object_table_name, directional_flag
          relationship_type, relationship_code, party_id
     from hz_relationships
    where relationship_id = 10091074;
```

The following tables provide information about this select statement.

---

RELATIONSHIP_ID	SUBJECT_ID	SUBJECT_TYPE	SUBJECT_TABLE_NAME
10091074	16077	PERSON	HZ_PARTIES
10091074	1272023	ORGANIZATION	HZ_PARTIES

---

OBJECT_ID	OBJECT_TYPE	OBJECT_TABLE_NAME	DIRECTIONAL_FLAG
1272023	ORGANIZATION	HZ_PARTIES	F
16077	PERSON	HZ_PARTIES	B

---

RELATIONSHIP_TYPE	RELATIONSHIP_CODE	PARTY_ID
CONTACT	CONTACT_OF	1272029
CONTACT	CONTACT	1272029

---

A party record is created.

```
select party_id, party_number, party_name, party_type
  from hz_parties
 where party_id = 1272029;
```

The following table provides information about this select statement.

PARTY_ID	PARTY_NUMBER	PARTY_NAME	PARTY_TYPE
1272029	1268627	John Doe -ABC Corporation-1268627	PARTY_RELATIONS HIP

## Update a Location Record

This procedure updates the new location and sets address2 to null and county to 'San Mateo'.

### PL/SQL Example

To update an attribute to null in the PL/SQL API, you must set the attribute's value to FND\_API.G\_MISS\_XXX.

```

DECLARE

    p_location_rec          HZ_LOCATION_V2PUB.
    LOCATION_REC_TYPE;
    p_object_version_number NUMBER;
    x_return_status          VARCHAR2(2000);
    x_msg_count              NUMBER;
    x_msg_data               VARCHAR2(2000);

    BEGIN
        p_location_rec.location_id := 359086;
        p_location_rec.address2 := FND_API.G_MISS_CHAR;
        p_location_rec.county := 'San Mateo';
                                -- get the object version number of the current
        record
            select object_version_number
            into p_object_version_number
            from hz_locations
            where location_id = 359086;
        hz_location_v2pub.update_location(
            'T',
            p_location_rec,
            p_object_version_number,
            x_return_status,
            x_msg_count,
            x_msg_data);

        dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
            1,255));
        dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
        dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

        IF x_msg_count >1 THEN
            FOR I IN 1..x_msg_count
            LOOP
                dbms_output.put_line(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 HzLocationV2Pub.
        LocationRec();

        String[]                                         x_return_status = new String
        [1];
        BigDecimal[]                                     x_msg_count = new BigDecimal
        [1];
        String[]                                         x_msg_data = new String[1];
        BigDecimal[]                                     p_object_version_number = new
        BigDecimal[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzLocationV2Pub locationV2Pub = new HzLocationV2Pub();

            p_location_rec.location_id = new BigDecimal("359086");
            p_location_rec.address2 = null;
            p_location_rec.county = "San Mateo";

            -- please note, you should pass the current value of object
            -- version number column for this record
            p_object_version_number[0] = new BigDecimal("1");

            locationV2Pub.updateLocation(
                conn
                , "T"
                , p_location_rec
                , p_object_version_number
                , x_return_status
                , x_msg_count
                , x_msg_data
                );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ) ; end;");
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }
        }
    }
}

```

```

        conn.close();
    } catch (ClassNotFoundException e) {
        System.out.println("Driver Not Found: " + e);
    } catch (SQLException e) {
        System.out.println("SQL Error." + e);
    }
}
}
}

```

### Select Statement to Check Data

```

select address1, address2, county, object_version_number
from hz_locations
where location_id = 359086;

```

The following table provides information about this select statement.

ADDRESS1	ADDRESS2	COUNTY	OBJECT_VERSION_NUMBER
300 Oracle Parkway		San Mateo	2

### Updating Geometry Column Using hz\_location\_v2pub.update\_location TCA API

The following sample code explains the method of passing parameter for updating Geometry column value in HZ\_LOCATIONS table.

```

DECLARE

x_geo mdsys.sdo_geometry ;
l_longitude number;
l_latitude number;
p_location_rec hz_location_v2pub.location_rec_type ;
BEGIN

l_longitude := &longitude_value_of_the_location;
l_latitude := &latitude_value_of_the_location;

x_geo := MDSYS.SDO_GEOMETRY(2001,
8307,
MDSYS.SDO_POINT_TYPE(l_longitude, l_latitude, NULL),
NULL,
NULL);

-- Assign x_geo to geometry column in p_location_rec record_type

if (l_longitude is not null AND
l_latitude is not null AND
x_geo is not null ) THEN

p_location_rec.geometry := x_geo ;

end if;

-- Call create / update location API as usual
-- hz_location_v2pub.create_location ( p_location_rec ....); OR
-- hz_location_v2pub.update_location ( p_location_rec ....);

END;
/

```

## Create a Customer Account

### PL/SQL Example

```
DECLARE
  p_cust_account_rec          HZ_CUST_ACCOUNT_V2PUB.
  CUST_ACCOUNT_REC_TYPE;
  p_person_rec                 HZ_PARTY_V2PUB.
  PERSON_REC_TYPE;
  p_customer_profile_rec       HZ_CUSTOMER_PROFILE_V2PUB.
  CUSTOMER_PROFILE REC_TYPE;
  x_cust_account_id            NUMBER;
  x_account_number             VARCHAR2(2000);
  x_party_id                   NUMBER;
  x_party_number                VARCHAR2(2000);
  x_profile_id                 NUMBER;
  x_return_status               VARCHAR2(2000);
  x_msg_count                  NUMBER;
  x_msg_data                    VARCHAR2(2000);
BEGIN
  p_cust_account_rec.account_name := 'John''s A/c';
  p_cust_account_rec.created_by_module := 'TCA_EXAMPLE';
  p_person_rec.person_first_name := 'John';
  p_person_rec.person_last_name := 'Smith';

  hz_cust_account_v2pub.create_cust_account(
    'T',
    p_cust_account_rec,
    p_person_rec,
    p_customer_profile_rec,
    'F',
    x_cust_account_id,
    x_account_number,
    x_party_id,
    x_party_number,
    x_profile_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

  dbms_output.put_line(SubStr('x_return_status = ' || x_return_status,
  1,255));
  dbms_output.put_line('x_msg_count = ' || TO_CHAR(x_msg_count));
  dbms_output.put_line(SubStr('x_msg_data = ' || x_msg_data,1,255));

  IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
      dbms_output.put_line(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
        HzCustAccountV2Pub.CustAccountRec();
        HzPartyV2Pub.PersonRec p_person_rec = new HzPartyV2Pub.PersonRec();
        HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec =
        new HzCustomerProfileV2Pub.CustomerProfileRec();
        BigDecimal[] x_cust_account_id = new BigDecimal[1];
        String[] x_account_number = new String
        [1];
        BigDecimal[] x_party_id = new BigDecimal[1];
        String[] x_party_number = new String[1];
        BigDecimal[] x_profile_id = new BigDecimal[1];
        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzCustAccountV2Pub custaccountV2Pub = new HzCustAccountV2Pub();

            p_cust_account_rec.account_name = "John A/c";
            p_person_rec.person_first_name = "John";
            p_person_rec.person_last_name = "Smith";
            p_cust_account_rec.created_by_module = "TCA_EXAMPLE";

            custaccountV2Pub.createCustAccount(
                conn
                , "T"
                , p_cust_account_rec
                , p_person_rec
                , p_customer_profile_rec
                , "F"
                , x_cust_account_id
                , x_account_number
                , x_party_id
                , x_party_number
                , x_profile_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
        }
    }
}
```

```

        System.out.println( "x_msg_data = " + x_msg_data[0] );

        if (x_msg_count[0].intValue() > 1) {
            OracleCallableStatement ocs = null;
            for (int i=0; i<x_msg_count[0].intValue(); i++) {
                ocs = (OracleCallableStatement)conn.prepareCall(
                    "begin ? := fnd_msg_pub.get( p_encoded => 'F' ) ; end;");
                ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                ocs.execute();
                System.out.println((i + 1) + ". " + ocs.getString(1));
            }
        }
        conn.close();
    } catch (ClassNotFoundException e) {
        System.out.println("Driver Not Found: " + e);
    } catch (SQLException e) {
        System.out.println("SQL Error." + e);
    }
}
}

```

## Out Parameters

```

x_cust_account_id = 3472
x_account_number = 1745
x_party_id = 3575
x_party_number = 1647
x_profile_id = 1483
x_return_status = S
x_msg_count = 0
x_msg_data =

```

## Select Statement to Check Data

Created a record in the HZ\_CUST\_ACCOUNTS table.

```

select cust_account_id, party_id, account_number, account_name
from hz_cust_accounts
where cust_account_id = 3472;

```

The following table provides information about this select statement.

CUST_ACCOUNT_ID	PARTY_ID	ACCOUNT_NUMBER	ACCOUNT_NAME
3472	3575	1745	John's A/c

Created a record in the HZ\_PARTIES table

```

select party_id, party_number, party_name, party_type
from hz_parties
where party_id = 3575;

```

The following table provides information about this select statement.

PARTY_ID	PARTY_NUMBER	PARTY_NAME	PARTY_TYPE
3575	1647	John Smith	PERSON

Created a record in hz\_person\_profiles table.

```
select person_profile_id, party_id, person_first_name, person_last_name
from hz_person_profiles
where party_id = 3575;
```

The following table provides information about this select statement.

PERSON_PROFILE_ID	PARTY_ID	PERSON_FIRST_NAME	PERSON_LAST_NAME
1483	3575	John	Smith

Created a record in hz\_customer\_profiles.

```
select cust_account_profile_id, cust_account_id, profile_class_id,
collector_id
from hz_customer_profiles
where cust_account_id = 3472;
```

The following table provides information about this select statement.

CUST_ACCOUNT_PROFILE_ID	CUST_ACCOUNT_ID	PROFILE_CLASS_ID	COLLECTOR_ID
3691	3472	0	1000

## Create a Customer Account Site

This procedure creates a site for the account created in previous example, using an existing site.

## PL/SQL Example

```
DECLARE
  p_cust_acct_site_rec hz_cust_account_site_v2pub.cust_acct_site_rec_type;
  x_return_status          VARCHAR2(2000);
  x_msg_count              NUMBER;
  x_msg_data                VARCHAR2(2000);
  x_cust_acct_site_id      NUMBER;
BEGIN
  p_cust_acct_site_rec.cust_account_id := 3472;
  p_cust_acct_site_rec.party_site_id := 1024;
  p_cust_acct_site_rec.language := 'US';
  p_cust_acct_site_rec.created_by_module := 'TCA-EXAMPLE';

  hz_cust_account_site_v2pub.create_cust_acct_site(
    'T',
    p_cust_acct_site_rec,
    x_cust_acct_site_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

  dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
  1,255));
  dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
  dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

  IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
      dbms_output.put_line(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 BigDecimal
        [1];

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzCustAccountSiteV2Pub custaccountsiteV2Pub = new
            HzCustAccountSiteV2Pub();

            p_cust_acct_site_rec.cust_account_id = new BigDecimal("3472");
            p_cust_acct_site_rec.party_site_id = new BigDecimal("1024");
            p_cust_acct_site_rec.language = "US";
            p_cust_acct_site_rec.created_by_module = "TCA_EXAMPLE";

            custaccountsiteV2Pub.createCustAcctSite(
                conn
                , "T"
                , p_cust_acct_site_rec
                , x_cust_account_site_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }
            conn.close();
        }
    }
}
```

```

        } catch (ClassNotFoundException e) {
            System.out.println("Driver Not Found: " + e);
        } catch (SQLException e) {
            System.out.println("SQL Error." + e);
        }
    }
}

```

### **Out parameters**

```

x_return_status = S
x_msg_count = 0
x_msg_data =
x_cust_acct_site_id = 3580

```

### **Select statement to check data**

```

select cust_acct_site_id, cust_account_id, party_site_id, org_id
from hz_cust_acct_sites_all
where cust_acct_site_id = 3580;

```

The following table provides information about this select statement.

CUST_ACCT_SITE_ID	CUST_ACCOUNT_ID	PARTY_SITE_ID	ORG_ID
3580	3472	1024	204

### **Create a Customer Account Site Use**

This procedure creates a customer account site use for the customer account site created in the previous example.

## PL/SQL Example

```
DECLARE
  p_cust_site_use_rec      HZ_CUST_ACCOUNT_SITE_V2PUB.
  CUST_SITE_USE_REC_TYPE;
  p_customer_profile_rec   HZ_CUSTOMER_PROFILE_V2PUB.
  CUSTOMER_PROFILE_REC_TYPE;
  x_site_use_id            NUMBER;
  x_return_status          VARCHAR2(2000);
  x_msg_count              NUMBER;
  x_msg_data                VARCHAR2(2000);
BEGIN
  p_cust_site_use_rec.cust_acct_site_id := 3580;
  p_cust_site_use_rec.site_use_code := 'INV';
  p_cust_site_use_rec.location := 'TCA';
  p_cust_site_use_rec.created_by_module := 'TCA_EXAMPLE';
  hz_cust_account_site_v2pub.create_cust_site_use(
    'T',
    p_cust_site_use_rec,
    p_customer_profile_rec,
    '',
    '',
    x_site_use_id,
    x_return_status,
    x_msg_count,
    x_msg_data);

  dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
  1,255));
  dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
  dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

  IF x_msg_count >1 THEN
    FOR I IN 1..x_msg_count
    LOOP
      dbms_output.put_line(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 = new
        HzCustAccountSiteV2Pub.CustSiteUseRec();
        HzCustomerProfileV2Pub.CustomerProfileRec p_customer_profile_rec =
        new HzCustomerProfileV2Pub.CustomerProfileRec();
        BigDecimal[]                                x_site_use_id = new BigDecimal
        [1];

        String[]                                     x_return_status = new String
        [1];
        BigDecimal[]                                 x_msg_count = new BigDecimal
        [1];
        String[]                                     x_msg_data = new String[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzCustAccountSiteV2Pub custaccountsiteV2Pub = new
            HzCustAccountSiteV2Pub();

            p_cust_site_use_rec.cust_acct_site_id = new BigDecimal("3580");
            p_cust_site_use_rec.site_use_code = "INV";
            p_cust_site_use_rec.location = "TCA";
            p_cust_site_use_rec.created_by_module = "TCA_EXAMPLE";

            custaccountsiteV2Pub.createCustSiteUse(
                conn
                , "T"
                , p_cust_site_use_rec
                , p_customer_profile_rec
                , "T"
                , "T"
                , x_site_use_id
                , x_return_status
                , x_msg_count
                , x_msg_data
            );
        }

        System.out.println( "x_return_status = " + x_return_status[0] );
        System.out.println( "x_msg_count = " + x_msg_count[0] );
        System.out.println( "x_msg_data = " + x_msg_data[0] );

        if (x_msg_count[0].intValue() > 1) {
            OracleCallableStatement ocs = null;
            for (int i=0; i<x_msg_count[0].intValue(); i++) {
```

```

        ocs = (OracleCallableStatement)conn.prepareCall(
            "begin ? := fnd_msg_pub.get( p_encoded => 'F' ) ; end; ");
        ocs.registerOutParameter(1, OracleTypes.VARCHAR);
        ocs.execute();
        System.out.println((i + 1) + ". " + ocs.getString(1));
    }
}

conn.close();
} catch (ClassNotFoundException e) {
    System.out.println("Driver Not Found: " + e);
} catch (SQLException e) {
    System.out.println("SQL Error." + e);
}
}
}

```

## Out Parameters

```

x_site_use_id = 3756
x_return_status = S
x_msg_count = 0
x_msg_data =

```

## Select Statement to Check Data

```

select site_use_id, cust_acct_site_id, site_use_code, location, org_id
from hz_cust_site_uses_all
where site_use_id = 3756;

```

The following table provides information about this select statement.

SITE_USE_ID	CUST_ACCT_SITE_ID	SITE_USE_CODE	LOCATION	ORG_ID
3756	3580	INV	TCA	204

## Update a Customer Account Record

This example updates the customer account record that was created earlier by setting the account\_name column to null and the customer\_type column to *R*.

### PL/SQL Example

To update an attribute to null in the PL/SQL API, you need to set the attribute's value to FND\_API.G\_MISS\_XXX.

```

DECLARE
    p_cust_account_rec          HZ_CUST_ACCOUNT_V2PUB.
    CUST_ACCOUNT_REC_TYPE;
    p_object_version_number      NUMBER;
    x_return_status               VARCHAR2(2000);
    x_msg_count                  NUMBER;
    x_msg_data                   VARCHAR2(2000);
    BEGIN
        p_cust_account_rec.cust_account_id := 3472;
        p_cust_account_rec.customer_type := 'R';
        p_cust_account_rec.account_name := FND_API.G_MISS_CHAR;

        -- get the object version number of the current record
        select object_version_number
        into p_object_version_number
        from hz_cust_accounts
        where cust_account_id = 3472;

        hz_cust_account_v2pub.update_cust_account(
            'T',
            p_cust_account_rec,
            p_object_version_number,
            x_return_status,
            x_msg_count,
            x_msg_data);

        dbms_output.put_line(SubStr('x_return_status = '||x_return_status,
            1,255));
        dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
        dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));

        IF x_msg_count >1 THEN
            FOR I IN 1..x_msg_count
            LOOP
                dbms_output.put_line(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
        HzCustAccountV2Pub.CustAccountRec();

        String[] x_return_status = new String[1];
        BigDecimal[] x_msg_count = new BigDecimal[1];
        String[] x_msg_data = new String[1];
        BigDecimal[] p_object_version_number = new
        BigDecimal[1];

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
            OracleConnection conn = (OracleConnection)
                DriverManager.getConnection (
                    "jdbc:oracle:thin:@ap103fam:1521:findv115",
                    "apps", "apps");

            HzCustAccountV2Pub custAccountV2Pub = new HzCustAccountV2Pub();

            p_cust_account_rec.cust_account_id = new BigDecimal("3472");
            p_cust_account_rec.customer_type = "R";
            p_cust_account_rec.account_name = null;

            -- please note, you should pass the current value of object
            -- version number column for this record
            p_object_version_number [0]= new BigDecimal("1");

            custAccountV2Pub.updateCustAccount(
                conn
                , "F"
                , p_cust_account_rec
                , p_object_version_number
                , x_return_status
                , x_msg_count
                , x_msg_data
            );

            System.out.println( "x_return_status = " + x_return_status[0] );
            System.out.println( "x_msg_count = " + x_msg_count[0] );
            System.out.println( "x_msg_data = " + x_msg_data[0] );

            if (x_msg_count[0].intValue() > 1) {
                OracleCallableStatement ocs = null;
                for (int i=0; i<x_msg_count[0].intValue(); i++) {
                    ocs = (OracleCallableStatement)conn.prepareCall(
                        "begin ? := fnd_msg_pub.get( p_encoded => 'F' ); end;");
                    ocs.registerOutParameter(1, OracleTypes.VARCHAR);
                    ocs.execute();
                    System.out.println((i + 1) + ". " + ocs.getString(1));
                }
            }

            conn.close();
        } catch (ClassNotFoundException e) {
    }
}

```

```
System.out.println("Driver Not Found: " + e);
    } catch (SQLException e) {
        System.out.println("SQL Error." + e);
    }
}
}
```

## Out Parameters

```
p_object_version_number = 2  
x_return_status = S  
x_msg_count = 0  
x_msg_data =
```

## Select Statement to Check Data

```
select cust_account_id, customer_type, account_name  
from hz_cust_accounts  
where cust_account_id = 3472;
```

The following table provides information about this select statement.

CUST_ACCOUNT_ID	CUSTOMER_TYPE	ACCOUNT_NAME
3472	R	

## 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 ( Internal Name
    )
      , 'Test Data Updated'        --ATTR_VALUE_STR ( Value for Attribute1
    )
      ,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_return_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).
message_text);
        DBMS_OUTPUT.Put_Line('Msg Type:  '||l_errors_tbl(i).
message_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_end_time -
l_start_time)*86400));

END;

```

Run the following query and verify the data.

```
SELECT * FROM HZ_ORG_PROFILES_EXT_VL WHERE ORGANIZATION_PROFILE_ID = :1
```



# 20

---

## Business Objects

This chapter covers the following topics:

- Business Objects Overview
- Customer Account Business Object
- Customer Account Contact Business Object
- Customer Account Site Business Object
- EDI Business Object
- EFT Business Object
- E-Mail Business Object
- Organization Business Object
- Organization Contact Business Object
- Organization Customer Business Object
- Party Site Business Object
- Person Business Object
- Person Customer Business Object
- Phone Business Object
- SMS Business Object
- Telex Business Object
- Web Business Object
- Contact Person Information Business Structure
- Customer Account Site Use Business Structure
- Customer Profile Business Structure
- Employment History Business Structure
- Financial Report Business Structure

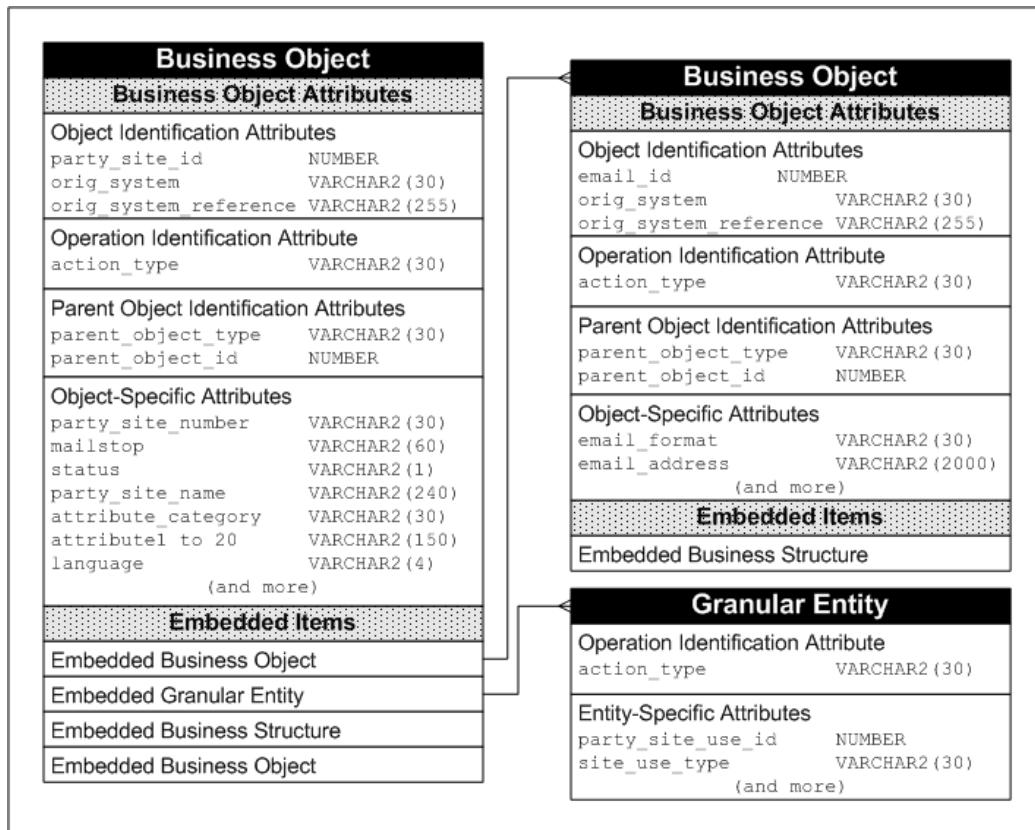
- Location Business Structure

## Business Objects Overview

A business object is an abstract grouping of Oracle Trading Community Architecture (TCA) entities to form an operable, logical business unit. You perform operations and services, for example through APIs and business events, on the business object as a whole, abstracting away the underlying TCA data model.

This diagram provides an example of the hierarchical structure of business objects and their contents: attributes and embedded items. See:

- Business Object Attributes, page 20-2
- Business Object Embedded Items, page 20-3



## Business Object Attributes

Each business object contains attributes that describe the business object at the root object level. Aside from object-specific attributes, all business objects have:

- **Object Identification Attributes:** Uniquely identifies the business object, for

example using columns such as PERSON\_ID, or ORIG\_SYSTEM and ORIG\_SYSTEM\_REFERENCE.

- **ACTION\_TYPE:** Defines operations performed on the business object. This attribute value is relevant to data hubs and only for event-specific extraction of data.
- **Parent Object Identification Attributes:**
  - *PARENT\_OBJECT\_TYPE:* Indicates the type of the parent business object, if any, for example Organization, Person, or Organization Contact.
  - *PARENT\_OBJECT\_ID:* Uniquely identifies the parent object, if any.

These attributes are informational in nature and read only.

- **Internal Administrative Attributes:** Provides administrative information regarding the business object, using, for example, CREATED\_BY\_NAME and LAST\_UPDATE\_DATE columns. These attributes are read only and internally maintained.

## Business Object Embedded Items

Aside from object-level attributes, a business object can also contain any of these embedded items, which have their own attributes:

- **Other, embedded business objects:** Which are considered atomic and logically abstracted as an entity within the parent business object. Items that exist within the embedded business object are packaged together and not exposed as items that can be viewed outside the context of the top level parent business object. An embedded business object can itself contain other embedded items.
- **Business structures:** Which are essentially the same as business objects except that you cannot perform operations and services on them individually. A business structure itself can contain embedded structures and granular entities, but not business objects.
- **Embedded granular TCA entities:** Which map to specific TCA tables. You can perform operations and services on these entities using granular APIs and business events. A granular entity within a business object is also known as an embedded entity.

With this paradigm, business objects allow logical and hierarchical representations of realistic business units. On the other hand, granular entities and the underlying TCA schema can be too detailed and complicated to relate directly to high-level conceptual business scenarios. The hierarchical nature of business objects lets you perform operations and services at the appropriate level, on specific objects.

## Rules for Embedded Items

Business objects are flexible and unrestrictive, with different rules for including embedded items within the object. These rules are identified in the business object definition and enforced at the API level.

- **Requirement (Mandatory or Optional):** Mandatory items must be embedded in a business object for the business object to be valid. If the API validates this rule, then you cannot create a business object without all its mandatory items.
- **Instance (Single or Multiple):** Single instance embedded items can only exist once in a business object, while multiple instance items can exist more than once. Both types of item can be either mandatory or optional.

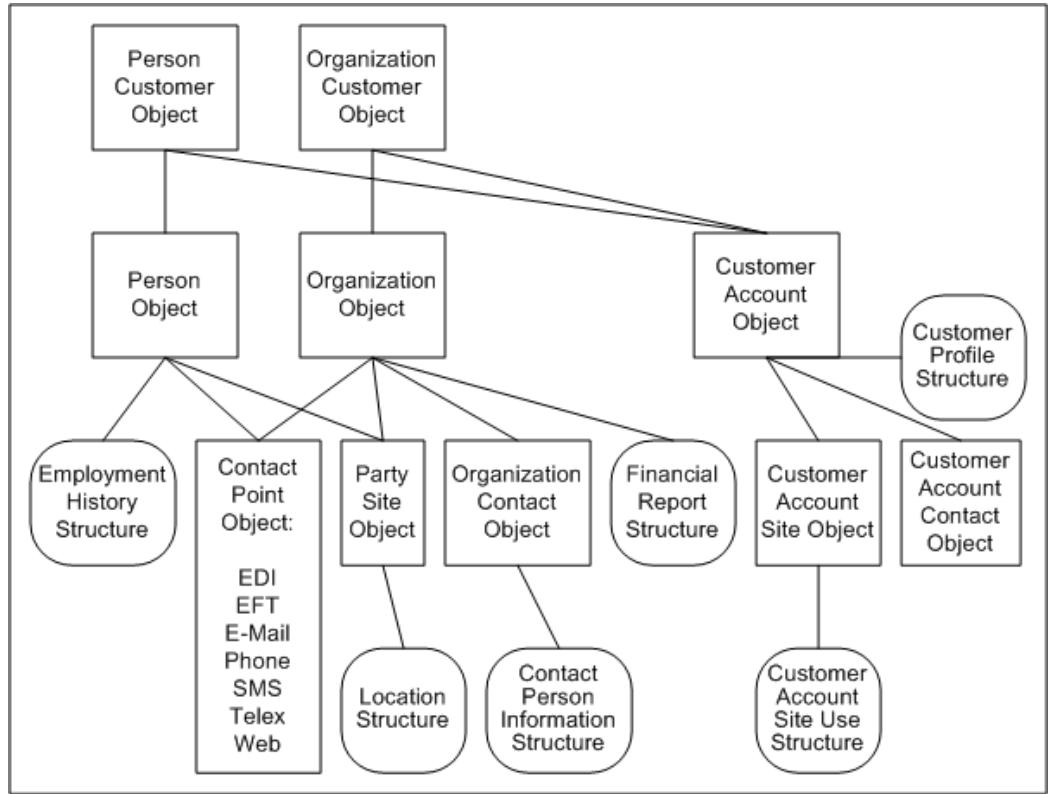
## Seeded Business Objects

Oracle Trading Community Architecture provides a comprehensive set of business objects and structures that covers the entire TCA data model. These objects and structures have defined rules for their embedded items. See: Rules for Embedded Items, page 20-4.

This diagram shows the seeded business objects and structures, as well as the hierarchical manner in which they are embedded. For example, Person Customer and Organization Customer are the highest level business objects, each containing one or more Customer Account business objects, and one Person or Organization business object, respectively.

**Note:** Not shown in the diagram:

- Granular entities are also embedded within the business objects.
- The Party Site and Organization Contact business objects can also have embedded E-Mail, Phone, Telex, or Web business objects.



For details on each seeded business object and structure, including the listing of their embedded granular entities, see:

- Person Customer Business Object, page 20-43; Organization Customer Business Object, page 20-34
  - Person Business Object, page 20-38; Organization Business Object, page 20-23
    - Employment History Business Structure, page 20-65
    - Financial Report Business Structure, page 20-68
    - Organization Contact Business Object, page 20-31
    - Contact Person Information Business Structure, page 20-53
  - Contact Point Business Object
    - EDI Business Object, page 20-16
    - EFT Business Object, page 20-19
    - E-Mail Business Object, page 20-21

- Phone Business Object, page 20-44
- SMS Business Object, page 20-46
- Telex Business Object, page 20-49
- Web Business Object, page 20-51
- Party Site Business Object, page 20-35
  - Location Business Structure, page 20-70
- Customer Account Business Object, page 20-6
  - Customer Profile Business Structure, page 20-61
  - Customer Account Contact Business Object, page 20-11
  - Customer Account Site Business Object, page 20-14
    - Customer Account Site Use Business Structure, page 20-56

## Customer Account Business Object

### Integration Features

This table shows the seeded integration features for the Customer Account business object.

*Customer Account Business Object Integration Features*

Type	Name	Code	Procedure
Object	Customer Account Business Object API, page 24-3	HZ_CUST_SITE_USE_BO	HZ_CUST_SITE_USE_BO.create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Customer Account Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Customer Account Site, page 20-14	Business Object	Optional	Multiple
Customer Account Contact, page 20-11	Business Object	Optional	Multiple
Customer Profile, page 20-61	Business Structure	Mandatory	Single
Customer Account Relationship	Granular Entity	Optional	Multiple
Bank Account Use	Entity (not TCA)	Optional	Multiple
Payment Method	Entity (not TCA)	Optional	Single
Customer Account Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
cust_acct_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE

---

<b>Attribute</b>	<b>Data Type</b>
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
account_number	VARCHAR2(30)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
global_attribute1 to global_attribute20	VARCHAR2(150)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
fob_point	VARCHAR2(30)
freight_term	VARCHAR2(30)
ship_partial	VARCHAR2(1)
ship_via	VARCHAR2(30)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
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)
pricing_event	VARCHAR2(30)

---

Attribute	Data Type
status_update_date	DATE
autopay_flag	VARCHAR2(1)
notify_flag	VARCHAR2(1)
last_batch_id	NUMBER
selling_party_id	NUMBER
cancel_unshipped_lines_flag	VARCHAR2(1)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Customer Account Contact Business Object

### Integration Features

This table shows the seeded integration features for the Customer Account Contact business object.

#### *Customer Account Contact Business Object Integration Features*

Type	Name	Code	Procedure
Object	Customer Account Contact Business Object API, page 24-25	HZ_CUST_ACCT_CONT ACT_BO	HZ_CUST_ACCT_CONTACT_BO. create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### ***Customer Account Contact Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Account Contact Role	Granular Entity	Optional	Multiple
Relationship	Reference	Mandatory	Single
Customer Account Contact Source System Information		Optional	Multiple

## **Attributes**

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
cust_acct_contact_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE

<b>Attribute</b>	<b>Data Type</b>
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
cust_acct_site_id	NUMBER
primary_flag	VARCHAR2(1)
role_type	VARCHAR2(30)
source_code	VARCHAR2(150)
attribute_category	VARCHAR2(30)
attribute1 to attribute25	VARCHAR2(150)
relationship_id	NUMBER
contact_person_id	NUMBER
contact_person_orig_system	VARCHAR2(30)
contact_person_osr	VARCHAR2(255)
relationship_code	VARCHAR2(30)
relationship_type	VARCHAR2(30)
start_date	DATE

## Related Topics

[Business Objects Overview, page 20-2](#)

# Customer Account Site Business Object

## Integration Features

This table shows the seeded integration features for the Customer Account Site business object.

***Customer Account Site Business Object Integration Features***

Type	Name	Code	Procedure
Object	Customer Account Site Business Object API, page 24-33	HZ_CUST_ACCT_SITE_B O	HZ_CUST_ACCT_SITE_BO. create_object

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Customer Account Site Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Customer Account Site Use, page 20-56	Business Structure	Optional	Multiple
Customer Account Contact, page 20-11	Business Object	Optional	Multiple
Party Site	Reference		
Account Site Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

---

<b>Attribute</b>	<b>Data Type</b>
cust_acct_site_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
global_attribute1 to global_attribute20	VARCHAR2(150)
customer_category_code	VARCHAR2(30)
language	VARCHAR2(4)
key_account_flag	VARCHAR2(1)

---

<b>Attribute</b>	<b>Data Type</b>
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)
party_site_id	NUMBER
party_site_orig_system	VARCHAR2(30)
party_site_osr	VARCHAR2(255)
org_id	NUMBER

## Related Topics

[Business Objects Overview, page 20-2](#)

## EDI Business Object

### Integration Features

This table shows the seeded integration features for the EDI business object.

#### ***EDI Business Object Integration Features***

Type	Name	Code	Procedure
Object	EDI Business Object API, page 24-42	HZ_EDI_CP_BO	HZ_EDI_CP_BO.create_object

#### **Embedded Items**

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### ***EDI Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

#### **Attributes**

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
edi_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)

---

<b>Attribute</b>	<b>Data Type</b>
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
edi_transaction_handling	VARCHAR2(25)
edi_id_number	VARCHAR2(30)
edi_payment_method	VARCHAR2(30)
edi_payment_format	VARCHAR2(30)
edi_remittance_method	VARCHAR2(30)

---

---

Attribute	Data Type
edi_remittance_instruction	VARCHAR2(30)
edi_tp_header_id	NUMBER
edi_ece_tp_location_code	VARCHAR2(40)

---

## Related Topics

[Business Objects Overview, page 20-2](#)

## EFT Business Object

### Integration Features

This table shows the seeded integration features for the EFT business object.

#### *EFT Business Object Integration Features*

---

Type	Name	Code	Procedure
Object	EFT Business Object API, page 24-51	HZ_EFT_CP_BO	HZ_EFT_CP_BO.create_object

---

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### *EFT Business Object Embedded Items*

---

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

---

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
eft_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)

Attribute	Data Type
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
eft_transmission_program_id	NUMBER
eft_printing_program_id	NUMBER
eft_user_number	VARCHAR2(30)
eft_swift_code	VARCHAR2(30)

## Related Topics

[Business Objects Overview, page 20-2](#)

## E-Mail Business Object

### Integration Features

This table shows the seeded integration features for the E-Mail business object.

#### *E-Mail Business Object Integration Features*

Type	Name	Code	Procedure
Object	E-Mail Business Object API, page 24-59	HZ_EMAIL_CP_BO	HZ_EMAIL_CP_BO.create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### **E-Mail Business Object Embedded Items**

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

## **Attributes**

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
email_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)

Attribute	Data Type
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
email_format	VARCHAR2(30)
email_address	VARCHAR2(2000)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Organization Business Object

### Integration Features

This table shows the seeded integration features for the Organization business object.

### ***Organization Business Object Integration Features***

Type	Name	Code	Procedures
Object	Organization Business Object API, page 24-67	HZ_ORGANIZATION_B O	HZ_ORGANIZATION_BO. create_object
Business Event			<ul style="list-style-type: none"> <li>• Create (orgBO.create)</li> <li>• Update (orgBO.update)</li> </ul>

### **Embedded Items**

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

### ***Organization Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Extensible Attributes	Granular Entity	Optional	Multiple
Organization Contact, page 20-31	Business Object	Optional	Multiple
Party Site, page 20-35	Business Object	Optional	Multiple
Phone, page 20-44	Business Object	Optional	Multiple
Telex, page 20-49	Business Object	Optional	Multiple
E-Mail, page 20-21	Business Object	Optional	Multiple
Web, page 20-51	Business Object	Optional	Multiple
EDI, page 20-16	Business Object	Optional	Multiple
EFT, page 20-19	Business Object	Optional	Multiple
Financial Report, page 20-65	Business Structure	Optional	Multiple

Name	Embedded Type	Requirement	Instance
Preference	Granular Entity	Optional	Multiple
Relationship	Granular Entity	Optional	Multiple
Classification	Granular Entity	Optional	Multiple
Credit Rating	Granular Entity	Optional	Multiple
Certification	Granular Entity	Optional	Multiple
Financial Profile	Granular Entity	Optional	Multiple
Contact Preference	Granular Entity	Optional	Multiple
Organization Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
organization_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE

---

<b>Attribute</b>	<b>Data Type</b>
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
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

---

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
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 to 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)

---

<b>Attribute</b>	<b>Data Type</b>
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 to credit_score_commentary10	VARCHAR2(30)
failure_score_class	NUMBER
failure_score_incd_default	NUMBER
failure_score_age	NUMBER

Attribute	Data Type
failure_score_date	DATE
failure_score_commentary2 to failure_score_commentary10	VARCHAR2(30)
maximum_credit_recommendation	NUMBER
maximum_credit_currency_code	VARCHAR2(240)
displayed_duns_party_id	NUMBER
do_not_confuse_with	VARCHAR2(255)

## Related Topics

[Business Objects Overview](#), page 20-2

## Organization Contact Business Object

### Integration Features

This table shows the seeded integration features for the Organization Contact business object.

#### *Organization Contact Business Object Integration Features*

Type	Name	Code	Procedure
Object	Organization Contact Business Object API, page 24-96	HZ_ORG_CONTACT_BO	HZ_ORG_CONTACT_BO.create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### **Organization Contact Business Object Embedded Items**

Name	Embedded Type	Requirement	Instance
Contact Person Information, page 20-53	Business Structure	Mandatory	Single
Contact Role	Granular Entity	Optional	Multiple
Party Site, page 20-35	Business Object	Optional	Multiple
Phone, page 20-44	Business Object	Optional	Multiple
Telex, page 20-49	Business Object	Optional	Multiple
E-Mail, page 20-21	Business Object	Optional	Multiple
Web, page 20-51	Business Object	Optional	Multiple
SMS, page 20-46	Business Object	Optional	Multiple
Financial Report, page 20-65	Business Structure	Optional	Multiple
Contact Preference	Granular Entity	Optional	Multiple
Organization Source System Information		Optional	Multiple

## **Attributes**

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
org_contact_id	NUMBER
action_type	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
organization_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)

---

<b>Attribute</b>	<b>Data Type</b>
rank	VARCHAR2(30)
party_site_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
relationship_code	VARCHAR2(30)
relationship_type	VARCHAR2(30)
relationship_comments	VARCHAR2(240)
start_date	DATE
end_date	DATE
relationship_status	VARCHAR2(1)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Organization Customer Business Object

### Integration Features

This table shows the seeded integration features for the Organization Customer business object.

#### *Organization Customer Business Object Integration Features*

Type	Name	Code	Procedure
Object	Organization Customer Business Object API, page 24-108	HZ_ORG_CUST_BO	HZ_ORG_CUST_BO.create_object

Type	Name	Code	Procedure
Business Event			<ul style="list-style-type: none"> <li>• Create (orgCustBO.create)</li> <li>• Update (orgCustBO.update)</li> </ul>

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Organization Customer Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Organization, page 20-23	Business Object	Mandatory	Single
Customer Account, page 20-6	Business Object	Mandatory	Multiple

## Related Topics

[Business Objects Overview, page 20-2](#)

## Party Site Business Object

### Integration Features

This table shows the seeded integration features for the Party Site business object.

***Party Site Business Object Integration Features***

Type	Name	Code	Procedure
Object	Party Site Business Object API, page 24-111	HZ_PARTY_SITE_BO	HZ_PARTY_SITE_BO.create_object

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Party Site Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Location, page 20-70	Business Structure	Mandatory	Single
Extensible Attributes – Party Site	Granular Entity	Optional	Multiple
Party Site Use	Granular Entity	Optional	Multiple
Phone, page 20-44	Business Object	Optional	Multiple
Telex, page 20-49	Business Object	Optional	Multiple
E-Mail, page 20-21	Business Object	Optional	Multiple
Web, page 20-51	Business Object	Optional	Multiple
Contact Preference	Granular Entity	Optional	Multiple
Party Site Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
party_site_id	NUMBER
action_type	VARCHAR2(30)

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
party_site_number	VARCHAR2(30)
mailstop	VARCHAR2(60)
identifying_address_flag	VARCHAR2(1)
party_site_name	VARCHAR2(240)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
language	VARCHAR2(4)

Attribute	Data Type
addressee	VARCHAR2(150)
global_location_number	VARCHAR2(40)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Person Business Object

### Integration Features

This table shows the seeded integration features for the Person business object.

#### *Person Business Object Integration Features*

Type	Name	Code	Procedures
Object	Person Business Object API, page 24-121	HZ_PERSON_BO	HZ_PERSON_BO.create_object
Business Event			<ul style="list-style-type: none"> <li>• Create (personBO.create)</li> <li>• Update (personBO.update)</li> </ul>

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### *Person Business Object Embedded Items*

Name	Embedded Type	Requirement	Instance
Extensible Attributes	Granular Entity	Optional	Multiple

Name	Embedded Type	Requirement	Instance
Party Site, page 20-35	Business Object	Optional	Multiple
Phone, page 20-44	Business Object	Optional	Multiple
E-Mail, page 20-21	Business Object	Optional	Multiple
Web, page 20-51	Business Object	Optional	Multiple
SMS, page 20-46	Business Object	Optional	Multiple
Employment History, page 20-65	Business Structure	Optional	Multiple
Preference	Granular Entity	Optional	Multiple
Relationship	Granular Entity	Optional	Multiple
Classification	Granular Entity	Optional	Multiple
Language	Granular Entity	Optional	Multiple
Education	Granular Entity	Optional	Multiple
Citizenship	Granular Entity	Optional	Multiple
Interest	Granular Entity	Optional	Multiple
Certification	Granular Entity	Optional	Multiple
Financial Profile	Granular Entity	Optional	Multiple
Contact Preference	Granular Entity	Optional	Multiple
Party Usage	Granular Entity	Optional	Multiple
Person Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the

documentation for the corresponding business objects. This table lists the root business object level attributes.

<b>Attribute</b>	<b>Data Type</b>
person_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
person_iden_type	VARCHAR2(30)
person_identifier	VARCHAR2(60)
date_of_birth	DATE
place_of_birth	VARCHAR2(60)
date_of_death	DATE
deceased_flag	VARCHAR2(1)
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)
internal_flag	VARCHAR2(2)

---

## Related Topics

[Business Objects Overview, page 20-2](#)

# Person Customer Business Object

## Integration Features

This table shows the seeded integration features for the Person Customer business object.

**Person Customer Business Object Integration Features**

Type	Name	Code	Procedures
Object	Person Customer Business Object API, page 24-136	HZ_PERSON_CUST_BO	HZ_PERSON_CUST_BO.create_object
Business Event			<ul style="list-style-type: none"><li>• Create (personCustBO.create)</li><li>• Update (personCustBO.update)</li></ul>

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

**Person Customer Business Object Embedded Items**

Name	Embedded Type	Requirement	Instance
Person, page 20-38	Business Object	Mandatory	Single
Customer Account, page 20-6	Business Object	Mandatory	Multiple

## Related Topics

[Business Objects Overview, page 20-2](#)

# Phone Business Object

## Integration Features

This table shows the seeded integration features for the Phone business object.

***Phone Business Object Integration Features***

Type	Name	Code	Procedure
Object	Phone Business Object API, page 24-139	HZ_PHONE_CP_BO	HZ_PHONE_CP_BO.create_object

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Phone Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
phone_id	NUMBER

---

<b>Attribute</b>	<b>Data Type</b>
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
phone_calling_calendar	VARCHAR2(30)

---

Attribute	Data Type
last_contact_dt_time	DATE
timezone_id	NUMBER
phone_area_code	VARCHAR2(10)
phone_country_code	VARCHAR2(10)
phone_number	VARCHAR2(40)
phone_extension	VARCHAR2(20)
phone_line_type	VARCHAR2(30)
raw_phone_number	VARCHAR2(60)

## Related Topics

[Business Objects Overview, page 20-2](#)

## SMS Business Object

### Integration Features

This table shows the seeded integration features for the SMS business object.

#### *SMS Business Object Integration Features*

Type	Name	Code	Procedure
Object	SMS Business Object API, page 24-148	HZ_SMS_CP_BO	HZ_SMS_CP_BO.create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

**SMS Business Object Embedded Items**

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

**Attributes**

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
sms_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)

<b>Attribute</b>	<b>Data Type</b>
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
phone_calling_calendar	VARCHAR2(30)
last_contact_dt_time	DATE
timezone_id	NUMBER
phone_area_code	VARCHAR2(10)
phone_country_code	VARCHAR2(10)
phone_number	VARCHAR2(40)
phone_extension	VARCHAR2(20)
phone_line_type	VARCHAR2(30)
raw_phone_number	VARCHAR2(60)

## Related Topics

[Business Objects Overview, page 20-2](#)

# Telex Business Object

## Integration Features

This table shows the seeded integration features for the Telex business object.

***Telex Business Object Integration Features***

Type	Name	Code	Procedure
Object	Telex Business Object API, page 24-158	HZ_TELEX_CP_BO	HZ_TELEX_CP_BO.create_object

## Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

***Telex Business Object Embedded Items***

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

## Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

Attribute	Data Type
telex_id	NUMBER

---

<b>Attribute</b>	<b>Data Type</b>
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
telex_number	VARCHAR2(50)

---

## Related Topics

Business Objects Overview, page 20-2

## Web Business Object

### Integration Features

This table shows the seeded integration features for the Web business object.

#### *Web Business Object Integration Features*

Type	Name	Code	Procedure
Object	Web Business Object API, page 24-166	HZ_WEB_CP_BO	HZ_WEB_CP_BO.create_object

### Embedded Items

This table shows the items embedded within the business object, as well as the defined rules for embedded items.

#### *Web Business Object Embedded Items*

Name	Embedded Type	Requirement	Instance
Contact Preference	Granular Entity	Optional	Multiple
Contact Point Source System Information		Optional	Multiple

### Attributes

The business object has attributes that describe the object at the root level, as well as attributes of embedded items. For the embedded business object attributes, see the documentation for the corresponding business objects. This table lists the root business object level attributes.

---

<b>Attribute</b>	<b>Data Type</b>
web_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)

---

Attribute	Data Type
web_type	VARCHAR2(60)
URL	VARCHAR2(2000)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Contact Person Information Business Structure

### Embedded Items

This table shows the items embedded within the business structure, as well as the defined rules for embedded items.

***Contact Person Information Business Structure Embedded Items***

Name	Embedded Type	Requirement	Instance
Extensible Attributes	Granular Entity	Optional	Multiple

### Attributes

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

Attribute	Data Type
person_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
status	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
person_pre_name_adjunct	VARCHAR2(30)
person_first_name	VARCHAR2(150)
person_middle_name	VARCHAR2(60)
person_last_name	VARCHAR2(150)

---

<b>Attribute</b>	<b>Data Type</b>
person_name_suffix	VARCHAR2(30)
person_title	VARCHAR2(60)
person_academic_title	VARCHAR2(30)
person_previous_last_name	VARCHAR2(150)
person_initials	VARCHAR2(6)
known_as	VARCHAR2(240)
known_as2	VARCHAR2(240)
known_as3	VARCHAR2(240)
known_as4	VARCHAR2(240)
known_as5	VARCHAR2(240)
person_name_phonetic	VARCHAR2(320)
person_first_name_phonetic	VARCHAR2(60)
person_last_name_phonetic	VARCHAR2(60)
middle_name_phonetic	VARCHAR2(60)
tax_reference	VARCHAR2(50)
jgzz_fiscal_code	VARCHAR2(20)
person_iden_type	VARCHAR2(30)
person_identifier	VARCHAR2(60)
date_of_birth	DATE
place_of_birth	VARCHAR2(60)

<b>Attribute</b>	<b>Data Type</b>
date_of_death	DATE
deceased_flag	VARCHAR2(1)
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)
internal_flag	VARCHAR2(2)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Customer Account Site Use Business Structure

### Integration Features

This table shows the seeded integration features for the Customer Account Site Use business structure.

### ***Customer Account Site Use Business Structure Integration Features***

Type	Name	Code	Procedure
Object	Customer Account Site Use Business Structure Attributes, page 24-185	HZ_CUST_SITE_USE_BO	HZ_ACCT_SITE_USE_BO. create_object

### **Embedded Items**

This table shows the items embedded within the business structure, as well as the defined rules for embedded items.

### ***Customer Account Site Use Business Structure Embedded Items***

Name	Embedded Type	Requirement	Instance
Customer Profile, page 20-61	Business Structure	Optional	Single
Bank Account Use	Entity (not TCA)	Optional	Multiple
Payment Method	Entity (not TCA)	Optional	Single
Account Site Use Source System Information		Optional	Multiple

### **Attributes**

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

Attribute	Data Type
site_use_id	NUMBER
action_type	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_site_id	NUMBER
site_use_code	VARCHAR2(30)
primary_flag	VARCHAR2(1)
location	VARCHAR2(40)
bill_to_site_use_id	NUMBER
sic_code	VARCHAR2(30)
payment_term_id	NUMBER
gsa_indicator	VARCHAR2(1)
ship_partial	VARCHAR2(1)
ship_via	VARCHAR2(30)

---

---

<b>Attribute</b>	<b>Data Type</b>
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 to attribute25	VARCHAR2(150)
tax_reference	VARCHAR2(50)
sort_priority	NUMBER
tax_code	VARCHAR2(50)
demand_class_code	VARCHAR2(30)
tax_header_level_flag	VARCHAR2(1)
tax_rounding_rule	VARCHAR2(30)
global_attribute1 to 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

---

<b>Attribute</b>	<b>Data Type</b>
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)
pricing_event	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

Attribute	Data Type
gl_id_factor	NUMBER
tax_classification	VARCHAR2(30)
org_id	NUMBER

## Related Topics

[Business Objects Overview, page 20-2](#)

## Customer Profile Business Structure

### Integration Features

This table shows the seeded integration features for the Customer Profile business structure.

#### *Customer Profile Business Structure Integration Features*

Type	Name	Code	Procedure
Object	Customer Profile Business Structure Attributes, page 24-200	HZ_CUSTOMER_PROFILE_E_BO	HZ_CUSTOMER_PROFILE_BO.create_object

### Embedded Items

This table shows the items embedded within the business structure, as well as the defined rules for embedded items.

#### *Customer Profile Business Structure Embedded Items*

Name	Embedded Type	Requirement	Instance
Customer Profile Amount	Granular Entity	Optional	Multiple

## Attributes

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

Attribute	Data Type
cust_acct_profile_id	NUMBER
action_type	VARCHAR2(30)
last_update_date	DATE
common_obj_id	VARCHAR2(255)
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
collector_id	NUMBER
credit_analyst_id	NUMBER
credit_checking	VARCHAR2(1)
next_credit_review_date	DATE
tolerance	NUMBER
discount_terms	VARCHAR2(1)

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

<b>Attribute</b>	<b>Data Type</b>
attribute1 to 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 to jgzz_attribute15	VARCHAR2(150)
global_attribute1 to global_attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
cons_inv_flag	VARCHAR2(1)
cons_inv_type	VARCHAR2(30)
autocash_hierarchy_id_for_addr	NUMBER
lockbox_matching_option	VARCHAR2(30)
review_cycle	VARCHAR2(30)
last_credit_review_date	DATE
credit_classification	VARCHAR2(30)
cons_bill_level	VARCHAR2(30)
late_charge_calculation_trx	VARCHAR2(30)
credit_items_flag	VARCHAR2(1)

Attribute	Data Type
disputed_transactions_flag	VARCHAR2(1)
late_charge_type	VARCHAR2(30)
late_charge_term_id	NUMBER
interest_calculation_period	VARCHAR2(30)
hold_charged_invoices_flag	VARCHAR2(1)
message_text_id	NUMBER
multiple_interest_rates_flag	VARCHAR2(1)
charge_begin_date	DATE

## Related Topics

[Business Objects Overview, page 20-2](#)

## Employment History Business Structure

### Integration Features

This table shows the seeded integration features for the Employment History business structure.

#### ***Employment History Business Structure Integration Features***

Type	Name	Code	Procedure
Object	Employment History Business Structure Attributes, page 24-220	HZ_EMPLOY_HIST_BO	HZ_EMPLOY_HIST_BO.create_object

### Embedded Items

This table shows the items embedded within the business structure, as well as the

defined rules for embedded items.

#### ***Employment History Business Structure Embedded Items***

Name	Embedded Type	Requirement	Instance
Work Class	Granular Entity	Optional	Multiple

## **Attributes**

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

Attribute	Data Type
employment_history_id	NUMBER
action_type	VARCHAR2(30)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_id	NUMBER
begin_date	DATE

<b>Attribute</b>	<b>Data Type</b>
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_flag	VARCHAR2(1)
tenure_code	VARCHAR2(30)
fraction_of_tenure	NUMBER
comments	VARCHAR2(2000)

## Related Topics

Business Objects Overview, page 20-2

## Financial Report Business Structure

### Integration Features

This table shows the seeded integration features for the Financial Report Information business structure.

*Financial Report Information Business Structure Integration Features*

Type	Name	Code	Procedure
API	Financial Report Business Structure Attributes, page 24-228	HZ_FINANCIAL_BO	HZ_FINANCIAL_BO.create_object

### Embedded Items

This table shows the items embedded within the business structure, as well as the defined rules for embedded items.

*Financial Report Business Structure Embedded Items*

Name	Embedded Type	Requirement	Instance
Financial Number	Granular Entity	Mandatory	Multiple

### Attributes

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

---

<b>Attribute</b>	<b>Data Type</b>
financial_report_id	NUMBER
action_type	VARCHAR2(30)
common_obj_id	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
organization_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
requiring_authority	VARCHAR2(60)
audit_ind	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
consolidated_ind	VARCHAR2(30)
estimated_ind	VARCHAR2(30)
fiscal_ind	VARCHAR2(30)
final_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)

## Related Topics

[Business Objects Overview, page 20-2](#)

## Location Business Structure

### Integration Features

This table shows the seeded integration features for the Location business structure.

#### ***Location Business Structure Integration Features***

Type	Name	Code	Procedure
Object	Location Business Structure Attributes, page 24-237	HZ_LOCATION_BO	HZ_LOCATION_BO.create_object

#### **Embedded Items**

This table shows the items embedded within the business structure, as well as the defined rules for embedded items.

#### ***Location Business Structure Embedded Items***

Name	Embedded Type	Requirement	Instance
Extensible Attributes	Granular Entity	Optional	Multiple
Location Source System Information		Optional	Multiple

#### **Attributes**

The business structure has attributes that describe the structure at the root level, as well as attributes of embedded items. For the embedded business structure attributes, see the documentation for the corresponding business structures. This table lists the root business structure level attributes.

Attribute	Data Type
location_id	NUMBER
action_type	VARCHAR2(30)
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
common_obj_id	VARCHAR2(255)

---

<b>Attribute</b>	<b>Data Type</b>
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	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)

---

---

<b>Attribute</b>	<b>Data Type</b>
address_lines_phonetic	VARCHAR2(560)
postal_plus4_code	VARCHAR2(10)
position	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)
loc_hierarchy_id	NUMBER
sales_tax_geocode	VARCHAR2(30)
sales_tax_inside_city_limits	VARCHAR2(30)
fa_location_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
timezone_id	NUMBER
delivery_point_code	VARCHAR2(50)
geometry_status_code	VARCHAR2(30)
geometry	mdsys.sdo_geometry

---

## **Related Topics**

Business Objects Overview, page 20-2

# 21

---

## Business Objects V2

This chapter covers the following topics:

- Business Objects V2 Overview
- Customer Account V2 Business Object
- Customer Account Site Use V2 Business Object
- Customer Account Site V2 Business Object
- Organization Customer V2 Business Object
- Person Customer V2 Business Object

### Business Objects V2 Overview

The Trading Community Architecture data model allows multiple payment methods for one account or account site use. The version 2 APIs are specifically created for the one-to-many mapping. To maintain backward compatibility, the existing APIs are also supported.

### Customer Account V2 Business Object

#### Integration Features

This table shows the seeded integration features for the Customer Account V2 business object.

Type	Name	Code	Procedure
Object	Customer Account V2 Business Object	HZ_CUST_ACCT_V2_BO	HZ_CUST_ACCT_V2_BO.create_object

## **Customer Account Site Use V2 Business Object**

### **Integration Features**

This table shows the seeded integration features for the Customer Account Site Use V2 business object.

Type	Name	Code	Procedure
Object	Customer Account Site Use V2 Business Object	HZ_CUST_SITE_USE_V2_BO	HZ_CUST_SITE_USE_V2_BO.create_object

## **Customer Account Site V2 Business Object**

### **Integration Features**

This table shows the seeded integration features for the Customer Account Site V2 business object.

Type	Name	Code	Procedure
Object	Customer Account Site V2 Business Object	HZ_CUST_ACCT_SITE_V2_BO	HZ_CUST_ACCT_SITE_V2_BO.create_object

## **Organization Customer V2 Business Object**

### **Integration Features**

This table shows the seeded integration features for the Organization Customer V2 business object.

Type	Name	Code	Procedure

---

Object	Organization Customer V2 Business Object	HZ_ORG_CUST_V2_ BO	HZ_ORG_CUST_V2_ BO.create_object
--------	--	-----------------------	-------------------------------------

---

## Person Customer V2 Business Object

### Integration Features

This table shows the seeded integration features for the Person Customer V2 business object.

---

Type	Name	Code	Procedure
Object	Person Customer V2 Business Object	HZ_PERSON_CUST_ V2_BO	HZ_PERSON_CUST_ V2_BO.create_object

---



---

## Business Object API Features and Procedures

This chapter covers the following topics:

- Business Object APIs Overview
- Seeded Business Object APIs
- Business Object API Features
- Create Business Object API Procedures
- Update Business Object API Procedures
- Save Business Object API Procedures
- Get Business Object API Procedures

### Business Object APIs Overview

Use the public business object APIs to perform operations on the seeded business objects. See: Business Objects Overview, page 20-2. Because business objects logically encapsulate specific areas of the Oracle Trading Community Architecture (TCA) data model, these types of business object API procedures let you directly manipulate TCA data:

- **Create:** Creates an instance of the business object in the database.
- **Update:** Updates a specific business object according to provided values.
- **Save:** Either creates or updates a business object, depending if the passed identification information matches an existing business object.
- **Get:** Extracts and returns data for either a single, complete business object, or all objects affected by a specific business event.

Business objects can contain a hierarchy of embedded business objects, business

structures, and granular entities. Even though you use business object APIs at the object level, that initial API call triggers subsequent business object or granular API calls down the hierarchy. For example, see: *Granular Versus Business Object APIs*, page 1-3.

## Related Topics

[Seeded Business Object APIs](#), page 22-2

[Business Object API Features](#), page 22-5

## Seeded Business Object APIs

Business object APIs are provided for every seeded business object. For details on each API, including description, PL/SQL procedure, parameters, and validations, see: *Oracle Integration Repository*. For information about what each API does with the corresponding business object's attributes, see: *Business Object API Attributes Information Overview*, page 24-2.

The seeded business object APIs are:

- Customer Account Business Object API
- Customer Account Contact Business Object API
- Customer Account Site Business Object API
- EDI Business Object API
- EFT Business Object API
- E-Mail Business Object API
- Organization Business Object API
- Organization Contact Business Object API
- Organization Customer Business Object API
- Party Site Business Object API
- Person Business Object API
- Person Customer Business Object API
- Phone Business Object API
- Source System Management Business Object API

- SMS Business Object API
- Telex Business Object API
- Web Business Object API

The seeded APIs have all four types of procedures: Create, Update, Save, and Get. See: Business Object API Features, page 22-5.

**Note:** Unlike the other seeded APIs, the Source System Management Business Object API does not create, update, save, or get a specific seeded business object. You use this API to create and maintain source system mappings for any business object. See: Source System Management Business Object API, page 22-3.

## Related Topics

[Business Object APIs Overview](#), page 22-1

## Source System Management Business Object API

Source System Management (SSM) maintains mappings, or references, of records between source systems, for example between Oracle Trading Community Architecture and a legacy system, or between a data hub and a spoke system. See: Source System Management Overview, *Oracle Trading Community Architecture Administration Guide*.

The Source System Management Business Object API maps entire business objects, not specific granular entities, to their source systems. This table shows the root TCA entities for each business object or business structure, representing the transition link between entity SSM and business object SSM.

Business Object or Structure	TCA Entity Table
Organization	HZ_PARTIES
Person	HZ_PARTIES
Party Site	HZ_PARTY_SITES
Organization Contact	HZ_ORG_CONTACTS
Customer Account	HZ_CUST_ACCOUNTS
Customer Account Site	HZ_CUST_ACCT_SITES_ALL

<b>Business Object or Structure</b>	<b>TCA Entity Table</b>
Customer Account Contact	HZ_CUST_ACCOUNT_ROLES
Phone	HZ_CONTACT_POINTS
Telex	HZ_CONTACT_POINTS
E-Mail	HZ_CONTACT_POINTS
Web	HZ_CONTACT_POINTS
SMS	HZ_CONTACT_POINTS
EDI	HZ_CONTACT_POINTS
EFT	HZ_CONTACT_POINTS
Location	HZ_LOCATIONS
Person Contact Information	HZ_PARTIES

**Note:** The Organization Customer and Person Customer business objects are not mapped to any TCA entities. Business object SSM support is not available for these business objects.

The Source System Management Business Object API procedures are:

- Create Business Object Source System Reference
- Update Business Object Source System Reference
- Remap Internal Business Object Identifier

For details on these procedures, including description, PL/SQL procedure, parameters, and validations, see: Source System Management Business Object, *Oracle Integration Repository*.

## Related Topics

Seeded Business Object APIs, page 22-2

## Business Object API Features

All business object API procedures of the same type share common features and functionality. See:

- Create Business Object API Procedures, page 22-6
- Update Business Object API Procedures, page 22-9
- Save Business Object API Procedures, page 22-11
- Get Business Object API Procedures, page 22-13

Business object API procedures of all types also share common return status and error handling.

## Business Object API Return Status

The `x_return_status` output parameter provides the status of the API:

- **Success (FND\_API.G\_RET\_STS\_SUCCESS):** The business object API procedure was successful in its operation. The entire business object was created, updated, or retrieved, depending on the procedure, and all of the caller's expected operations occurred.
- **Error (FND\_API.G\_RET\_STS\_ERROR):** The business object API procedure did not accomplish its function. Relevant error messages are provided. Common errors include missing attributes or embedded entities.
- **Unexpected Error (FND\_API.G\_RET\_STS\_UNEXP\_ERROR):** The business object API procedure did not accomplish its function, due to errors that the API could not handle.

## Business Object API Error Handling

Errors can occur at either the business object API level, or at the level of the embedded business object or granular API calls. For example, a business object has an embedded granular entity. When you use an API to update that business object, that initial call triggers a call to the appropriate granular API for the embedded entity. You can get errors from either API call.

Errors from the initial business object API call are reported in the usual way. Examples of errors include not entering mandatory embedded entities, or trying to create a business object that already exists.

Errors from embedded API calls, however, are propagated to the top business object API, which then provides you additional messages regarding the embedded business

object, business structure, or granular entity that is the cause of the error. Because a business object can have multiple embedded items, with many levels, these messages help you determine the exact cause of the errors.

For example, you are trying to create a business object, which contains two embedded granular entities. One embedded entity cannot be created because a mandatory value is not passed, so an error message is provided specifying the missing value at the granular level. Another error message is provided at the business object level indicating in which embedded entity the attribute is missing.

## Related Topics

[API Overview, page 1-2](#)

[Business Object APIs Overview, page 22-1](#)

## Create Business Object API Procedures

Create business object API procedures create an instance of the particular business object in the database, returns the business object identifier to the caller, and provides information communicating the success of the operation. Create procedures are provided for every seeded business object. For details on specific procedures, see: *Oracle Integration Repository*.

If errors occur with the Create business object procedure call, then all parts of the call fail, including the API calls for embedded items. You cannot partially create a business object.

Any error that occurs in creating embedded items is propagated to the top business object level, and the creation of the entire object fails. See: [Business Object API Error Handling, page 22-5](#).

## Validations

- If the p\_validate\_bo\_flag parameter is set to TRUE, which it is by default, then you must provide all mandatory embedded items in the business object that you want to create. If you do not, then the Create procedure fails. See: [Rules for Embedded Items, page 20-4](#).
- Some business objects cannot exist on their own, for example Organization Contact, Party Site, and Customer Account Site. To create such business objects, by calling for example the Create Organization Contact Business Object API procedure, you must pass either a TCA identifier or a combination of source system and source system reference, or both, to identify the parent business object. This table describes the Create procedure behavior depending on the passed information.

**Create Procedure Behavior Based on Passed Parent Identification Information**

TCA ID Passed	TCA ID Valid and of Correct Type	Source System and Source System Reference Passed	Source System and Source System Reference Valid and of Correct Type	TCA ID and Source System Information Map to Each Other and of Correct Type	API Procedure Behavior
No		Yes	Yes		Creates business object
No		Yes	No		Fails with invalid identification error
Yes	Yes	No			Creates business object
Yes	No				Fails with invalid identification error
Yes	Yes	Yes	Yes	Yes	Creates business object
Yes	Yes	Yes	Yes	No	Fails with invalid identification error
Yes	Yes	Yes	No	No	Fails with invalid identification error

- Optionally provide source system information for the business object that you want to create, to verify that the object does not already exist. If the object exists, then the Create procedure fails. This table describes the Create procedure behavior depending on the passed information.

TCA ID Passed	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	API Procedure Behavior
No	Yes	Yes	Fails with error specifying that the business object already exists
No	Yes	No	Creates business object and source system mapping
No	No		Creates business object
Yes			Fails with invalid identification error

## Embedded Business Objects and Structures

In most cases, when you call a Create business object API procedure, the business object's embedded business objects and structures are also created, with their own Create business object API procedures called in a nested manner. In the single case where an existing Person business object can be used as an Organization Contact, when you use the Create Organization Contact Business Object API procedure, the Save Person Business Object API procedure is called to check that the person already exists. See: Save Business Object API Procedures, page 22-11.

## Embedded Granular Entities

When you call a Create business object API procedure, corresponding granular APIs are called for embedded granular entities, depending on information, if any, that you pass to identify existing entities.

- If you do not pass identification information, or the identifier does not match existing entities, then the granular Create procedure is called to create the entity.
- If the entity exists, then the granular Update procedure is called. This can happen for entities, such as locations, that are shared among multiple business objects.

## Related Topics

[Business Object API Features, page 22-5](#)

## Update Business Object API Procedures

Update business object API procedures modify a particular existing business object according to the provided values, returns the updated portions of the business object to the caller, and provides information communicating the success of the operation.

Update procedures are provided for every seeded business object. For details on specific procedures, see: *Oracle Integration Repository*.

If errors occur with the Update business object procedure call, then all parts of the call fail, including the API calls for embedded items. You cannot partially update a business object.

Any error that occurs in creating or updating embedded items is propagated to the top business object level, and the update of the entire object fails. See: Business Object API Error Handling, page 22-5.

## Validations

You must pass either a TCA identifier or a combination of source system and source system reference, or both, to identify the existing business object to update. This table describes the Update procedure behavior depending on the passed information.

TCA ID Passed	TCA ID Valid	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	TCA ID and Source System Information Map to Each Other	API Procedure Behavior
No	Yes	Yes	Yes		Updates business object
No	Yes	Yes	No		Fails with invalid identification error
Yes	Yes	Yes	No		Updates business object
Yes	No	Yes	Yes		Fails with invalid identification error

TCA ID Passed	TCA ID Valid	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	TCA ID and Source System Information Map to Each Other	API Procedure Behavior
Yes	Yes	Yes	Yes	Yes	Updates business object
Yes	Yes	Yes	Yes	No	Fails with invalid identification error
Yes	Yes	Yes	No		Fails with invalid identification error

## Embedded Business Objects and Structures

When you call an Update business object API procedure, the business object's embedded business objects and structures are either updated or created, with their own Save business object API procedures called in a nested manner. The Save procedure is called because:

- Embedded items can be shared, for example Phone business object between Person business objects.
- The Save procedure checks if items already exist.

See: Save Business Object API Procedures, page 22-11.

## Embedded Granular Entities

When you call an Update business object API procedure, corresponding granular APIs are called to create or update embedded granular entities, based on information, if any, that you pass to identify existing entities.

If you pass a valid TCA identifier for the entity, then the entity is updated. If the ID is invalid, then the granular API call returns an error. If you do not pass any TCA identifier, then one of two checks occur:

- If Source System Management (SSM) is supported for the granular entity, then the granular API call checks if you provided a source system and source system reference combination. If source system information is:

- Valid, then the entity is updated.
  - Provided but not matching an existing entity, then the entity is created.
  - Invalid, then the granular API call returns an error.
  - Not passed at all, then the entity is created.
- If SSM is not supported for the entity, the granular API call attempts to identify the entity that the caller is referring to, using a logical key. If the provided logical key attributes uniquely match an existing entity in TCA, then that entity is updated. For that reason, logical key attributes for an entity cannot be updated without the TCA identifier.

If the provided logical key attributes do not uniquely match an existing entity in TCA and no TCA identifier is provided, then the entity is created.

## Related Topics

[Business Object API Features](#), page 22-5

[Business Object APIs Overview](#), page 22-1

## Save Business Object API Procedures

Save business object API procedures are essentially logical wrappings of both Create and Update API procedures for the same business object. For details on specific Save procedures, see: *Oracle Integration Repository*.

The Save procedure calls either the Create or Update procedure:

- **Create:** If you do not pass identification information, or if the business object does not exist. See: [Create Business Object API Procedures](#), page 22-6.
- **Update:** If you pass valid information to identify an existing business object. See: [Update Business Object API Procedures](#), page 22-9.

## Validations

You must pass either a TCA identifier or a combination of source system and source system reference, or both, to identify the existing business object to update. This table describes the Save procedure behavior depending on the passed information.

TCA ID Passed	TCA ID Valid	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	TCA ID and Source System Information Map to Each Other	API Procedure Behavior
No	No				Creates business object
No	Yes		Yes		Updates business object
No	Yes		No		Fails with invalid identification error
Yes	Yes		No		Updates business object
Yes	No				Fails with invalid identification error
Yes	Yes	Yes	Yes	Yes	Updates business object
Yes	Yes	Yes	Yes	No	Fails with invalid identification error
Yes	Yes	Yes	No	No	Fails with invalid identification error

## Related Topics

[Business Object API Features, page 22-5](#)

[Business Object APIs Overview, page 22-1](#)

## Get Business Object API Procedures

Two types of Get business object API procedures are available:

- Event Independent, page 22-13, for all seeded business objects.
- Event Dependent, page 22-14, for the seeded Organization Customer, Person Customer, Organization, and Person business objects.

For details on specific procedures, see: *Oracle Integration Repository*.

### Related Topics

[Business Object API Features](#), page 22-5

[Business Object APIs Overview](#), page 22-1

### Event Independent Get Procedures

Provided for each of the seeded business objects, these procedures retrieve and return business objects, recursively traversing down the business object's hierarchy, gathering the embedded objects, structures, and entities, to provide a complete, packaged business object. The returned object is a wrapping around data structures representing all of the business object's embedded items.

#### Validations

You must pass either a TCA identifier or a combination of source system and source system reference, or both, to identify the existing business object to extract. This table describes the Get procedure behavior depending on the passed information.

TCA ID Passed	TCA ID Valid	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	TCA ID and Source System Information Map to Each Other	API Procedure Behavior
No	Yes		Yes		Gets business object
No	Yes		No		Fails with invalid identification error

TCA ID Passed	TCA ID Valid	Source System and Source System Reference Passed	Passed Source System Information Matches Existing Mapping	TCA ID and Source System Information Map to Each Other	API Procedure Behavior
Yes	Yes	No			Gets business object
Yes	No				Fails with invalid identification error
Yes	Yes	Yes	Yes	Yes	Gets business object
Yes	Yes	Yes	Yes	No	Fails with invalid identification error
Yes	Yes	Yes	No		Fails with invalid identification error

## Related Topics

Get Business Object API Procedures, page 22-13

## Event Dependent Get Procedures

Provided for the seeded Organization Customer, Person Customer, Organization, and Person business objects, these procedures are affiliated with each seeded Create or Update business event for these objects. See: <[link to business events overview](#)>.

For each event, the corresponding Get procedure returns, to the event subscriber in one API call, all the business objects that the single business object affects:

- **Create Business Event:** The entire new business object is returned.
- **Update Business Event:** The entire updated business object is returned, as well as details about what and how specific portions of the object were updated.

For example, when an Organization business object is created, a corresponding Create

Organization business event is raised to event subscribers, and the Get Created Organization Business Object API procedure returns the new Organization business object to the subscribers.

**Note:** The Event Dependent procedures are specifically used only in the context of business object business events. Do not use them otherwise.

These procedures allow systems to retrieve business objects as soon as the business object data is created or updated. Subscribing parties must be able to act on a business event by receiving the created or updated business objects.

## Event Dependent Extraction Process

The event dependent business object extraction process involves different types of Event Dependent procedures:

- Get Business Object IDs
  - Get Updated Business Object
  - Get Business Object Data in Bulk
1. Even though one business event can involve multiple business objects, the event passes a single numeric event identifier to the event subscriber. This ID is then passed to the corresponding **Get Business Object IDs procedure** and used as a key to access a tracking table, to determine the affected object data for the business event. Given the passed business event ID, the procedure extracts an array of business object IDs.

**Important:** The business event identifier passed to the Get Business Object IDs procedure must be valid for the business object. An error occurs if the passed ID is not found in the tracking table or is not valid for the business object.

Get Business Object IDs procedures are provided for the Create and Update business events for Organization Customer, Person Customer, Organization, and Person business objects. For example, if the business event is that an Organization business object is created or updated, then either the Get Created Organization Business Object ID or Get Updated Organization Business Object ID procedure is called, respectively.

2. With business object identifiers extracted by the Get Business Object IDs procedure, another procedure is called next to retrieve each of the business objects in the array of IDs. One procedure is called for each ID in the array, with the business event and object IDs passed in, depending on the type of business event.

- *Create Business Event*: The corresponding Event Independent Get procedure is called. See: Event Independent Get Procedures, page 22-13.
- *Update Business Event*: The corresponding **Get Updated Business Object procedure** is called to extract not only the business object data, but also indicators that define which pieces of data has changed. See: Get Updated Business Object Procedures, page 22-16.

An error occurs if you attempt to get an updated business object by passing a business event ID and business object ID, but the object was not updated as part of that event.

Alternatively, you can combine these two steps and use the **Get Business Object Data in Bulk procedure**. As essentially a wrapping of the Get Business Object IDs procedures and the appropriate procedures to retrieve business object data, the Get Business Object Data in Bulk:

1. Receives the single business event identifier from the business event and retrieves an array of business object IDs.
2. For each business object ID in the array, calls the appropriate Event Independent or Get Updated Business Object procedure and adds the created or updated business object to a table.
3. Returns the table of objects to the caller.

## Get Updated Business Object Procedures

Get Updated Business Object procedures are provided for the Update business events for Organization Customer, Person Customers, Organization, and Person business objects. See: Event Dependent Extraction Process, page 22-15.

These procedures extract the entire updated business object, for a specific Update business event. By returning an action type, the procedure also specifies how the business object's root level attributes, as well as the object's embedded items, have changed as a result of the Update business event. See: Business Object Attributes, page 20-2 and Business Object Embedded Items, page 20-3.

**Note:** The procedures provide action types for business objects, business structures, and embedded granular entities at any level of the top business object's hierarchy, except at the root business object level for Organization Customer and Person Customer business objects.

This table describes what each action type means for business objects or structures, at any level of the business object hierarchy, and embedded entities. If an action type indicates that the business object or structure has new or updated embedded items, as a result of the Update business event, you refer to the action type for that embedded item

to determine what happened.

Action Type	Meaning for Business Object or Structure	Meaning for Granular Entity
CHILD_UPDATED	<ul style="list-style-type: none"> <li>Business object or structure attributes were not updated.</li> <li>At least one embedded item was created or updated.</li> </ul>	Not used.
CREATED	<ul style="list-style-type: none"> <li>Business object or structure attributes were created.</li> <li>Any embedded items were created.</li> </ul>	Attributes of the embedded entity were created.
UPDATED	<ul style="list-style-type: none"> <li>Business object or structure attributes were updated.</li> <li>One or more embedded items might have been created or updated.</li> </ul>	Attributes of the embedded entity were updated.
UNCHANGED	No changes were made to business object or structure attributes, nor to any of the embedded items.	No changes were made to attributes of the embedded entity.

### Action Type Example

This example illustrates the action types that are returned based on the changes made as a result of a specific Update business event. For this example, an Organization business object with ID of 1200 has this hierarchy:

- Organization business object (ID: 1200)
  - Party Site business object (ID: 4445)
    - Location business structure (ID: 11)
  - Phone business object (ID: 1345)
  - Organization Contact business object (ID: 98)
    - Contact Person Information business structure (ID: 5)

### **Phone Business Object Added to Organization Contact Business Object**

If a Phone business object with ID 321 is added to the Organization Contact business object 98, within Organization business object 1200, then an Organization Updated business event is raised. The Get Updated Organization Business Object procedure would return Organization object 1200 with the following action types.

- Organization business object (ID: 1200, Action Type: CHILD\_UPDATED)
  - Party Site business object (ID: 4445, Action Type: UNCHANGED)
    - Location business structure (ID: 11, Action Type: UNCHANGED)
  - Phone business object (ID: 1345, Action Type: UNCHANGED)
  - Organization Contact business object (ID: 98, Action Type: CHILD\_UPDATED)
    - Contact Person Information business structure (ID: 5, Action Type: UNCHANGED)
  - Phone business object (ID: 321, Action Type: CREATED)

### **Organization Business Object Attributes Updated**

If root level attributes of Organization business object 1200 are updated, then an Organization Updated business event is raised. The Get Updated Organization Business Object procedure would return Organization Object 1200 with the following action types.

- Organization business object (ID: 1200, Action Type: UPDATED)
- Everything else (Action Type: Unchanged)

### **Location Business Structure Updated and Contact Preference Added**

If, within Organization business object 1200, the Location business structure 11 is updated, and Contact Preference entity 117 is added to Phone business object 321, then an Organization Updated business event is raised. The Get Updated Organization Business Object procedure would return Organization object 1200 with the following action types.

- Organization business object (ID: 1200, Action Type: CHILD\_UPDATED)
  - Party Site business object (ID: 4445, Action Type: CHILD\_UPDATED)
    - Location business structure (ID: 11, Action Type: UPDATED)
  - Phone business object (ID: 1345, Action Type: UNCHANGED)
  - Organization Contact business object (ID: 98, Action Type: CHILD\_UPDATED)
    - Contact Person Information business structure (ID: 5, Action Type: UNCHANGED)

UNCHANGED)

- Phone business object (ID: 321, Action Type: CHILD\_UPDATED)
  - Contact Preference entity (ID: 117, Action Type: CREATED)

## Related Topics

[Get Business Object API Procedures](#), page 22-13



---

## Business Object V2 API Procedures

This chapter covers the following topics:

- Customer Account V2 Business Object API
- Customer Account Site V2 Business Object API
- Organization Customer V2 Business Object API
- Person Customer V2 Business Object API

### Customer Account V2 Business Object API

#### Create Customer Account V2 Business Object API

```
PROCEDURE create_cust_acct_v2_bo(
    p_validate_bo_flag      IN      VARCHAR2 := fnd_api.g_true,
    p_cust_acct_v2_obj     IN      HZ_CUST_ACCT_V2_BO,
    p_created_by_module    IN      VARCHAR2,
    p_obj_source            IN      VARCHAR2 := null,
    p_return_obj_flag       IN      VARCHAR2 := fnd_api.g_true,
    x_return_status          OUT     VARCHAR2,
    x_messages               OUT     HZ_MESSAGE_OBJ_TBL,
    x_return_obj              OUT     HZ_CUST_ACCT_V2_BO,
    x_cust_acct_id           OUT     NUMBER,
    x_cust_acct_os            OUT     VARCHAR2,
    x_cust_acct_osr           OUT     VARCHAR2,
    px_parent_id              IN OUT  NUMBER,
    px_parent_os              IN OUT  VARCHAR2,
    px_parent_osr             IN OUT  VARCHAR2,
    px_parent_obj_type        IN OUT  VARCHAR2
);
```

## Update Customer Account V2 Business Object API

```
PROCEDURE update_cust_acct_v2_bo(
    p_cust_acct_v2_obj           IN      HZ_CUST_ACCT_V2_BO,
    p_created_by_module          IN      VARCHAR2,
    p_obj_source                 IN      VARCHAR2 := null,
    p_return_obj_flag            IN      VARCHAR2 := fnd_api.g_true,
    x_return_status               OUT     NOCOPY  VARCHAR2,
    x_messages                   OUT     NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_return_obj                 OUT     NOCOPY  HZ_CUST_ACCT_V2_BO,
    x_cust_acct_id               OUT     NOCOPY  NUMBER,
    x_cust_acct_os               OUT     NOCOPY  VARCHAR2,
    x_cust_acct_osr              OUT     NOCOPY  VARCHAR2
);
;
```

## Save Customer Account V2 Business Object API

```
PROCEDURE save_cust_acct_v2_bo(
    p_validate_bo_flag           IN      VARCHAR2 := fnd_api.g_true,
    p_cust_acct_v2_obj          IN      HZ_CUST_ACCT_V2_BO,
    p_created_by_module          IN      VARCHAR2,
    p_obj_source                 IN      VARCHAR2 := null,
    p_return_obj_flag            IN      VARCHAR2 := fnd_api.g_true,
    x_return_status               OUT     NOCOPY  VARCHAR2,
    x_messages                   OUT     NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_return_obj                 OUT     NOCOPY  HZ_CUST_ACCT_V2_BO,
    x_cust_acct_id               OUT     NOCOPY  NUMBER,
    x_cust_acct_os               OUT     NOCOPY  VARCHAR2,
    x_cust_acct_osr              OUT     NOCOPY  VARCHAR2,
    px_parent_id                 IN OUT    NOCOPY  NUMBER,
    px_parent_os                 IN OUT    NOCOPY  VARCHAR2,
    px_parent_osr                IN OUT    NOCOPY  VARCHAR2,
    px_parent_obj_type           IN OUT    NOCOPY  VARCHAR2
);
;
```

## Get Customer Account V2 Business Object API

```
PROCEDURE get_cust_acct_v2_bo(
    p_init_msg_list              IN      VARCHAR2 := fnd_api.g_false,
    p_cust_acct_id               IN      NUMBER,
    p_cust_acct_os               IN      VARCHAR2,
    p_cust_acct_osr              IN      VARCHAR2,
    x_cust_acct_v2_obj           OUT    NOCOPY  HZ_CUST_ACCT_V2_BO,
    x_return_status               OUT    NOCOPY  VARCHAR2,
    x_msg_count                  OUT    NOCOPY  NUMBER,
    x_msg_data                   OUT    NOCOPY  VARCHAR2
);
;
```

# Customer Account Site V2 Business Object API

## Create Customer Account Site V2 Business Object API

```
PROCEDURE create_cust_acct_site_v2_bo(
    p_validate_bo_flag           IN      VARCHAR2 := fnd_api.g_true,
    p_cust_acct_site_v2_obj     IN      HZ_CUST_ACCT_SITE_V2_BO,
    p_created_by_module         IN      VARCHAR2,
    p_obj_source                IN      VARCHAR2 := null,
    p_return_obj_flag           IN      VARCHAR2 := fnd_api.g_true,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_messages                  OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj                 OUT NOCOPY HZ_CUST_ACCT_SITE_V2_BO,
    x_cust_acct_site_id         OUT NOCOPY NUMBER,
    x_cust_acct_site_os          OUT NOCOPY VARCHAR2,
    x_cust_acct_site_osr        OUT NOCOPY VARCHAR2,
    px_parent_acct_id           IN OUT NOCOPY NUMBER,
    px_parent_acct_os            IN OUT NOCOPY VARCHAR2,
    px_parent_acct_osr          IN OUT NOCOPY VARCHAR2
);
```

## Update Customer Account Site V2 Business Object API

```
PROCEDURE update_cust_acct_site_v2_bo(
    p_cust_acct_site_v2_obj     IN      HZ_CUST_ACCT_SITE_V2_BO,
    p_created_by_module         IN      VARCHAR2,
    p_obj_source                IN      VARCHAR2 := null,
    p_return_obj_flag           IN      VARCHAR2 := fnd_api.g_true,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_messages                  OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj                 OUT NOCOPY HZ_CUST_ACCT_SITE_V2_BO,
    x_cust_acct_site_id         OUT NOCOPY NUMBER,
    x_cust_acct_site_os          OUT NOCOPY VARCHAR2,
    x_cust_acct_site_osr        OUT NOCOPY VARCHAR2
);
```

## Save Customer Account Site V2 Business Object API

```
PROCEDURE save_cust_acct_site_v2_bo(
    p_validate_bo_flag           IN      VARCHAR2 := fnd_api.g_true,
    p_cust_acct_site_v2_obj     IN      HZ_CUST_ACCT_SITE_V2_BO,
    p_created_by_module         IN      VARCHAR2,
    p_obj_source                IN      VARCHAR2 := null,
    p_return_obj_flag           IN      VARCHAR2 := fnd_api.g_true,
    x_return_status              OUT NOCOPY VARCHAR2,
    x_messages                  OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj                 OUT NOCOPY HZ_CUST_ACCT_SITE_V2_BO,
    x_cust_acct_site_id         OUT NOCOPY NUMBER,
    x_cust_acct_site_os          OUT NOCOPY VARCHAR2,
    x_cust_acct_site_osr        OUT NOCOPY VARCHAR2,
    px_parent_acct_id           IN OUT NOCOPY NUMBER,
    px_parent_acct_os            IN OUT NOCOPY VARCHAR2,
    px_parent_acct_osr          IN OUT NOCOPY VARCHAR2
);
```

## Get Customer Account Site V2 Business Object API

```
PROCEDURE get_cust_acct_site_v2_bo (
    p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
    p_cust_acct_site_id IN NUMBER,
    p_cust_acct_site_os IN VARCHAR2,
    p_cust_acct_site_osr IN VARCHAR2,
    x_cust_acct_site_v2_obj OUT NOCOPY HZ_CUST_ACCT_SITE_V2_BO,
    x_return_status OUT NOCOPY VARCHAR2,
    x_msg_count OUT NOCOPY NUMBER,
    x_msg_data OUT NOCOPY VARCHAR2
);
```

## Organization Customer V2 Business Object API

### Create Organization Customer V2 Business Object API

```
PROCEDURE create_org_cust_v2_bo(
    p_validate_bo_flag      IN          VARCHAR2 := fnd_api.g_true,
    p_org_cust_v2_obj       IN          HZ_org_cust_v2_bo,
    p_created_by_module    IN          VARCHAR2,
    p_obj_source            IN          VARCHAR2 := null,
    p_return_obj_flag       IN          VARCHAR2 := fnd_api.g_true,
    x_return_status          OUT NOCOPY VARCHAR2,
    x_messages              OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj             OUT NOCOPY HZ_org_cust_v2_bo,
    x_organization_id        OUT NOCOPY NUMBER
);
```

### Update Organization Customer V2 Business Object API

```
PROCEDURE update_org_cust_v2_bo(
    p_org_cust_v2_obj       IN          HZ_org_cust_v2_bo,
    p_created_by_module    IN          VARCHAR2,
    p_obj_source            IN          VARCHAR2 := null,
    p_return_obj_flag       IN          VARCHAR2 := fnd_api.g_true,
    x_return_status          OUT NOCOPY VARCHAR2,
    x_messages              OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj             OUT NOCOPY HZ_org_cust_v2_bo,
    x_organization_id        OUT NOCOPY NUMBER
);
```

### Save Organization Customer V2 Business Object API

```
PROCEDURE save_org_cust_v2_bo(
    p_validate_bo_flag      IN          VARCHAR2 := fnd_api.g_true,
    p_org_cust_v2_obj       IN          HZ_org_cust_v2_bo,
    p_created_by_module    IN          VARCHAR2,
    p_obj_source            IN          VARCHAR2 := null,
    p_return_obj_flag       IN          VARCHAR2 := fnd_api.g_true,
    x_return_status          OUT NOCOPY VARCHAR2,
    x_messages              OUT NOCOPY HZ_MESSAGE_OBJ_TBL,
    x_return_obj             OUT NOCOPY HZ_org_cust_v2_bo,
    x_organization_id        OUT NOCOPY NUMBER
);
```

## Get Organization Customer V2 Business Object API

```
PROCEDURE get_org_cust_v2_bo(
    p_organization_id           IN          NUMBER,
    p_organization_os           IN          VARCHAR2,
    p_organization_osr          IN          VARCHAR2,
    x_org_cust_v2_obj           OUT NOCOPY   HZ_ORG_CUST_V2_BO,
    x_return_status              OUT NOCOPY   VARCHAR2,
    x_messages                  OUT NOCOPY   HZ_MESSAGE_OBJ_TBL
) ;

PROCEDURE get_v2_org_custs_created(
    p_event_id                 IN          NUMBER,
    x_org_cust_v2_objs          OUT NOCOPY   HZ_ORG_CUST_V2_BO_TBL,
    x_return_status              OUT NOCOPY   VARCHAR2,
    x_messages                  OUT NOCOPY   HZ_MESSAGE_OBJ_TBL
) ;

PROCEDURE get_v2_org_custs_updated(
    p_event_id                 IN          NUMBER,
    x_org_cust_v2_objs          OUT NOCOPY   HZ_ORG_CUST_V2_BO_TBL,
    x_return_status              OUT NOCOPY   VARCHAR2,
    x_messages                  OUT NOCOPY   HZ_MESSAGE_OBJ_TBL
) ;

PROCEDURE get_v2_org_cust_updated(
    p_event_id                 IN          NUMBER,
    p_org_cust_id               IN          NUMBER,
    x_org_cust_v2_obj           OUT NOCOPY   HZ_ORG_CUST_V2_BO,
    x_return_status              OUT NOCOPY   VARCHAR2,
    x_messages                  OUT NOCOPY   HZ_MESSAGE_OBJ_TBL
) ;
```

## Person Customer V2 Business Object API

### Create Person Customer V2 Business Object API

```
PROCEDURE create_person_cust_v2_bo(
    p_validate_bo_flag          IN          VARCHAR2 := fnd_api.g_true,
    p_person_cust_v2_obj        IN          HZ_PERSON_CUST_V2_BO,
    p_created_by_module         IN          VARCHAR2,
    p_obj_source                IN          VARCHAR2 := null,
    p_return_obj_flag           IN          VARCHAR2 := fnd_api.g_true,
    x_return_status              OUT NOCOPY   VARCHAR2,
    x_messages                  OUT NOCOPY   HZ_MESSAGE_OBJ_TBL,
    x_return_obj                OUT NOCOPY   HZ_PERSON_CUST_V2_BO,
    x_person_id                 OUT NOCOPY   NUMBER
) ;
```

## Update Person Customer V2 Business Object API

```
PROCEDURE update_person_cust_v2_bo(
    p_person_cust_v2_obj      IN          HZ_PERSON_CUST_V2_BO,
    p_created_by_module       IN          VARCHAR2,
    p_obj_source              IN          VARCHAR2 := null,
    p_return_obj_flag         IN          VARCHAR2 := fnd_api.g_true,
    x_return_status            OUT NOCOPY  VARCHAR2,
    x_messages                OUT NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_return_obj               OUT NOCOPY  HZ_PERSON_CUST_V2_BO,
    x_person_id                OUT NOCOPY  NUMBER
);
```

## Save Person Customer V2 Business Object API

```
PROCEDURE save_person_cust_v2_bo(
    p_validate_bo_flag        IN          VARCHAR2 := fnd_api.g_true,
    p_person_cust_v2_obj      IN          HZ_PERSON_CUST_V2_BO,
    p_created_by_module       IN          VARCHAR2,
    p_obj_source              IN          VARCHAR2 := null,
    p_return_obj_flag         IN          VARCHAR2 := fnd_api.g_true,
    x_return_status            OUT NOCOPY  VARCHAR2,
    x_messages                OUT NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_return_obj               OUT NOCOPY  HZ_PERSON_CUST_V2_BO,
    x_person_id                OUT NOCOPY  NUMBER
);
```

## Get Person Customer V2 Business Object API

```
PROCEDURE get_person_cust_v2_bo(
    p_init_msg_list           IN          VARCHAR2 := fnd_api.g_false,
    p_person_id                IN          NUMBER,
    p_person_os                IN          HZ_PERSON_CUST_V2_BO,
    p_person_osr               IN          VARCHAR2,
    x_person_cust_v2_obj      OUT NOCOPY  NUMBER,
    x_return_status             OUT NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_msg_count                OUT NOCOPY  VARCHAR2,
    x_msg_data                 OUT NOCOPY  VARCHAR2
);

PROCEDURE get_v2_person_custs_created(
    p_event_id                IN          NUMBER,
    x_person_cust_v2_objs     OUT NOCOPY  HZ_PERSON_CUST_V2_BO_TBL,
    x_return_status             OUT NOCOPY  VARCHAR2,
    x_messages                  OUT NOCOPY  HZ_MESSAGE_OBJ_TBL
);

PROCEDURE get_v2_person_custs_updated(
    p_event_id                IN          NUMBER,
    x_person_cust_v2_objs     OUT NOCOPY  HZ_PERSON_CUST_V2_BO_TBL,
    x_return_status             OUT NOCOPY  VARCHAR2,
    x_messages                  OUT NOCOPY  HZ_MESSAGE_OBJ_TBL
);

PROCEDURE get_v2_person_cust_updated(
    p_event_id                IN          NUMBER,
    p_person_cust_id           IN          HZ_PERSON_CUST_V2_BO,
    x_person_cust_v2_obj      OUT NOCOPY  VARCHAR2,
    x_return_status             OUT NOCOPY  HZ_MESSAGE_OBJ_TBL,
    x_messages                  OUT NOCOPY  VARCHAR2
);
```

---

## Business Object API Attributes Information

This chapter covers the following topics:

- Business Object API Attributes Information Overview
- Customer Account Business Object API
- Customer Account Contact Business Object API
- Customer Account Site Business Object API
- EDI Business Object API
- EFT Business Object API
- E-Mail Business Object API
- Organization Business Object API
- Organization Contact Business Object API
- Organization Customer Business Object API
- Party Site Business Object API
- Person Business Object API
- Person Customer Business Object API
- Phone Business Object API
- SMS Business Object API
- Telex Business Object API
- Web Business Object API
- Business Structure Attributes
- Granular Entity Attributes

## **Business Object API Attributes Information Overview**

When you use a seeded business object API, attributes of the corresponding business object are affected differently depending on the type of API procedure. These sections provide information about what each API does with the business object attributes:

- Customer Account Business Object API, page 24-3
- Customer Account Contact Business Object API, page 24-25
- Customer Account Site Business Object API, page 24-33
- EDI Business Object API, page 24-42
- EFT Business Object API, page 24-51
- E-Mail Business Object API, page 24-59
- Organization Business Object API, page 24-67
- Organization Contact Business Object API, page 24-96
- Organization Customer Business Object API, page 24-108
- Party Site Business Object API, page 24-111
- Person Business Object API, page 24-121
- Person Customer Business Object API, page 24-136
- Phone Business Object API, page 24-139
- SMS Business Object API, page 24-148
- Telex Business Object API, page 24-158
- Web Business Object API, page 24-166

The seeded business object APIs also affect attributes of the business objects' embedded business structures and granular entities. See:

- Business Structure Attributes, page 24-174
- Granular Entity Attributes, page 24-245

## **Related Topics**

[Business Object APIs Overview, page 22-1](#)

# Customer Account Business Object API

For the details on this API, see: Customer Account Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Customer Account Business Object API procedures do with the business object attributes.

- Create Customer Account Business Object, page 24-3
- Update Customer Account Business Object, page 24-11
- Get Customer Account Business Object, page 24-20

The Save Customer Account Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Customer Account Business Object, *Oracle Integration Repository*.

## Related Topics

[Customer Account Business Object, page 20-6](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Customer Account Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Customer Account Business Object, *Oracle Integration Repository*.

Details of the Customer Account business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• status is lookup code in lookup type CODE_STATUS.</li> <li>• status cannot be set to null during update. It is defaulted to 'A' if user does not pass any value.</li> </ul>
account_number	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory Attribute. If GENERATE_CUSTOMER_NUMBER of AR_SYSTEM_PARAMETERS is on, if user has passed in an account_number, error out.</li> <li>• account_number will be generated from sequence. If autonumbering is off, if user has not passed in value, error out.</li> </ul>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
customer_type	VARCHAR2(30)	No	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE.
customer_class_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type CUSTOMER CLASS.

Attribute	Data Type	Required	Validation, Default, Comment
primary_salesrep_id	NUMBER	No	Validation: Must be valid salesrep_id from RA_SALESREPS table.
sales_channel_code	VARCHAR2(30)	No	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups.
order_type_id	NUMBER	No	Validation: Must be a valid order_type_id from the OE_ORDER_TYPES_V.
price_list_id	NUMBER	No	Validation: Must be a valid price_list_id from SO_PRICE_LISTS table.
tax_code	VARCHAR2(50)	No	Validation: Must be a valid tax_code from the AR_VAT_TAX table.
fob_point	VARCHAR2(30)	No	Validation: Validated against AR lookup type FOB.
freight_term	VARCHAR2(30)	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups.
ship_partial	VARCHAR2(1)	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	VARCHAR2(30)	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be passed only in single org case.
warehouse_id	NUMBER	No	Validation: Must be valid organization_id from the ORG_ORGANIZATION_DEFINITIONS table.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
tax_header_level_flag	VARCHAR2(1)	No	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
coterminate_day_month	VARCHAR2(6)	No	
primary_specialist_id	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f.
secondary_specialist_id	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f.
account_liable_flag	VARCHAR2(1)	No	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	NUMBER	No	
account_established_date	DATE	No	
account_termination_date	DATE	No	Validation: <ul style="list-style-type: none"> <li>• account_termination_date should be greater than account_established_date.</li> <li>• account_termination_date should be greater than account_activation_date.</li> </ul>
account_activation_date	DATE	No	Validation: account_activation_date should be greater than account_established_date.
department	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
held_bill_expiration_date	DATE	No	
hold_bill_flag	VARCHAR2(1)	No	Validation: hold_bill_flag is lookup code in lookup type YES/NO.
			Default: 'N' .
realtime_rate_flag	VARCHAR2(1)	No	
acct_life_cycle_status	VARCHAR2(30)	No	
account_name	VARCHAR2(240)	No	
deposit_refund_method	VARCHAR2(20)	No	
dormant_account_flag	VARCHAR2(1)	No	Validation: dormant_account_flag is lookup code in lookup type YES/NO.
			Default: 'N'.
npa_number	VARCHAR2(60)	No	
suspension_date	DATE	No	
source_code	VARCHAR2(150)	No	
comments	VARCHAR2(240)	No	
dates_negative_tolerance	NUMBER	No	
dates_positive_tolerance	NUMBER	No	
date_type_preference	VARCHAR2(20)	No	Validation: Validated against OE lookup type REQUEST_DATE_TYPE.
over_shipment_tolerance	NUMBER	No	
under_shipment_toleranc	NUMBER	No	e

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
over_return_tolerance	NUMBER	No	
under_return_tolerance	NUMBER	No	
item_cross_ref_pref	VARCHAR2(30)	No	Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYP ES.
ship_sets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• ship_sets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
			Default: N.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
arrivalsets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	VARCHAR2(1)	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO.
invoice_quantity_rule	VARCHAR2(30)	No	Validated against OE lookup type INVOICE_BASIS.
pricing_event	VARCHAR2(30)	No	Comment: This attribute is no longer used.
status_update_date	DATE	No	
autopay_flag	VARCHAR2(1)	No	Validation: autopay_flag is lookup code in lookup type YES/NO.
notify_flag	VARCHAR2(1)	No	Validation: notify_flag is lookup code in lookup type YES/NO.
last_batch_id	NUMBER	No	
selling_party_id	NUMBER	No	Validation: selling_party_id is foreign key of HZ_PARTIES.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
cancel_unshipped_lines_f lag	VARCHAR2(1)	No	<p>Validation: cancel_unshipped_lines_flag is lookup code in lookup type YES/NO.</p> <p>Comment: Indicates whether to cancel all unshipped orders/lines.</p> <p>Default: Null</p>

## Mapped Attributes

This procedure creates a mapping between the internally maintained CUST\_ACCT\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Business Objects and Structures

The Customer Account business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Customer Account Contact Business Object, page 24-25
- Create Customer Account Site Business Object, page 24-34
- Create Customer Profile Business Structure, page 24-200

## Attributes of Embedded Entities

The Customer Account business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Bank Account Use Entity, page 24-247
- Create Customer Account Relationship Entity, page 24-293
- Create Payment Method Entity, page 24-335

The Customer Account Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Customer Account Business Object API, page 24-3](#)

[Create Business Object API Procedures, page 22-6](#)

[Customer Account Business Object, page 20-6](#)

## Update Customer Account Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Customer Account Business Object, Oracle Integration Repository](#).

Details of the Customer Account business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Customer Account business object in the data model. You must provide sufficient information for these attributes to identify an existing Customer Account object, even if you are not updating any of the business object attributes.

Attribute	Data Type
cust_acct_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"><li>• status is lookup code in lookup type CODE_STATUS.</li><li>• status cannot be set to null during update. It is defaulted to 'A' if user does not pass any value.</li></ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
account_number	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>Mandatory Attribute. If GENERATE_CUSTOMER_NUMBER of AR_SYSTEM_PARAMETERS is on, if user has passed in an account_number, error out.</li> <li>account_number will be generated from sequence. If autonumbering is off, if user has not passed in value, error out.</li> </ul>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
customer_type	VARCHAR2(30)	No	Validation: customer_type is lookup code in AR lookup type CUSTOMER_TYPE.
customer_class_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type CUSTOMER CLASS.
primary_salesrep_id	NUMBER	No	Validation: Must be valid salesrep_id from RA_SALESREPS table.
sales_channel_code	VARCHAR2(30)	No	Validation: sales_channel_code is lookup code in lookup type SALES_CHANNEL in so_lookups.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
order_type_id	NUMBER	No	Validation: Must be a valid order_type_id from the OE_ORDER_TYPES_V.
price_list_id	NUMBER	No	Validation: Must be a valid price_list_id from SO_PRICE_LISTS table.
tax_code	VARCHAR2(50)	No	Validation: Must be a valid tax_code from the AR_VAT_TAX table.
fob_point	VARCHAR2(30)	No	Validation: Validated against AR lookup type FOB.
freight_term	VARCHAR2(30)	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups.
ship_partial	VARCHAR2(1)	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	VARCHAR2(30)	No	Validation: ship_via is foreign key to oe_ship_methods_v and can be passed only in single org case.
warehouse_id	NUMBER	No	Validation: Must be valid organization_id from the ORG_ORGANIZATION_DEFINITIONS table.
tax_header_level_flag	VARCHAR2(1)	No	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type TAX_ROUNDING_RULE.
ceterminate_day_month	VARCHAR2(6)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
primary_specialist_id	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f.
secondary_specialist_id	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f.
account_liable_flag	VARCHAR2(1)	No	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	NUMBER	No	
account_established_date	DATE	No	
account_termination_date	DATE	No	Validation: <ul style="list-style-type: none"> <li>• account_termination_date should be greater than account_established_date.</li> <li>• account_termination_date should be greater than account_activation_date.</li> </ul>
account_activation_date	DATE	No	Validation: account_activation_date should be greater than account_established_date.
department	VARCHAR2(30)	No	
held_bill_expiration_date	DATE	No	
hold_bill_flag	VARCHAR2(1)	No	Validation: hold_bill_flag is lookup code in lookup type YES/NO. Default: 'N' .
realtime_rate_flag	VARCHAR2(1)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
acct_life_cycle_status	VARCHAR2(30)	No	
account_name	VARCHAR2(240)	No	
deposit_refund_method	VARCHAR2(20)	No	
dormant_account_flag	VARCHAR2(1)	No	Validation: dormant_account_flag is lookup code in lookup type YES/NO. Default: 'N'.
npa_number	VARCHAR2(60)	No	
suspension_date	DATE	No	
source_code	VARCHAR2(150)	No	
comments	VARCHAR2(240)	No	
dates_negative_tolerance	NUMBER	No	
dates_positive_tolerance	NUMBER	No	
date_type_preference	VARCHAR2(20)	No	Validation: Validated against OE lookup type REQUEST_DATE_TYPE.
over_shipment_tolerance	NUMBER	No	
under_shipment_tolerance	NUMBER	No	
over_return_tolerance	NUMBER	No	
under_return_tolerance	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
item_cross_ref_pref	VARCHAR2(30)	No	<p>Validation: Allowed values are INT, CUST, and valid cross_reference_type from MTL_CROSS_REFERENCE_TYPES.</p>
ship_sets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• ship_sets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul> <p>Default: N.</p>
arrivalsets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
sched_date_push_flag	VARCHAR2(1)	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO.
invoice_quantity_rule	VARCHAR2(30)	No	Validated against OE lookup type INVOICE_BASIS.
pricing_event	VARCHAR2(30)	No	Comment: This attribute is no longer used.
status_update_date	DATE	No	
autopay_flag	VARCHAR2(1)	No	Validation: autopay_flag is lookup code in lookup type YES/NO.
notify_flag	VARCHAR2(1)	No	Validation: notify_flag is lookup code in lookup type YES/NO.
last_batch_id	NUMBER	No	
selling_party_id	NUMBER	No	Validation: selling_party_id is foreign key of HZ_PARTIES.
cancel_unshipped_lines_flag	VARCHAR2(1)	No	Validation: cancel_unshipped_lines_flag is lookup code in lookup type YES/NO.
			Comment: Indicates whether to cancel all unshipped orders/lines.
			Default: Null

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return

object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Business Objects and Structures

The Customer Account business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child. Otherwise, the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Customer Account Contact Business Object, page 24-25 or Update Customer Account Contact Business Object, page 24-28
- Create Customer Account Site Business Object, page 24-34 or Update Customer Account Site Business Object, page 24-37
- Create Customer Profile Business Structure, page 24-200 or Update Customer Profile Business Structure, page 24-209

## Attributes of Embedded Entities

The Customer Account business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Bank Account Use Entity, page 24-247 or Update Bank Account Use Entity, page 24-248
- Create Customer Account Relationship Entity, page 24-293 or Update Customer Account Relationship Entity, page 24-294
- Create Payment Method Entity, page 24-335 or Update Payment Method Entity, page 24-336

The Customer Account Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Customer Account Business Object API, page 24-3](#)

[Update Business Object API Procedures, page 22-9](#)

[Customer Account Business Object, page 20-6](#)

## Get Customer Account Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get Customer Account Business Object, Oracle Integration Repository](#).

Details of the Customer Account business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Customer Account Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Account business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
cust_acct_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
account_number	VARCHAR2(30)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
global_attribute1 to global_attribute20	VARCHAR2(150)
customer_type	VARCHAR2(30)
customer_class_code	VARCHAR2(30)
primary_salesrep_id	NUMBER
sales_channel_code	VARCHAR2(30)
order_type_id	NUMBER

---

---

<b>Attribute</b>	<b>Data Type</b>
price_list_id	NUMBER
tax_code	VARCHAR2(50)
fob_point	VARCHAR2(30)
freight_term	VARCHAR2(30)
ship_partial	VARCHAR2(1)
ship_via	VARCHAR2(30)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

<b>Attribute</b>	<b>Data Type</b>
invoice_quantity_rule	VARCHAR2(30)
pricing_event	VARCHAR2(30)
status_update_date	DATE
autopay_flag	VARCHAR2(1)
notify_flag	VARCHAR2(1)
last_batch_id	NUMBER
selling_party_id	NUMBER
cancel_unshipped_lines_flag	VARCHAR2(1)

## Attributes of Embedded Business Objects and Structures

The Customer Account business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Customer Account Contact Business Object, page 24-31
- Get Customer Account Site Business Object, page 24-40
- Get Customer Profile Business Structure, page 24-216

## Attributes of Embedded Entities

The Customer Account business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Bank Account Use Entity, page 24-249
- Get Customer Account Relationship Entity, page 24-296
- Get Payment Method Entity, page 24-337
- Source System Information Entity Attributes, page 24-358 (for the Customer Account Source System Information entity)

## Related Topics

- [Customer Account Business Object API, page 24-3](#)
- [Get Business Object API Procedures, page 22-13](#)
- [Customer Account Business Object, page 20-6](#)

## Customer Account Contact Business Object API

For the details on this API, see: Customer Account Contact Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Customer Account Contact Business Object API procedures do with the business object attributes.

- [Create Customer Account Contact Business Object, page 24-25](#)
- [Update Customer Account Contact Business Object, page 24-28](#)
- [Get Customer Account Contact Business Object, page 24-31](#)

The Save Customer Account Contact Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Customer Account Contact Business Object, *Oracle Integration Repository*.

## Related Topics

- [Customer Account Contact, page 20-11](#)
- [Business Object APIs Overview, page 22-1](#)
- [Business Object API Attributes Information Overview, page 24-2](#)

## Create Customer Account Contact Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Customer Account Contact Business Object, *Oracle Integration Repository*.

Details of the Customer Account Contact business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	Yes	<p>Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.</p> <p>Default: A.</p>
cust_acct_site_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Foreign key to hz_cust_acct_sites.</li> <li>• The cust_account_id in hz_cust_acct_sites which cust_acct_site_id points to should be same as the cust_account_id put in the hz_cust_account_roles.</li> </ul>
primary_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Primary flag is lookup code in lookup type YES/NO.</li> <li>• It is unique per cust_account_id or cust_acct_site_id.</li> </ul> <p>Default: N.</p>
role_type	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• It is a lookup code in lookup type ACCT_ROLE_TYPE.</li> <li>• The combination of CUST_ACCOUNT_ID, PARTY_ID, ROLE_TYPE should be unique.</li> <li>• Or the combination of CUST_ACCT_SITE_ID, PARTY_ID, ROLE_TYPE should be unique.</li> </ul>

Attribute	Data Type	Required	Validation, Default, Comment
source_code	VARCHAR2(150)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute25	VARCHAR2(150)	No	

### Mapped Attributes

This procedure creates a mapping between the internally maintained CUST\_ACCT\_CONTACT\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Attributes of Embedded Entities

The Customer Account Contact business object also has embedded child granular

entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Customer Account Relationship Entity, page 24-293
- Create Account Contact Role Entity, page 24-354

The Customer Account Contact Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Customer Account Contact Business Object API, page 24-25](#)

[Create Business Object API Procedures, page 22-6](#)

[Customer Account Contact, page 20-11](#)

## Update Customer Account Contact Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Customer Account Contact Business Object, Oracle Integration Repository](#).

Details of the Customer Account Contact business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Customer Account Contact business object in the data model. You must provide sufficient information for these attributes to identify an existing Customer Account Contact object, even if you are not updating any of the business object attributes.

Attribute	Data Type
cust_acct_contact_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	Yes	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: A.
cust_acct_site_id	NUMBER	No	Validation: <ul style="list-style-type: none"><li>• Foreign key to hz_cust_acct_sites.</li><li>• The cust_account_id in hz_cust_acct_sites which cust_acct_site_id points to should be same as the cust_account_id put in the hz_cust_account_roles.</li></ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Require d</b>	<b>Validation, Default, Comment</b>
primary_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>Primary flag is lookup code in lookup type YES/NO.</li> <li>It is unique per cust_account_id or cust_acct_site_id.</li> </ul> <p>Default: N.</p>
role_type	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>Mandatory attribute</li> <li>It is a lookup code in lookup type ACCT_ROLE_TYPE.</li> <li>The combination of CUST_ACCOUNT_ID, PARTY_ID, ROLE_TYPE should be unique.</li> <li>Or the combination of CUST_ACCT_SITE_ID, PARTY_ID, ROLE_TYPE should be unique.</li> </ul>
source_code	VARCHAR2(150)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute25	VARCHAR2(150)	No	

## Attributes of Embedded Entities

The Customer Account Contact business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Customer Account Relationship Entity, page 24-293 or Update Customer Account Relationship Entity, page 24-294

- Create Account Contact Role Entity, page 24-354 or Update Account Contact Role Entity, page 24-355

The Customer Account Contact Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Customer Account Contact Business Object API, page 24-25](#)

[Update Business Object API Procedures, page 22-9](#)

[Customer Account Contact, page 20-11](#)

## Get Customer Account Contact Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get Customer Account Contact Business Object, Oracle Integration Repository](#).

Details of the Customer Account Contact business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Customer Account Contact Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Account Contact business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
cust_acct_contact_id	NUMBER
orig_system_reference	VARCHAR2(255)
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
cust_acct_site_id	NUMBER
primary_flag	VARCHAR2(1)
role_type	VARCHAR2(30)
source_code	VARCHAR2(150)
attribute_category	VARCHAR2(30)
attribute1 to attribute25	VARCHAR2(150)
relationship_id	NUMBER
contact_person_id	NUMBER

Attribute	Data Type
contact_person_osr	VARCHAR2(255)
relationship_code	VARCHAR2(30)
relationship_type	VARCHAR2(30)
start_date	DATE

### Attributes of Embedded Entities

The Customer Account Contact business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Customer Account Relationship Entity, page 24-296
- Get Account Contact Role Entity, page 24-357
- Source System Information Entity Attributes, page 24-358 (for the Customer Account Contact Source System Information entity)

### Related Topics

[Customer Account Contact Business Object API](#), page 24-25

[Get Business Object API Procedures](#), page 22-13

[Customer Account Contact](#), page 20-11

## Customer Account Site Business Object API

For the details on this API, see: [Customer Account Site Business Object API, Oracle Integration Repository](#).

These sections provide information about what the Customer Account Site Business Object API procedures do with the business object attributes.

- [Create Customer Account Site Business Object](#), page 24-34
- [Update Customer Account Site Business Object](#), page 24-37
- [Get Customer Account Site Business Object](#), page 24-40

The Save Customer Account Site Business Object procedure calls either the Create or

Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Customer Account Site Business Object, *Oracle Integration Repository*.

## Related Topics

- [Customer Account Site, page 20-14](#)
- [Business Object APIs Overview, page 22-1](#)
- [Business Object API Attributes Information Overview, page 24-2](#)

## Create Customer Account Site Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Customer Account Site Business Object, *Oracle Integration Repository*.

Details of the Customer Account Site business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type CODE_STATUS. Default: A.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
global_attribute1 to attribute20	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
customer_category_code	VARCHAR2(30)	No	Validation: It is a lookup code in lookup type ADDRESS_CATEGORY.
language	VARCHAR2(4)	No	Validation: language is foreign key of fnd installed languages.
key_account_flag	VARCHAR2(1)	No	
tp_header_id	NUMBER	No	Validation: tp_header_id must be unique if pass in.
ece_tp_location_code	VARCHAR2(40)	No	Validation: The ece_tp_location_code should be unique for a customer within the organization.
primary_specialist_id	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f
secondary_specialist_id	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f
territory_id	NUMBER	No	
territory	VARCHAR2(30)	No	
translated_customer_name	VARCHAR2(50)	No	
org_id	NUMBER		

### Mapped Attributes

This procedure creates a mapping between the internally maintained CUST\_ACCT\_SITE\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

---

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Attributes of Embedded Business Objects and Structures

The Customer Account Site business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Customer Account Site Use Business Structure, page 24-185
- Create Customer Account Contact Business Object, page 24-25

### Attributes of Embedded Entities

The Customer Account Site business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Customer Account Relationship Entity, page 24-293

The Account Site Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the

business object level.

## Related Topics

[Customer Account Site Business Object API, page 24-33](#)

[Create Business Object API Procedures, page 22-6](#)

[Customer Account Site, page 20-14](#)

## Update Customer Account Site Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Customer Account Site Business Object, Oracle Integration Repository](#).

Details of the Customer Account Site business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Customer Account Site business object in the data model. You must provide sufficient information for these attributes to identify an existing Customer Account Site object, even if you are not updating any of the business object attributes.

Attribute	Data Type
cust_acct_site_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>Status cannot be set to null during update.</li> <li>Status is lookup code in lookup type CODE_STATUS.</li> </ul>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
global_attribute1 to attribute20	VARCHAR2(150)	No	
customer_category_code	VARCHAR2(30)	No	Validation: It is a lookup code in lookup type ADDRESS_CATEGORY.
language	VARCHAR2(4)	No	Validation: language is foreign key of fnd installed languages.
key_account_flag	VARCHAR2(1)	No	
tp_header_id	NUMBER	No	Validation: tp_header_id must be unique.
ece_tp_location_code	VARCHAR2(40)	No	

---

Attribute	Data Type	Required	Validation, Default, Comment
primary_specialist_id	NUMBER	No	Validation: primary_specialist_id is foreign key to per_all_people_f.
secondary_specialist_id	NUMBER	No	Validation: secondary_specialist_id is foreign key to per_all_people_f.
territory_id	NUMBER	No	
territory	VARCHAR2(30)	No	
translated_customer_name	VARCHAR2(50)	No	
org_id	NUMBER		

## Attributes of Embedded Business Objects and Structures

The Customer Account Site business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Customer Account Site Use Business Structure, page 24-185 or Update Customer Account Site Use Business Structure, page 24-191
- Create Customer Account Contact Business Object, page 24-25 or Update Customer Account Contact Business Object, page 24-28

## Attributes of Embedded Entities

The Customer Account Site business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Customer Account Relationship Entity, page 24-293 or Update Customer Account Relationship Entity, page 24-294

The Account Site Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an

original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Customer Account Site Business Object API, page 24-33](#)

[Update Business Object API Procedures, page 22-9](#)

[Customer Account Site, page 20-14](#)

## Get Customer Account Site Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get Customer Account Site Business Object, Oracle Integration Repository](#).

Details of the Customer Account Site business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Customer Account Site Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Account Site business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
cust_acct_site_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
global_attribute1 to attribute20	VARCHAR2(150)
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

---

Attribute	Data Type
territory_id	NUMBER
territory	VARCHAR2(30)
translated_customer_name	VARCHAR2(50)
party_site_id	NUMBER
org_id	NUMBER

### Attributes of Embedded Business Objects and Structures

The Customer Account Site business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Customer Account Site Use Business Structure, page 24-196
- Get Customer Account Contact Business Object, page 24-31

### Attributes of Embedded Entities

The Customer Account Site business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Customer Account Relationship Entity, page 24-296
- Source System Information Entity Attributes, page 24-358 (for the Account Site Source System Information entity)

### Related Topics

[Customer Account Site Business Object API, page 24-33](#)

[Get Business Object API Procedures, page 22-13](#)

[Customer Account Site, page 20-14](#)

## EDI Business Object API

For the details on this API, see: EDI Business Object API, *Oracle Integration Repository*.

These sections provide information about what the EDI Business Object API procedures do with the business object attributes.

- Create EDI Business Object, page 24-43
- Update EDI Business Object, page 24-46
- Get EDI Business Object, page 24-49

The Save EDI Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save EDI Business Object, *Oracle Integration Repository*.

## Related Topics

EDI, page 20-16

Business Object APIs Overview, page 22-1

Business Object API Attributes Information Overview, page 24-2

## Create EDI Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create EDI Business Object, *Oracle Integration Repository*.

Details of the EDI business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of Y.  Default: USER_ENTERED.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: <i>A</i> .
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.  Default: <i>N</i> .
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li><li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li></ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
edi_transaction_handling	VARCHAR2(25)	No	
edi_id_number	VARCHAR2(30)	No	
edi_payment_method	VARCHAR2(30)	No	
edi_payment_format	VARCHAR2(30)	No	
edi_remittance_method	VARCHAR2(30)	No	

Attribute	Data Type	Required	Validation, Default, Comment
edi_remittance_instruction	VARCHAR2(30)	No	
edi_tp_header_id	NUMBER	No	
edi_ece_tp_location_code	VARCHAR2(40)	No	

### Mapped Attributes

This procedure creates a mapping between the internally maintained EDI\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Attributes of Embedded Entities

The EDI business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the

embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[EDI Business Object API](#), page 24-42

[Create Business Object API Procedures](#), page 22-6

[EDI](#), page 20-16

## Update EDI Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update EDI Business Object](#), *Oracle Integration Repository*.

Details of the EDI business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct EDI business object in the data model. You must provide sufficient information for these attributes to identify an existing EDI object, even if you are not updating any of the business object attributes.

Attribute	Data Type
edi_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common

object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"><li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li><li>• Cannot be set to null during update.</li></ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
edi_transaction_handling	VARCHAR2(25)	No	
edi_id_number	VARCHAR2(30)	No	
edi_payment_method	VARCHAR2(30)	No	
edi_payment_format	VARCHAR2(30)	No	
edi_remittance_method	VARCHAR2(30)	No	
edi_remittance_instruction	VARCHAR2(30)	No	
edi_tp_header_id	NUMBER	No	
edi_ece_tp_location_code	VARCHAR2(40)	No	

## Attributes of Embedded Entities

The EDI business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[EDI Business Object API, page 24-42](#)

[Update Business Object API Procedures, page 22-9](#)

[EDI, page 20-16](#)

## Get EDI Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get EDI Business Object, Oracle Integration Repository](#).

Details of the EDI business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get EDI Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved EDI business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
edi_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
edi_transaction_handling	VARCHAR2(25)
edi_id_number	VARCHAR2(30)
edi_payment_method	VARCHAR2(30)
edi_payment_format	VARCHAR2(30)

---

Attribute	Data Type
edi_remittance_method	VARCHAR2(30)
edi_remittance_instruction	VARCHAR2(30)
edi_tp_header_id	NUMBER
edi_ece_tp_location_code	VARCHAR2(40)

### Attributes of Embedded Entities

The EDI business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

### Related Topics

[EDI Business Object API, page 24-42](#)

[Get Business Object API Procedures, page 22-13](#)

[EDI, page 20-16](#)

## EFT Business Object API

For the details on this API, see: EFT Business Object API, *Oracle Integration Repository*.

These sections provide information about what the EFT Business Object API procedures do with the business object attributes.

- Create EFT Business Object, page 24-52
- Update EFT Business Object, page 24-54
- Get EFT Business Object, page 24-57

The Save EFT Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save EFT Business Object, Oracle Integration Repository](#).

## Related Topics

EFT, page 20-19

Business Object APIs Overview, page 22-1

Business Object API Attributes Information Overview, page 24-2

## Create EFT Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create EFT Business Object, *Oracle Integration Repository*.

Details of the EFT business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
eft_transmission_program_id	NUMBER	No	
eft_printing_program_id	NUMBER	No	
eft_user_number	VARCHAR2(30)	No	
eft_swift_code	VARCHAR2(30)	No	

### Mapped Attributes

This procedure creates a mapping between the internally maintained EFT\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a

"common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The EFT business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

EFT Business Object API, page 24-51

Create Business Object API Procedures, page 22-6

EFT, page 20-19

## Update EFT Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update EFT Business Object, *Oracle Integration Repository*.

Details of the EFT business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct EFT business object in the data model. You must provide sufficient information for these

attributes to identify an existing EFT object, even if you are not updating any of the business object attributes.

Attribute	Data Type
eft_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
eft_transmission_program_id	NUMBER	No	
eft_printing_program_id	NUMBER	No	
eft_user_number	VARCHAR2(30)	No	
eft_swift_code	VARCHAR2(30)	No	

## Attributes of Embedded Entities

The EFT business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

EFT Business Object API, page 24-51

Update Business Object API Procedures, page 22-9

EFT, page 20-19

## Get EFT Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Get EFT Business Object, *Oracle Integration Repository*.

Details of the EFT business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get EFT Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved EFT business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
eft_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
eft_transmission_program_id	NUMBER

Attribute	Data Type
eft_printing_program_id	NUMBER
eft_user_number	VARCHAR2(30)
eft_swift_code	VARCHAR2(30)

### Attributes of Embedded Entities

The EFT business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

### Related Topics

EFT Business Object API, page 24-51

Get Business Object API Procedures, page 22-13

EFT, page 20-19

## E-Mail Business Object API

For the details on this API, see: E-Mail Business Object API, *Oracle Integration Repository*.

These sections provide information about what the E-Mail Business Object API procedures do with the business object attributes.

- Create E-Mail Business Object, page 24-60
- Update E-Mail Business Object, page 24-62
- Get E-Mail Business Object, page 24-65

The Save E-Mail Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save E-Mail Business Object, *Oracle Integration Repository*.

## Related Topics

E-Mail, page 20-21

Business Object APIs Overview, page 22-1

Business Object API Attributes Information Overview, page 24-2

## Create E-Mail Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create E-Mail Business Object, *Oracle Integration Repository*.

Details of the E-Mail business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
email_format	VARCHAR2(30)	No	<p>Validation: It is a lookup code in AR lookup type EMAIL_FORMAT.</p> <p>Default: MAILHTML.</p>
email_address	VARCHAR2(2000)	Yes	Validation: Mandatory attribute.

## Mapped Attributes

This procedure creates a mapping between the internally maintained EMAIL\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit

outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The E-Mail business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

E-Mail Business Object API, page 24-59

Create Business Object API Procedures, page 22-6

E-Mail, page 20-21

## Update E-Mail Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update E-Mail Business Object, *Oracle Integration Repository*.

Details of the E-Mail business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct E-Mail business object in the data model. You must provide sufficient information for these attributes to identify an existing E-Mail object, even if you are not updating any of the

business object attributes.

Attribute	Data Type
email_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
email_format	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type EMAIL_FORMAT.</li> <li>• Cannot set to null during update.</li> </ul>
email_address	VARCHAR2(2000)	No	Validation: Cannot be set to null during update.

## Attributes of Embedded Entities

The E-Mail business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[E-Mail Business Object API, page 24-59](#)

[Update Business Object API Procedures, page 22-9](#)

[E-Mail, page 20-21](#)

## Get E-Mail Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Get E-Mail Business Object, *Oracle Integration Repository*.

Details of the E-Mail business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get E-Mail Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved E-Mail business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
email_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
email_format	VARCHAR2(30)

---

Attribute	Data Type
email_address	VARCHAR2(2000)

---

## Attributes of Embedded Entities

The E-Mail business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

## Related Topics

[E-Mail Business Object API](#), page 24-59

[Get Business Object API Procedures](#), page 22-13

[E-Mail](#), page 20-21

## Organization Business Object API

For the details on this API, see: [Organization Business Object API, Oracle Integration Repository](#).

These sections provide information about what the Organization Business Object API procedures do with the business object attributes.

- [Create Organization Business Object](#), page 24-68
- [Update Organization Business Object](#), page 24-78
- [Get Organization Business Object](#), page 24-89

The Save Organization Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save Organization Business Object, Oracle Integration Repository](#).

## Related Topics

[Organization](#), page 20-23

[Business Object APIs Overview](#), page 22-1

## Create Organization Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Organization Business Object, *Oracle Integration Repository*.

Details of the Organization business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.  Default: SST.
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: A.
party_number	VARCHAR2(30)	Yes or No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else mandatory.
validated_flag	VARCHAR2(1)	No	Default: N.
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	
attribute_category	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1 to attribute24	VARCHAR2(150)	No	
organization_name	VARCHAR2(360)	Yes	Validation: Mandatory attribute.
duns_number_c	VARCHAR2(30)	No	
enquiry_duns	VARCHAR2(15)	No	
ceo_name	VARCHAR2(240)	No	
ceo_title	VARCHAR2(240)	No	
principal_name	VARCHAR2(240)	No	
principal_title	VARCHAR2(240)	No	
legal_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type LEGAL_STATUS.
control_yr	NUMBER	No	
employees_total	NUMBER	No	
hq_branch_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type HQ_BRANCH_IND.
branch_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
oob_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
line_of_business	VARCHAR2(240)	No	
cong_dist_code	VARCHAR2(2)	No	
sic_code	VARCHAR2(30)	No	
import_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
export_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
labor_surplus_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
debarment_ind	VARCHAR2(30)	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
minority_owned_type	VARCHAR2(30)	No	
woman_owned_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
disadv_8a_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
small_bus_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
rent_own_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type OWN_RENT_IND.
debarments_count	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
debarments_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
failure_score_natnl_perce ntile	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_override_co de	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentar y	VARCHAR2(30)	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTA RY.
global_failure_score	VARCHAR2(5)	No	Comment: This parameter should only be populated with data provided by D&B.
db_rating	VARCHAR2(5)	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary	VARCHAR2(30)	No	Validation: Validated against AR lookup type CREDIT_SCORE_COMMENTAR Y.
			Comment: This parameter should only be populated with data provided by D&B.
paydex_score	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_three_months_ag o	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
paydex_norm	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.
best_time_contact_begin	DATE	No	
best_time_contact_end	DATE	No	
organization_name_phonetic	VARCHAR2(320)	No	
tax_reference	VARCHAR2(50)	No	
gsa_indicator_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
jgzz_fiscal_code	VARCHAR2(20)	No	
analysis_fy	VARCHAR2(5)	No	
fiscal_yearend_month	VARCHAR2(30)	No	Validation: Validated against AR lookup type MONTH.
curr_fy_potential_revenue	NUMBER	No	
next_fy_potential_revenue	NUMBER	No	
year_established	NUMBER	No	
mission_statement	VARCHAR2(2000)	No	
organization_type	VARCHAR2(30)	No	
business_scope	VARCHAR2(20)	No	
corporation_class	VARCHAR2(60)	No	
known_as	VARCHAR2(240)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
known_as2 to known_as5	VARCHAR2(240)	No	
local_bus_iden_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type LOCAL_BUS_IDEN_TYPE.
local_bus_identifier	VARCHAR2(60)	No	
pref_functional_currency	VARCHAR2(30)	No	
registration_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type REGISTRATION TYPE.
total_employees_text	VARCHAR2(60)	No	
total_employees_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMPLOYEES_INDICATOR.
total_emp_est_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMP_EST_IND.
total_emp_min_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMP_MIN_IND.
parent_sub_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
incorp_year	NUMBER	No	
sic_code_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type SIC_CODE_TYPE.
public_private_ownershi p_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
internal_flag	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.  Default: N.
local_activity_code_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type LOCAL_ACTIVITY_CODE_TYPE.
local_activity_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type = value of local_activity_code_type.
emp_at_primary_adr	VARCHAR2(10)	No	
emp_at_primary_adr_text	VARCHAR2(12)	No	
emp_at_primary_adr_est _ind	VARCHAR2(30)	No	Validation: Validated against lookup type EMP_AT_PRIMARY_ADR_EST.
emp_at_primary_adr_mi n_ind	VARCHAR2(30)	No	Validation: Validated against lookup type EMP_AT_PRIMARY_ADR_MIN.
high_credit	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
avg_high_credit	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
total_payments	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_class	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
credit_score_natl_percentile	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_incd_default	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_age	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary 2 to credit_score_commentary 10	VARCHAR2(30)	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	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_incd_default	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_age	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
failure_score_commentary2 to failure_score_commentary10	VARCHAR2(30)	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	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
maximum_credit_currency_code	VARCHAR2(240)	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	NUMBER	No	Validation: Foreign Key to HZ_PARTIES.PARTY_ID.  Comment: This parameter should only be populated with data provided by D&B.
do_not_confuse_with	VARCHAR2(255)	No	

## Mapped Attributes

This procedure creates a mapping between the internally maintained ORGANIZATION\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Business Objects and Structures

The Organization business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Organization Contact Business Object, page 24-97
- Create Party Site Business Object, page 24-112
- Create Phone Business Object, page 24-140
- Create Telex Business Object, page 24-159
- Create E-Mail Business Object, page 24-60
- Create Web Business Object, page 24-166
- Create EDI Business Object, page 24-43
- Create EFT Business Object, page 24-52
- Create Financial Report Business Structure, page 24-229

## Attributes of Embedded Entities

The Organization business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311
- Create Party Preference Entity, page 24-327
- Create Relationship Entity, page 24-348
- Create Classification Entity, page 24-259
- Create Credit Rating Entity, page 24-276
- Create Certification Entity, page 24-251
- Create Financial Profile Entity, page 24-318
- Create Contact Preference Entity, page 24-264

The Organization Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

- Organization Business Object API, page 24-67
- Create Business Object API Procedures, page 22-6
- Organization, page 20-23

## Update Organization Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update Organization Business Object, *Oracle Integration Repository*.

Details of the Organization business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Organization business object in the data model. You must provide sufficient information for these attributes to identify an existing Organization object, even if you are not updating any of the business object attributes.

---

<b>Attribute</b>	<b>Data Type</b>
organization_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
party_number	VARCHAR2(30)	No	Validation: Cannot be updated.
validated_flag	VARCHAR2(1)	No	
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
organization_name	VARCHAR2(360)	Yes	Validation: Cannot be updated to null.
duns_number_c	VARCHAR2(30)	No	
enquiry_duns	VARCHAR2(15)	No	
ceo_name	VARCHAR2(240)	No	
ceo_title	VARCHAR2(240)	No	
principal_name	VARCHAR2(240)	No	
principal_title	VARCHAR2(240)	No	
legal_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type LEGAL_STATUS.
control_yr	NUMBER	No	
employees_total	NUMBER	No	
hq_branch_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type HQ_BRANCH_IND.
branch_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
oob_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
line_of_business	VARCHAR2(240)	No	
cong_dist_code	VARCHAR2(2)	No	
sic_code	VARCHAR2(30)	No	
import_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
export_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
labor_surplus_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
debarment_ind	VARCHAR2(30)	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
minority_owned_type	VARCHAR2(30)	No	
woman_owned_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
disadv_8a_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
small_bus_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
rent_own_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type OWN_RENT_IND.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
debarments_count	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
debarments_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_natnl_perce ntile	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_override_co de	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary	VARCHAR2(30)	No	Validation: Validated against AR lookup type FAILURE_SCORE_COMMENTA RY.
global_failure_score	VARCHAR2(5)	No	Comment: This parameter should only be populated with data provided by D&B.
db_rating	VARCHAR2(5)	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score	VARCHAR2(30)	No	Comment: This parameter should only be populated with data provided by D&B.

Attribute	Data Type	Required	Validation, Default, Comment
credit_score_commentary	VARCHAR2(30)	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	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_three_months ago	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.
paydex_norm	VARCHAR2(3)	No	Comment: This parameter should only be populated with data provided by D&B.
best_time_contact_begin	DATE	No	
best_time_contact_end	DATE	No	
organization_name_phonetic	VARCHAR2(320)	No	
tax_reference	VARCHAR2(50)	No	
gsa_indicator_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
jgzz_fiscal_code	VARCHAR2(20)	No	
analysis_fy	VARCHAR2(5)	No	
fiscal_yearend_month	VARCHAR2(30)	No	Validation: Validated against AR lookup type MONTH.
curr_fy_potential_revenue	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
next_fy_potential_revenue	NUMBER	No	
year_established	NUMBER	No	
mission_statement	VARCHAR2(2000)	No	
organization_type	VARCHAR2(30)	No	
business_scope	VARCHAR2(20)	No	
corporation_class	VARCHAR2(60)	No	
known_as	VARCHAR2(240)	No	
known_as2 to known_as5	VARCHAR2(240)	No	
local_bus_iden_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type LOCAL_BUS_IDEN_TYPE.
local_bus_identifier	VARCHAR2(60)	No	
pref_functional_currency	VARCHAR2(30)	No	
registration_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type REGISTRATION TYPE.
total_employees_text	VARCHAR2(60)	No	
total_employees_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMPLOYEES_INDICATOR.
total_emp_est_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMP_EST_IND.

Attribute	Data Type	Required	Validation, Default, Comment
total_emp_min_ind	VARCHAR2(30)	No	Validation: Validated against lookup type TOTAL_EMP_MIN_IND.
parent_sub_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
incorp_year	NUMBER	No	
sic_code_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type SIC_CODE_TYPE.
public_private_ownership_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
internal_flag	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO. Default: N
local_activity_code_type	VARCHAR2(30)	No	Validation: Validated against AR lookup type LOCAL_ACTIVITY_CODE_TYPE.
local_activity_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type = value of local_activity_code_type.
emp_at_primary_adr	VARCHAR2(10)	No	
emp_at_primary_adr_text	VARCHAR2(12)	No	
emp_at_primary_adr_est_ind	VARCHAR2(30)	No	Validation: Validated against lookup type EMP_AT_PRIMARY_ADR_EST.
emp_at_primary_adr_min_ind	VARCHAR2(30)	No	Validation: Validated against lookup type EMP_AT_PRIMARY_ADR_MIN.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
high_credit	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
avg_high_credit	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
total_payments	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_class	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_natl_percentile	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_incd_default	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_age	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
credit_score_commentary 2 to credit_score_commentary 10	VARCHAR2(30)	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	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.

Attribute	Data Type	Required	Validation, Default, Comment
failure_score_incd_default	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_age	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_date	DATE	No	Comment: This parameter should only be populated with data provided by D&B.
failure_score_commentary 2 to failure_score_commentary 10	VARCHAR2(30)	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_recomm endation	NUMBER	No	Comment: This parameter should only be populated with data provided by D&B.
maximum_credit_currenc y_code	VARCHAR2(240)	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	NUMBER	No	Validation: Foreign Key to HZ_PARTIES.PARTY_ID.
do_not_confuse_with	VARCHAR2(255)	No	

## Attributes of Embedded Business Objects and Structures

The Organization business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Organization Contact Business Object, page 24-97 or Update Organization Contact Business Object, page 24-101
- Create Party Site Business Object, page 24-112 or Update Party Site Business Object, page 24-115
- Create Phone Business Object, page 24-140 or Update Phone Business Object, page 24-143
- Create Telex Business Object, page 24-159 or Update Telex Business Object, page 24-161
- Create E-Mail Business Object, page 24-60 or Update E-Mail Business Object, page 24-62
- Create Web Business Object, page 24-166 or Update Web Business Object, page 24-169
- Create EDI Business Object, page 24-43 or Update EDI Business Object, page 24-46
- Create EFT Business Object, page 24-52 or Update EFT Business Object, page 24-54
- Create Financial Report Business Structure, page 24-229 or Update Financial Report Business Structure, page 24-231

## Attributes of Embedded Entities

The Organization business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311 or Update Extension Attribute Entity, page 24-312
- Create Party Preference Entity, page 24-327 or Update Party Preference Entity, page 24-328
- Create Relationship Entity, page 24-348 or Update Relationship Entity, page 24-350
- Create Classification Entity, page 24-259 or Update Classification Entity, page 24-260
- Create Credit Rating Entity, page 24-276 or Update Credit Rating Entity, page 24-282
- Create Certification Entity, page 24-251 or Update Certification Entity, page 24-252
- Create Financial Profile Entity, page 24-318 or Update Financial Profile Entity, page

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Organization Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

- Organization Business Object API, page 24-67
- Update Business Object API Procedures, page 22-9
- Organization, page 20-23

## Get Organization Business Object

For details including description, PL/SQL procedure, parameters, and validations, see: Organization Business Object, Oracle Integration Repository for the Get Organization Business Object procedure and all the event dependent Get procedures.

Details of the Organization business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Organization Business Object API procedure is event dependent or independent.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Organization business object. This attribute is only used when the business object is retrieved for an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
organization_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
organization_name	VARCHAR2(360)
duns_number_c	VARCHAR2(30)
enquiry_duns	VARCHAR2(15)

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

<b>Attribute</b>	<b>Data Type</b>
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)

---

<b>Attribute</b>	<b>Data Type</b>
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 to 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)

---

<b>Attribute</b>	<b>Data Type</b>
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

Attribute	Data Type
credit_score_commentary2 to 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 to failure_score_commentary10	VARCHAR2(30)
maximum_credit_recommendation	NUMBER
maximum_credit_currency_code	VARCHAR2(240)
displayed_duns_party_id	NUMBER
do_not_confuse_with	VARCHAR2(255)

## Attributes of Embedded Business Objects and Structures

The Organization business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Organization Contact Business Object, page 24-105
- Get Party Site Business Object, page 24-118
- Get Phone Business Object, page 24-146
- Get Telex Business Object, page 24-164
- Get E-Mail Business Object, page 24-65
- Get Web Business Object, page 24-172
- Get EDI Business Object, page 24-49

- Get EFT Business Object, page 24-57
- Get Financial Report Business Structure, page 24-234

## Attributes of Embedded Entities

The Organization business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Extension Attribute Entity, page 24-313
- Get Party Preference Entity, page 24-329
- Get Relationship Entity, page 24-352
- Get Classification Entity, page 24-262
- Get Credit Rating Entity, page 24-288
- Get Certification Entity, page 24-253
- Get Financial Profile Entity, page 24-321
- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Organization Source System Information entity)

## Related Topics

[Organization Business Object API, page 24-67](#)

[Get Business Object API Procedures, page 22-13](#)

[Organization, page 20-23](#)

## Organization Contact Business Object API

For the details on this API, see: [Organization Contact Business Object API, Oracle Integration Repository](#).

These sections provide information about what the Organization Contact Business Object API procedures do with the business object attributes.

- Create Organization Contact Business Object, page 24-97
- Update Organization Contact Business Object, page 24-101

- Get Organization Contact Business Object, page 24-105

The Save Organization Contact Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Organization Contact Business Object, *Oracle Integration Repository*.

## Related Topics

[Organization Contact, page 20-31](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Organization Contact Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Organization Contact Business Object, *Oracle Integration Repository*.

Details of the Organization Contact business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: A.
comments	VARCHAR2(240)	No	
contact_number	VARCHAR2(30)	Yes or No	Comment: If HZ_GENERATE_CONTACT_NUMBER = Y or null and caller does not pass any value, then it is generated from sequence, else caller is passed value is accepted.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
department_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type DEPARTMENT_TYPE.
department	VARCHAR2(60)	No	
title	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.
job_title	VARCHAR2(100)	No	
decision_maker_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
job_title_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type RESPONSIBILITY.
reference_use_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
rank	VARCHAR2(30)	No	
party_site_id	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 organization contact.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
relationship_code	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Validated against AR lookup type PARTY_RELATIONS_TYPE.</li> <li>• Required to be a valid forward_rel_code for the particular relationship type requested.</li> </ul>
relationship_type	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Must be a valid relationship_type from the HZ_RELATIONSHIP_TYPE table for the combination of subject_type, object_type, and relationship_code passed.</li> </ul>
relationship_comments	VARCHAR2(240)	No	
start_date	DATE	No	Validation: Mandatory attribute
end_date	DATE	No	<p>Validation: Must not be less than start_date.</p> <p>Default: 31-DEC-4712</p>
relationship_status	VARCHAR2(1)	No	<p>Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.</p> <p>Default: A.</p>

## Mapped Attributes

This procedure creates a mapping between the internally maintained ORG\_CONTACT\_ID and the attributes in this table. The created Source System

Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Attributes of Embedded Business Objects and Structures

The Organization Contact business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Contact Person Information Business Structure, page 24-174
- Create Party Site Business Object, page 24-112
- Create Phone Business Object, page 24-140
- Create Telex Business Object, page 24-159
- Create E-Mail Business Object, page 24-60
- Create Web Business Object, page 24-166
- Create SMS Business Object, page 24-149

## Attributes of Embedded Entities

The Organization Contact business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Relationship Entity, page 24-348
- Create Organization Contact Role Entity, page 24-323
- Create Contact Preference Entity, page 24-264

The Organization Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Organization Contact Business Object API](#), page 24-96

[Create Business Object API Procedures](#), page 22-6

[Organization Contact](#), page 20-31

## Update Organization Contact Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Organization Contact Business Object](#), *Oracle Integration Repository*.

Details of the Organization Contact business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Organization Contact business object in the data model. You must provide sufficient information for these attributes to identify an existing Organization Contact object, even if you are not updating any of the business object attributes.

Attribute	Data Type
org_contact_id	NUMBER
orig_system	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
orig_system_reference	VARCHAR2(255)

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li><li>• Cannot be updated to null.</li></ul>
comments	VARCHAR2(240)	No	
contact_number	VARCHAR2(30)	No	

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
department_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type DEPARTMENT_TYPE.
department	VARCHAR2(60)	No	
title	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.
job_title	VARCHAR2(100)	No	
decision_maker_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
job_title_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type RESPONSIBILITY.
reference_use_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
rank	VARCHAR2(30)	No	
party_site_id	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 organization contact.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
relationship_code	VARCHAR2(30)	No	Cannot be updated.
relationship_type	VARCHAR2(30)	No	Cannot be updated.
relationship_comments	VARCHAR2(240)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
start_date	DATE	No	Validation: Cannot be updated to null.
end_date	DATE	No	Validation: Cannot be less than start_date.
relationship_status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>

## Attributes of Embedded Business Objects and Structures

The Organization Contact business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Contact Person Information Business Structure, page 24-174 or Update Contact Person Information Business Structure, page 24-178
- Create Party Site Business Object, page 24-112 or Update Party Site Business Object, page 24-115
- Create Phone Business Object, page 24-140 or Update Phone Business Object, page 24-143
- Create Telex Business Object, page 24-159 or Update Telex Business Object, page 24-161
- Create E-Mail Business Object, page 24-60 or Update E-Mail Business Object, page 24-62
- Create Web Business Object, page 24-166 or Update Web Business Object, page 24-169
- Create SMS Business Object, page 24-149 or Update SMS Business Object, page 24-152

## Attributes of Embedded Entities

The Organization Contact business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Relationship Entity, page 24-348 or Update Relationship Entity, page 24-350
- Create Organization Contact Role Entity, page 24-323 or Update Organization Contact Role Entity, page 24-324
- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Organization Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Organization Contact Business Object API, page 24-96](#)

[Update Business Object API Procedures, page 22-9](#)

[Organization Contact, page 20-31](#)

## Get Organization Contact Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Get Organization Contact Business Object, *Oracle Integration Repository*.

Details of the Organization Contact business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Organization Contact Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Organization Contact business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE

- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
org_contact_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
comments	VARCHAR2(240)
contact_number	VARCHAR2(30)
department_code	VARCHAR2(30)
department	VARCHAR2(60)
title	VARCHAR2(30)
job_title	VARCHAR2(100)

Attribute	Data Type
decision_maker_flag	VARCHAR2(1)
job_title_code	VARCHAR2(30)
reference_use_flag	VARCHAR2(1)
rank	VARCHAR2(30)
party_site_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
relationship_code	VARCHAR2(30)
relationship_type	VARCHAR2(30)
relationship_comments	VARCHAR2(240)
start_date	DATE
end_date	DATE
relationship_status	VARCHAR2(1)

## Attributes of Embedded Business Objects and Structures

The Organization Contact business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Contact Person Information Business Structure, page 24-182
- Get Party Site Business Object, page 24-118
- Get Phone Business Object, page 24-146
- Get Telex Business Object, page 24-164
- Get E-Mail Business Object, page 24-65

- Get Web Business Object, page 24-172
- Get SMS Business Object, page 24-156

## Attributes of Embedded Entities

The Organization Contact business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Create Relationship Entity, page 24-348
- Create Organization Contact Role Entity, page 24-323
- Create Contact Preference Entity, page 24-264
- Source System Information Entity Attributes, page 24-358 (for the Organization Source System Information entity)

## Related Topics

[Organization Contact Business Object API, page 24-96](#)

[Get Business Object API Procedures, page 22-13](#)

[Organization Contact, page 20-31](#)

## Organization Customer Business Object API

For the details on this API, see: [Organization Customer Business Object API, Oracle Integration Repository](#).

These sections provide information about what the Organization Customer Business Object API procedures do with the business object attributes.

- Create Organization Customer Business Object, page 24-109
- Update Organization Customer Business Object, page 24-109
- Get Organization Customer Business Object, page 24-110

The Save Organization Customer Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save Organization Customer Business Object, Oracle Integration Repository](#).

## Related Topics

[Organization Customer, page 20-34](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Organization Customer Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Organization Customer Business Object, *Oracle Integration Repository*.

Details of the Organization Customer business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Attributes of Embedded Business Objects and Structures

The Organization Customer business object has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- [Create Organization Business Object, page 24-68](#)
- [Create Customer Account Business Object, page 24-3](#)

## Related Topics

[Organization Customer Business Object API, page 24-108](#)

[Create Business Object API Procedures, page 22-6](#)

[Organization Customer, page 20-34](#)

## Update Organization Customer Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update Organization Customer Business Object, *Oracle Integration Repository*.

Details of the Organization Customer business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Attributes of Embedded Business Objects and Structures

The Organization Customer business object has embedded child business objects and

structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Organization Business Object, page 24-68 or Update Organization Business Object, page 24-78
- Create Customer Account Business Object, page 24-3 or Update Customer Account Business Object, page 24-11

## Related Topics

[Organization Customer Business Object API, page 24-108](#)

[Update Business Object API Procedures, page 22-9](#)

[Organization Customer, page 20-34](#)

## Get Organization Customer Business Object

For details including description, PL/SQL procedure, parameters, and validations, see: Organization Customer Business Object, Oracle Integration Repository for the Get Organization Customer Business Object procedure and all the event dependent Get procedures.

Details of the Organization Customer business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Organization Customer Business Object API procedure is event dependent or independent.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Organization Customer business object. This attribute is only used when the business object is retrieved for an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Attributes of Embedded Business Objects and Structures

The Organization Customer business object has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Organization Business Object, page 24-89
- Get Customer Account Business Object, page 24-20

## Related Topics

[Organization Customer Business Object API, page 24-108](#)

[Get Business Object API Procedures, page 22-13](#)

[Organization Customer, page 20-34](#)

## Party Site Business Object API

For the details on this API, see: Party Site Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Party Site Business Object API procedures do with the business object attributes.

- Create Party Site Business Object, page 24-112
- Update Party Site Business Object, page 24-115
- Get Party Site Business Object, page 24-118

The Save Party Site Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Party Site Business Object, *Oracle Integration Repository*.

## Related Topics

[Party Site, page 20-35](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Party Site Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Party Site Business Object, *Oracle Integration Repository*.

Details of the Party Site business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.
party_site_number	VARCHAR2(30)	Yes or No	Validation: <ul style="list-style-type: none"><li>• Required when profile HZ_GENERATE_PARTY_SITE_NUMBER = N, else it is generated from sequence.</li><li>• Unique when passed in.</li></ul>
mailstop	VARCHAR2(60)	No	
identifying_address_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
party_site_name	VARCHAR2(240)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
language	VARCHAR2(4)	No	Validation: Foreign Key to fnd_languages.language_code (installed).
addressee	VARCHAR2(150)	No	
global_location_number	VARCHAR2(40)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot have nonnumeric characters.</li> <li>• Must be 13 digits long.</li> <li>• Must satisfy the check-digit algorithm.</li> </ul> <p>The check-digit algorithm is:</p> <ol style="list-style-type: none"> <li>1. Add the digits in even position. Multiply this sum by three.</li> <li>2. Add the digits in odd position, excluding the thirteenth digit.</li> <li>3. Add the values obtained in the previous two steps.</li> </ol> <p>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>

## Mapped Attributes

This procedure creates a mapping between the internally maintained PARTY\_SITE\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA\_SSM tables.

---

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Attributes of Embedded Business Objects and Structures

The Party Site business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Location Business Structure, page 24-237
- Create Phone Business Object, page 24-140
- Create Telex Business Object, page 24-159
- Create E-Mail Business Object, page 24-60
- Create Web Business Object, page 24-166

### Attributes of Embedded Entities

The Party Site business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the

embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311
- Create Party Site Use Entity, page 24-331
- Create Contact Preference Entity, page 24-264

The Party Site Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Party Site Business Object API, page 24-111](#)

[Create Business Object API Procedures, page 22-6](#)

[Party Site, page 20-35](#)

## Update Party Site Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Party Site Business Object, Oracle Integration Repository](#).

Details of the Party Site business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Party Site business object in the data model. You must provide sufficient information for these attributes to identify an existing Party Site object, even if you are not updating any of the business object attributes.

Attribute	Data Type
party_site_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li><li>• Cannot be set to null during update.</li></ul>
party_site_number	VARCHAR2(30)	No	Validation: Cannot be updated.
mailstop	VARCHAR2(60)	No	
identifying_address_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
party_site_name	VARCHAR2(240)	No	
attribute_category	VARCHAR2(30)	No	

Attribute	Data Type	Required	Validation, Default, Comment
attribute1 to attribute24	VARCHAR2(150)	No	
language	VARCHAR2(4)	No	Validation: Foreign Key to fnd_languages.language_code (installed).
addressee	VARCHAR2(150)	No	
global_location_number	VARCHAR2(40)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot have nonnumeric characters.</li> <li>• Must be 13 digits long.</li> <li>• Must satisfy the check-digit algorithm.</li> </ul> <p>The check-digit algorithm is:</p> <ol style="list-style-type: none"> <li>1. Add the digits in even position. Multiply this sum by three.</li> <li>2. Add the digits in odd position, excluding the thirteenth digit.</li> <li>3. Add the values obtained in the previous two steps.</li> </ol> <p>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>

## Attributes of Embedded Business Objects and Structures

The Party Site business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes

to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Location Business Structure, page 24-237 or Update Location Business Structure, page 24-240
- Create Phone Business Object, page 24-140 or Update Phone Business Object, page 24-143
- Create Telex Business Object, page 24-159 or Update Telex Business Object, page 24-161
- Create E-Mail Business Object, page 24-60 or Update E-Mail Business Object, page 24-62
- Create Web Business Object, page 24-166 or Update Web Business Object, page 24-169

## Attributes of Embedded Entities

The Party Site business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values.

For the embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311 or Update Extension Attribute Entity, page 24-312
- Create Party Site Use Entity, page 24-331 or Update Party Site Use Entity, page 24-332
- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Party Site Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Party Site Business Object API](#), page 24-111

[Update Business Object API Procedures](#), page 22-9

[Party Site](#), page 20-35

## Get Party Site Business Object

For details on this API procedure, including description, PL/SQL procedure,

parameters, and validations, see: Get Party Site Business Object, *Oracle Integration Repository*.

Details of the Party Site business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Party Site Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Party Site business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
party_site_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)

<b>Attribute</b>	<b>Data Type</b>
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
party_site_number	VARCHAR2(30)
mailstop	VARCHAR2(60)
identifying_address_flag	VARCHAR2(1)
party_site_name	VARCHAR2(240)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
language	VARCHAR2(4)
addressee	VARCHAR2(150)
global_location_number	VARCHAR2(40)

## Attributes of Embedded Business Objects and Structures

The Party Site business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Location Business Structure, page 24-243
- Get Phone Business Object, page 24-146
- Get Telex Business Object, page 24-164

- Get E-Mail Business Object, page 24-65
- Get Web Business Object, page 24-172

## Attributes of Embedded Entities

The Party Site business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Extension Attribute Entity, page 24-313
- Get Party Site Use Entity, page 24-334
- Create Contact Preference Entity, page 24-264
- Source System Information Entity Attributes, page 24-358 (for the Party Site Source System Information entity)

## Related Topics

[Party Site Business Object API](#), page 24-111

[Get Business Object API Procedures](#), page 22-13

[Party Site](#), page 20-35

## Person Business Object API

For the details on this API, see: Person Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Person Business Object API procedures do with the business object attributes.

- Create Person Business Object, page 24-122
- Update Person Business Object, page 24-127
- Get Person Business Object, page 24-132

The Save Person Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save Person Business Object](#), *Oracle Integration Repository*.

## Related Topics

[Person](#), page 20-38

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Person Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Person Business Object, *Oracle Integration Repository*.

Details of the Person business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B. orig_system with sst_flag value of Y. Default: SST.
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
party_number	VARCHAR2(30)	Yes or No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBR is Y, else unique and mandatory.
validated_flag	VARCHAR2(1)	No	Default: N
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
person_pre_name_adjunct	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.
person_first_name	VARCHAR2(150)	Yes or No	Validation: Either one of person_first_name or person_last_name should be passed in.
person_middle_name	VARCHAR2(60)	No	
person_last_name	VARCHAR2(150)	Yes or No	Validation: Either one of person_first_name or person_last_name should be passed in.
person_name_suffix	VARCHAR2(30)	No	
person_title	VARCHAR2(60)	No	
person_academic_title	VARCHAR2(30)	No	
person_previous_last_name	VARCHAR2(150)	No	
person_initials	VARCHAR2(6)	No	
known_as	VARCHAR2(240)	No	
known_as2 to known_as5	VARCHAR2(240)	No	
person_name_phonetic	VARCHAR2(320)	No	
person_first_name_phone	VARCHAR2(60)	No	
person_last_name_phone	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
middle_name_phonetic	VARCHAR2(60)	No	
tax_reference	VARCHAR2(50)	No	
jgzz_fiscal_code	VARCHAR2(20)	No	
person_iden_type	VARCHAR2(30)	No	
person_identifier	VARCHAR2(60)	No	
date_of_birth	DATE	No	
place_of_birth	VARCHAR2(60)	No	
date_of_death	DATE	No	
deceased_flag	VARCHAR2(1)	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.
gender	VARCHAR2(30)	No	
declared_ethnicity	VARCHAR2(60)	No	
marital_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type MARITAL_STATUS.
marital_status_effective_date	DATE	No	
personal_income	NUMBER	No	
head_of_household_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
household_income	NUMBER	No	

Attribute	Data Type	Required	Validation, Default, Comment
household_size	NUMBER	No	
rent_own_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type OWN_RENT_IND.
last_known_gps	VARCHAR2(60)	No	
internal_flag	VARCHAR2(2)	No	Default: N

### Mapped Attributes

This procedure creates a mapping between the internally maintained PERSON\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## **Attributes of Embedded Business Objects and Structures**

The Person business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- Create Party Site Business Object, page 24-112
- Create Phone Business Object, page 24-140
- Create E-Mail Business Object, page 24-60
- Create Web Business Object, page 24-166
- Create SMS Business Object, page 24-149
- Create Employment History Business Structure, page 24-220

## **Attributes of Embedded Entities**

The Person business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311
- Create Party Preference Entity, page 24-327
- Create Relationship Entity, page 24-348
- Create Classification Entity, page 24-259
- Create Person Language Entity, page 24-343
- Create Education Entity, page 24-306
- Create Citizenship Entity, page 24-254
- Create Person Interest Entity, page 24-339
- Create Certification Entity, page 24-251
- Create Financial Profile Entity, page 24-318
- Create Contact Preference Entity, page 24-264

The Person Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an

original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Person Business Object API, page 24-121](#)

[Create Business Object API Procedures, page 22-6](#)

[Person, page 20-38](#)

## Update Person Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update Person Business Object, *Oracle Integration Repository*.

Details of the Person business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Person business object in the data model. You must provide sufficient information for these attributes to identify an existing Person object, even if you are not updating any of the business object attributes.

Attribute	Data Type
person_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>
party_number	VARCHAR2(30)	Yes or No	Validation: Cannot be updated.
validated_flag	VARCHAR2(1)	No	
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
person_pre_name_adjunct	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.
person_first_name	VARCHAR2(150)	Yes or No	Validation: During update both person_first_name and person_last_name cannot be set to null.
person_middle_name	VARCHAR2(60)	No	

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
person_last_name	VARCHAR2(150)	Yes or No	Validation: During update both person_first_name and person_last_name cannot be set to null.
person_name_suffix	VARCHAR2(30)	No	
person_title	VARCHAR2(60)	No	
person_academic_title	VARCHAR2(30)	No	
person_previous_last_name	VARCHAR2(150)	No	
person_initials	VARCHAR2(6)	No	
known_as	VARCHAR2(240)	No	
known_as2 to known_as5	VARCHAR2(240)	No	
person_name_phonetic	VARCHAR2(320)	No	
person_first_name_phonetic	VARCHAR2(60)	No	
person_last_name_phonetic	VARCHAR2(60)	No	
middle_name_phonetic	VARCHAR2(60)	No	
tax_reference	VARCHAR2(50)	No	
jgzz_fiscal_code	VARCHAR2(20)	No	
person_iden_type	VARCHAR2(30)	No	
person_identifier	VARCHAR2(60)	No	
date_of_birth	DATE	No	
place_of_birth	VARCHAR2(60)	No	
date_of_death	DATE	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
deceased_flag	VARCHAR2(1)	No	Default: If this parameter is null and date_of_death is not null, then this parameter changes to Y.
gender	VARCHAR2(30)	No	
declared_ethnicity	VARCHAR2(60)	No	
marital_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type MARITAL_STATUS.
marital_status_effective_date	DATE	No	
personal_income	NUMBER	No	
head_of_household_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
household_income	NUMBER	No	
household_size	NUMBER	No	
rent_own_ind	VARCHAR2(30)	No	Validation: If the value is changed, then the value is validated against the AR lookup type OWN_RENT_IND.
last_known_gps	VARCHAR2(60)	No	
internal_flag	VARCHAR2(2)	No	

## Attributes of Embedded Business Objects and Structures

The Person business object also has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Party Site Business Object, page 24-112 or Update Party Site Business Object, page 24-115

- Create Phone Business Object, page 24-140 or Update Phone Business Object, page 24-143
- Create E-Mail Business Object, page 24-60 or Update E-Mail Business Object, page 24-62
- Create Web Business Object, page 24-166 or Update Web Business Object, page 24-169
- Create SMS Business Object, page 24-149 or Update SMS Business Object, page 24-152
- Create Employment History Business Structure, page 24-220 or Update Employment History Business Structure, page 24-223

### **Attributes of Embedded Entities**

The Person business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Extension Attribute Entity, page 24-311 or Update Extension Attribute Entity, page 24-312
- Create Party Preference Entity, page 24-327 or Update Party Preference Entity, page 24-328
- Create Relationship Entity, page 24-348 or Update Relationship Entity, page 24-350
- Create Classification Entity, page 24-259 or Update Classification Entity, page 24-260
- Create Person Language Entity, page 24-343 or Update Person Language Entity, page 24-344
- Create Education Entity, page 24-306 or Update Education Entity, page 24-308
- Create Citizenship Entity, page 24-254 or Update Citizenship Entity, page 24-256
- Create Person Interest Entity, page 24-339 or Update Person Interest Entity, page 24-340
- Create Certification Entity, page 24-251 or Update Certification Entity, page 24-252
- Create Financial Profile Entity, page 24-318 or Update Financial Profile Entity, page 24-320
- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Person Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

- [Person Business Object API, page 24-121](#)
- [Update Business Object API Procedures, page 22-9](#)
- [Person, page 20-38](#)

## Get Person Business Object

For details including description, PL/SQL procedure, parameters, and validations, see: Person Business Object, Oracle Integration Repository for the Get Person Business Object procedure and all the event dependent Get procedures.

Details of the Person business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Person Business Object API procedure is event dependent or independent.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Person business object. This attribute is only used when the business object is retrieved for an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
person_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
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 to known_as5	VARCHAR2(240)
person_name_phonetic	VARCHAR2(320)
person_first_name_phonetic	VARCHAR2(60)
person_last_name_phonetic	VARCHAR2(60)
middle_name_phonetic	VARCHAR2(60)
tax_reference	VARCHAR2(50)
jgzz_fiscal_code	VARCHAR2(20)
person_iden_type	VARCHAR2(30)
person_identifier	VARCHAR2(60)
date_of_birth	DATE
place_of_birth	VARCHAR2(60)
date_of_death	DATE
deceased_flag	VARCHAR2(1)
gender	VARCHAR2(30)
declared_ethnicity	VARCHAR2(60)

---

Attribute	Data Type
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)
internal_flag	VARCHAR2(2)

## Attributes of Embedded Business Objects and Structures

The Person business object also has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Party Site Business Object, page 24-118
- Get Phone Business Object, page 24-146
- Get E-Mail Business Object, page 24-65
- Get Web Business Object, page 24-172
- Get SMS Business Object, page 24-156
- Get Employment History Business Structure, page 24-226

## Attributes of Embedded Entities

The Person business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Extension Attribute Entity, page 24-313
- Get Party Preference Entity, page 24-329
- Get Relationship Entity, page 24-352
- Get Classification Entity, page 24-262
- Get Person Language Entity, page 24-346
- Get Education Entity, page 24-309
- Get Citizenship Entity, page 24-257
- Get Person Interest Entity, page 24-341
- Get Certification Entity, page 24-253
- Get Financial Profile Entity, page 24-321
- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Person Source System Information entity)

## Related Topics

- [Person Business Object API](#), page 24-121  
[Get Business Object API Procedures](#), page 22-13  
[Person](#), page 20-38

## Person Customer Business Object API

For the details on this API, see: [Person Customer Business Object API](#), *Oracle Integration Repository*.

These sections provide information about what the Person Customer Business Object API procedures do with the business object attributes.

- Create Person Customer Business Object, page 24-137
- Update Person Customer Business Object, page 24-137
- Get Person Customer Business Object, page 24-138

The Save Person Customer Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on

this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Person Customer Business Object, *Oracle Integration Repository*.

## Related Topics

[Person Customer](#), page 20-43

[Business Object APIs Overview](#), page 22-1

[Business Object API Attributes Information Overview](#), page 24-2

## Create Person Customer Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Person Customer Business Object, *Oracle Integration Repository*.

Details of the Person Customer business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Attributes of Embedded Business Objects and Structures

The Person Customer business object has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation as if you were calling its own API procedure. For the embedded business object or structure's attribute information, see:

- [Create Person Business Object](#), page 24-122
- [Create Customer Account Business Object](#), page 24-3

## Related Topics

[Person Customer Business Object API](#), page 24-136

[Create Business Object API Procedures](#), page 22-6

[Person Customer](#), page 20-43

## Update Person Customer Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update Person Customer Business Object, *Oracle Integration Repository*.

Details of the Person Customer business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Attributes of Embedded Business Objects and Structures

The Person Customer business object has embedded child business objects and structures. You provide embedded object types for the children, submitting the data for creation or update as if you were calling its own API procedure.

If you are updating the child object, you must provide sufficient identification attributes to uniquely identify the existing child, else the child is newly created.

For the embedded business object or structure's attribute information, see:

- Create Person Business Object, page 24-122 or Update Person Business Object, page 24-127
- Create Customer Account Business Object, page 24-3 or Update Customer Account Business Object, page 24-11

## Related Topics

[Person Customer Business Object API, page 24-136](#)

[Update Business Object API Procedures, page 22-9](#)

[Person Customer, page 20-43](#)

## Get Person Customer Business Object

For details including description, PL/SQL procedure, parameters, and validations, see: [Person Customer Business Object, Oracle Integration Repository for the Get Person Customer Business Object procedure and all the event dependent Get procedures](#).

Details of the Person Customer business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Person Customer Business Object API procedure is event dependent or independent.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Person Customer business object. This attribute is only used when the business object is retrieved for an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)

- **Possible Returned Values:**
  - *Event Independent*: Null
  - *Event Dependent*: CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Attributes of Embedded Business Objects and Structures

The Person Customer business object has embedded child business objects and structures. This procedure retrieves attribute values of these embedded items.

For the embedded business object or structure's attribute information, see:

- Get Person Business Object, page 24-132
- Get Customer Account Business Object, page 24-20

## Related Topics

[Person Customer Business Object API](#), page 24-136

[Get Business Object API Procedures](#), page 22-13

[Person Customer](#), page 20-43

## Phone Business Object API

For the details on this API, see: Phone Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Phone Business Object API procedures do with the business object attributes.

- Create Phone Business Object, page 24-140
- Update Phone Business Object, page 24-143
- Get Phone Business Object, page 24-146

The Save Phone Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save Phone Business Object](#), *Oracle Integration Repository*.

## Related Topics

[Phone](#), page 20-44

[Business Object APIs Overview](#), page 22-1

## Create Phone Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Phone Business Object, *Oracle Integration Repository*.

Details of the Phone business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
phone_calling_calendar	VARCHAR2(30)	No	
last_contact_dt_time	DATE	No	
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id.
phone_area_code	VARCHAR2(10)	No	
phone_country_code	VARCHAR2(10)	Yes or No	Validation: Foreign key to hz_phone_country_codes.phone_country_code.
phone_number	VARCHAR2(40)	No	<p>Validation: Mandatory if raw_phone_number is not passed in.</p> <p>If raw_phone_number is NULL, then you cannot update phone_number to NULL.</p>
phone_extension	VARCHAR2(20)	No	

Attribute	Data Type	Required	Validation, Default, Comment
phone_line_type	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Validated against AR lookup type PHONE_LINE_TYPE.</li> </ul>
raw_phone_number	VARCHAR2(60)	Yes or No	Validation: Mandatory if phone_number is not passed in.

## Mapped Attributes

This procedure creates a mapping between the internally maintained PHONE\_ID and the attributes in this table. . The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The Phone business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Phone Business Object API](#), page 24-139

[Create Business Object API Procedures](#), page 22-6

[Phone](#), page 20-44

## Update Phone Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Phone Business Object, Oracle Integration Repository](#).

Details of the Phone business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Phone business object in the data model. You must provide sufficient information for these attributes to identify an existing Phone object, even if you are not updating any of the business object attributes.

Attribute	Data Type
phone_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"><li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li><li>• Cannot be set to null during update.</li></ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
phone_calling_calendar	VARCHAR2(30)	No	
last_contact_dt_time	DATE	No	
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id.
phone_area_code	VARCHAR2(10)	No	
phone_country_code	VARCHAR2(10)	No	Validation: Foreign key to hz_phone_country_codes.phone_country_code.
phone_number	VARCHAR2(40)	No	Validation: Mandatory, if raw_phone_number is not passed in.
phone_extension	VARCHAR2(20)	No	
phone_line_type	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type PHONE_LINE_TYPE.</li> <li>• Cannot be updated to null</li> </ul>
raw_phone_number	VARCHAR2(60)	No	Validation: Mandatory, if phone_number is not passed in.

## Attributes of Embedded Entities

The Phone business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Phone Business Object API](#), page 24-139

[Update Business Object API Procedures](#), page 22-9

[Phone](#), page 20-44

## Get Phone Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get Phone Business Object](#), *Oracle Integration Repository*.

Details of the Phone business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Phone Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Phone business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures](#), page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
phone_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
phone_calling_calendar	VARCHAR2(30)

<b>Attribute</b>	<b>Data Type</b>
last_contact_dt_time	DATE
timezone_id	NUMBER
phone_area_code	VARCHAR2(10)
phone_country_code	VARCHAR2(10)
phone_number	VARCHAR2(40)
phone_extension	VARCHAR2(20)
phone_line_type	VARCHAR2(30)
raw_phone_number	VARCHAR2(60)

## Attributes of Embedded Entities

The Phone business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

## Related Topics

- Phone Business Object API, page 24-139
- Get Business Object API Procedures, page 22-13
- Phone, page 20-44

## SMS Business Object API

For the details on this API, see: SMS Business Object API, *Oracle Integration Repository*.

These sections provide information about what the SMS Business Object API procedures do with the business object attributes.

- Create SMS Business Object, page 24-149

- Update SMS Business Object, page 24-152
- Get SMS Business Object, page 24-156

The Save SMS Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save SMS Business Object, *Oracle Integration Repository*.

## Related Topics

[SMS, page 20-46](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create SMS Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create SMS Business Object, *Oracle Integration Repository*.

Details of the SMS business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
phone_calling_calendar	VARCHAR2(30)	No	
last_contact_dt_time	DATE	No	
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id.
phone_area_code	VARCHAR2(10)	No	
phone_country_code	VARCHAR2(10)	Yes or No	Validation: Foreign key to hz_phone_country_codes.phone_country_code.

Attribute	Data Type	Required	Validation, Default, Comment
phone_number	VARCHAR2(40)	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	VARCHAR2(20)	No	
phone_line_type	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Validated against AR lookup type PHONE_LINE_TYPE.</li></ul>
raw_phone_number	VARCHAR2(60)	Yes or No	Validation: Mandatory, if phone_number is not passed in.

## Mapped Attributes

This procedure creates a mapping between the internally maintained SMS\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

Attribute	Data Type
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return

object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The SMS business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[SMS Business Object API](#), page 24-148

[Create Business Object API Procedures](#), page 22-6

[SMS](#), page 20-46

## Update SMS Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update SMS Business Object, Oracle Integration Repository](#).

Details of the SMS business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct SMS business object in the data model. You must provide sufficient information for these attributes to identify an existing SMS object, even if you are not updating any of the business object attributes.

Attribute	Data Type
sms_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PU RPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PU RPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
phone_calling_calendar	VARCHAR2(30)	No	
last_contact_dt_time	DATE	No	

Attribute	Data Type	Required	Validation, Default, Comment
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezone.timezone_id.
phone_area_code	VARCHAR2(10)	No	
phone_country_code	VARCHAR2(10)	No	Validation: Foreign key to hz_phone_country_codes.phone_country_code.
phone_number	VARCHAR2(40)	No	Validation: Mandatory if raw_phone_number is not passed in.
phone_extension	VARCHAR2(20)	No	
phone_line_type	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type PHONE_LINE_TYPE.</li> <li>• Cannot be updated to null</li> </ul>
raw_phone_number	VARCHAR2(60)	No	Validation: Mandatory if phone_number is not passed in.

## Attributes of Embedded Entities

The SMS business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[SMS Business Object API, page 24-148](#)

[Update Business Object API Procedures, page 22-9](#)

[SMS, page 20-46](#)

## Get SMS Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get SMS Business Object, Oracle Integration Repository](#).

Details of the SMS business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get SMS Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved SMS business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures, page 22-16](#).

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
sms_id	NUMBER

<b>Attribute</b>	<b>Data Type</b>
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
phone_calling_calendar	VARCHAR2(30)
last_contact_dt_time	DATE
timezone_id	NUMBER
phone_area_code	VARCHAR2(10)
phone_country_code	VARCHAR2(10)

Attribute	Data Type
phone_number	VARCHAR2(40)
phone_extension	VARCHAR2(20)
phone_line_type	VARCHAR2(30)
raw_phone_number	VARCHAR2(60)

## Attributes of Embedded Entities

The SMS business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

## Related Topics

SMS Business Object API, page 24-148

Get Business Object API Procedures, page 22-13

SMS, page 20-46

## Telex Business Object API

For the details on this API, see: Telex Business Object API, *Oracle Integration Repository*.

These sections provide information about what the Telex Business Object API procedures do with the business object attributes.

- Create Telex Business Object, page 24-159
- Update Telex Business Object, page 24-161
- Get Telex Business Object, page 24-164

The Save Telex Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Save Telex Business Object, *Oracle Integration Repository*.

## Related Topics

[Telex, page 20-49](#)

[Business Object APIs Overview, page 22-1](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Create Telex Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Create Telex Business Object, *Oracle Integration Repository*.

Details of the Telex business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
telex_number	VARCHAR2(50)	Yes	Validation: Mandatory attribute.

## Mapped Attributes

This procedure creates a mapping between the internally maintained TELEX\_ID and the attributes in this table. . The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return

object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The Telex business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Telex Business Object API](#), page 24-158

[Create Business Object API Procedures](#), page 22-6

[Telex](#), page 20-49

## Update Telex Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Update Telex Business Object, *Oracle Integration Repository*.

Details of the Telex business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

## Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Telex business object in the data model. You must provide sufficient information for these attributes to identify an existing Telex object, even if you are not updating any of the business object attributes.

<b>Attribute</b>	<b>Data Type</b>
telex_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PU_RPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PU_RPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
telex_number	VARCHAR2(50)	No	Validation: Mandatory attribute.

## Attributes of Embedded Entities

The Telex business object also has embedded child granular entities. You provide

values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Telex Business Object API](#), page 24-158

[Update Business Object API Procedures](#), page 22-9

[Telex](#), page 20-49

## Get Telex Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Get Telex Business Object](#), *Oracle Integration Repository*.

Details of the Telex business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Telex Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

## Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Telex business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: [Get Updated Business Object Procedures](#), page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

## Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
telex_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
telex_number	VARCHAR2(50)

## Attributes of Embedded Entities

The Telex business object also has embedded child granular entities. This procedure

retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

## Related Topics

[Telex Business Object API](#), page 24-158

[Get Business Object API Procedures](#), page 22-13

[Telex](#), page 20-49

## Web Business Object API

For the details on this API, see: [Web Business Object API, Oracle Integration Repository](#).

These sections provide information about what the Web Business Object API procedures do with the business object attributes.

- Create Web Business Object, page 24-166
- Update Web Business Object, page 24-169
- Get Web Business Object, page 24-172

The Save Web Business Object procedure calls either the Create or Update procedure. See the corresponding sections for attribute information. For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Save Web Business Object, Oracle Integration Repository](#).

## Related Topics

[Web](#), page 20-51

[Business Object APIs Overview](#), page 22-1

[Business Object API Attributes Information Overview](#), page 24-2

## Create Web Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Create Web Business Object, Oracle Integration Repository](#).

Details of the Web business object are captured as attributes of the object type. You provide values in the input object type to create the attribute values in Oracle Trading

## Community Architecture.

### Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS.orig_system with sst_flag value of Y. Default: USER_ENTERED.
status	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS. Default: A.
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO. Default: N.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
contact_point_purpose	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
web_type	VARCHAR2(60)	Yes	Validation: Mandatory attribute.
URL	VARCHAR2(2000)	Yes	Validation: Mandatory attribute.

## Mapped Attributes

This procedure creates a mapping between the internally maintained WEB\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

## Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

## Attributes of Embedded Entities

The Web business object also has embedded child granular entities. You provide values for the attributes of the business entities to create the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Web Business Object API](#), page 24-166

[Create Business Object API Procedures](#), page 22-6

[Web](#), page 20-51

## Update Web Business Object

For the details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: [Update Web Business Object](#), *Oracle Integration Repository*.

Details of the Web business object are captured as attributes of the object type. You provide values in the input object type to update the attribute values in Oracle Trading Community Architecture.

### Identification Attributes

This procedure uses the attributes in this table to uniquely identify the correct Web business object in the data model. You must provide sufficient information for these attributes to identify an existing Web object, even if you are not updating any of the business object attributes.

Attribute	Data Type
web_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

## Accepted Attributes

This procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be set to null during update.</li> </ul>
primary_flag	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type YES/NO.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

Attribute	Data Type	Required	Validation, Default, Comment
contact_point_purpose	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE when contact_point_type is not WEB.</li> <li>• Validated against AR lookup type CONTACT_POINT_PURPOSE_WEB when contact_point_type = WEB.</li> </ul>
primary_by_purpose	VARCHAR2(30)	No	Validation: It is a lookup code in AR lookup type YES/NO.
web_type	VARCHAR2(60)	No	Validation: Mandatory attribute.
URL	VARCHAR2(2000)	No	Validation: Mandatory attribute.

## Attributes of Embedded Entities

The Web business object also has embedded child granular entities. You provide values for the attributes of the business entities to create or update the attribute values. For the embedded granular entity's attribute information, see:

- Create Contact Preference Entity, page 24-264 or Update Contact Preference Entity, page 24-269

The Contact Point Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

## Related Topics

[Web Business Object API, page 24-166](#)

[Update Business Object API Procedures, page 22-9](#)

[Web, page 20-51](#)

## Get Web Business Object

For details on this API procedure, including description, PL/SQL procedure, parameters, and validations, see: Get Web Business Object, *Oracle Integration Repository*.

Details of the Web business object are captured as attributes of the object type. You provide identification information to retrieve attribute values of the identified business object.

Except for the ACTION\_TYPE attribute, these business object attributes are similarly extracted whether the Get Web Business Object API procedure is called independent of business events, or as part of an event dependent Get business object call.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Web business object. This attribute is only used when the business object is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

This procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
web_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE

Attribute	Data Type
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(30)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
contact_point_purpose	VARCHAR2(30)
primary_by_purpose	VARCHAR2(30)
web_type	VARCHAR2(60)
URL	VARCHAR2(2000)

## Attributes of Embedded Entities

The Web business object also has embedded child granular entities. This procedure retrieves attribute values of these embedded entities. For the embedded granular entity's attribute information, see:

- Get Contact Preference Entity, page 24-274
- Source System Information Entity Attributes, page 24-358 (for the Contact Point Source System Information entity)

## Related Topics

- [Web Business Object API, page 24-166](#)
- [Get Business Object API Procedures, page 22-13](#)
- [Web, page 20-51](#)

## Business Structure Attributes

Many seeded business objects have embedded business structures. When you use the seeded business object APIs, the embedded structures are correspondingly created, updated, or retrieved.

These sections provide information about what seeded business object API procedures do with the attributes of embedded business structures.

- [Contact Person Information Business Structure Attributes, page 24-174](#)
- [Customer Account Site Use Business Structure Attributes, page 24-185](#)
- [Customer Profile Business Structure Attributes, page 24-200](#)
- [Employment History Business Structure Attributes, page 24-220](#)
- [Financial Report Business Structure Attributes, page 24-228](#)
- [Location Business Structure Attributes, page 24-237](#)

## Related Topics

- [Business Object API Attributes Information Overview, page 24-2](#)
- [Business Object APIs Overview, page 22-1](#)

## Contact Person Information Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Contact Person Information business structure.

### Create Contact Person Information Business Structure

As part of creating a business object, you can provide data to create the Contact Person Information business structure.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.  Default: SST.
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: A.
party_number	VARCHAR2(30)	Yes or No	Validation: Generated by sequence if profile HZ_GENERATE_PARTY_NUMBER is Y, else unique and mandatory.
validated_flag	VARCHAR2(1)	No	Default: N
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
person_pre_name_adjunct	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.
person_first_name	VARCHAR2(150)	Yes or No	Validation: Either one of person_first_name or person_last_name should be passed in.
person_middle_name	VARCHAR2(60)	No	
person_last_name	VARCHAR2(150)	Yes or No	Validation: Either one of person_first_name or person_last_name should be passed in.
person_name_suffix	VARCHAR2(30)	No	
person_title	VARCHAR2(60)	No	
person_academic_title	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
person_previous_last_name	VARCHAR2(150)	No	
person_initials	VARCHAR2(6)	No	
known_as	VARCHAR2(240)	No	
known_as2 to known_as5	VARCHAR2(240)	No	
person_name_phonetic	VARCHAR2(320)	No	
person_first_name_phonetic	VARCHAR2(60)	No	
person_last_name_phonetic	VARCHAR2(60)	No	
middle_name_phonetic	VARCHAR2(60)	No	
tax_reference	VARCHAR2(50)	No	
jgzz_fiscal_code	VARCHAR2(20)	No	
person_iden_type	VARCHAR2(30)	No	
person_identifier	VARCHAR2(60)	No	
date_of_birth	DATE	No	
place_of_birth	VARCHAR2(60)	No	
date_of_death	DATE	No	
deceased_flag	VARCHAR2(1)	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.
gender	VARCHAR2(30)	No	
declared_ethnicity	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
marital_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type MARITAL_STATUS.
marital_status_effective_date	DATE	No	
personal_income	NUMBER	No	
head_of_household_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
household_income	NUMBER	No	
household_size	NUMBER	No	
rent_own_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type OWN_RENT_IND.
last_known_gps	VARCHAR2(60)	No	
internal_flag	VARCHAR2(2)	No	Default: N

### Mapped Attributes

This procedure creates a mapping between the internally maintained PERSON\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA\_SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common

object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Attributes of Embedded Entities

The Contact Person Information business structure has a couple of embedded granular entities. For the embedded entity's attribute information, see: Create Extension Attribute Entity, page 24-311.

The Contact Person Information Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

### Update Contact Person Information Business Structure

As part of updating a business object, you can provide data to create or update the Contact Person Information business structure.

#### Identification Attributes

The API procedure uses the attributes in this table to uniquely identify the correct Contact Person Information business structure in the data model. You must provide sufficient information for these attributes to identify an existing Contact Person Information business structure.

Attribute	Data Type	Logical Key
person_id	NUMBER	No
orig_system	VARCHAR2(30)	No
orig_system_reference	VARCHAR2(255)	No

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• It is a lookup code in AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>
party_number	VARCHAR2(30)	Yes or No	Validation: Cannot be updated.
validated_flag	VARCHAR2(1)	No	
category_code	VARCHAR2(30)	No	Validation: Validated against lookup type CUSTOMER_CATEGORY.
salutation	VARCHAR2(60)	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute24	VARCHAR2(150)	No	
person_pre_name_adjunct	VARCHAR2(30)	No	Validation: Validated against AR lookup type CONTACT_TITLE.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
person_first_name	VARCHAR2(150)	Yes or No	Validation: During update both person_first_name and person_last_name cannot be set to null.
person_middle_name	VARCHAR2(60)	No	
person_last_name	VARCHAR2(150)	Yes or No	Validation: During update both person_first_name and person_last_name cannot be set to null.
person_name_suffix	VARCHAR2(30)	No	
person_title	VARCHAR2(60)	No	
person_academic_title	VARCHAR2(30)	No	
person_previous_last_name	VARCHAR2(150)	No	
person_initials	VARCHAR2(6)	No	
known_as	VARCHAR2(240)	No	
known_as2 to known_as5	VARCHAR2(240)	No	
person_name_phonetic	VARCHAR2(320)	No	
person_first_name_phone	VARCHAR2(60)	No	
person_last_name_phonet	VARCHAR2(60)	No	
middle_name_phonetic	VARCHAR2(60)	No	
tax_reference	VARCHAR2(50)	No	
jgzz_fiscal_code	VARCHAR2(20)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
person_iden_type	VARCHAR2(30)	No	
person_identifier	VARCHAR2(60)	No	
date_of_birth	DATE	No	
place_of_birth	VARCHAR2(60)	No	
date_of_death	DATE	No	
deceased_flag	VARCHAR2(1)	No	Default: If this parameter is null and date_of_death is not null, then this parameter changes to Y.
gender	VARCHAR2(30)	No	
declared_ethnicity	VARCHAR2(60)	No	
marital_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type MARITAL_STATUS.
marital_status_effective_date	DATE	No	
personal_income	NUMBER	No	
head_of_household_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
household_income	NUMBER	No	
household_size	NUMBER	No	
rent_own_ind	VARCHAR2(30)	No	Validation: If the value is changed, then the value is validated against the AR lookup type OWN_RENT_IND.
last_known_gps	VARCHAR2(60)	No	
internal_flag	VARCHAR2(2)	No	

### Attributes of Embedded Entities

The Contact Person Information business structure has a couple of embedded granular entities. For the embedded entity's attribute information, see: Update Extension Attribute Entity, page 24-312.

The Contact Person Information Source System Information entity is used for extraction only, not for creating the business object. Do not pass any data for this entity. You can create an original system and original system reference mapping by passing those values at the business object level.

### Get Contact Person Information Business Structure

As part of getting a business object, embedded Contact Person Information business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Contact Person Information business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
person_id	NUMBER
last_update_date	DATE

---

<b>Attribute</b>	<b>Data Type</b>
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
party_number	VARCHAR2(30)
validated_flag	VARCHAR2(1)
status	VARCHAR2(1)
category_code	VARCHAR2(30)
salutation	VARCHAR2(60)
attribute_category	VARCHAR2(30)
attribute1 to attribute24	VARCHAR2(150)
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)

---

---

<b>Attribute</b>	<b>Data Type</b>
person_previous_last_name	VARCHAR2(150)
person_initials	VARCHAR2(6)
known_as	VARCHAR2(240)
known_as2 to known_as5	VARCHAR2(240)
person_name_phonetic	VARCHAR2(320)
person_first_name_phonetic	VARCHAR2(60)
person_last_name_phonetic	VARCHAR2(60)
middle_name_phonetic	VARCHAR2(60)
tax_reference	VARCHAR2(50)
jgzz_fiscal_code	VARCHAR2(20)
person_iden_type	VARCHAR2(30)
person_identifier	VARCHAR2(60)
date_of_birth	DATE
place_of_birth	VARCHAR2(60)
date_of_death	DATE
deceased_flag	VARCHAR2(1)
gender	VARCHAR2(30)
declared_ethnicity	VARCHAR2(60)
marital_status	VARCHAR2(30)
marital_status_effective_date	DATE

---

Attribute	Data Type
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)
internal_flag	VARCHAR2(2)

### Attributes of Embedded Entities

The Contact Person Information business structure has a couple of embedded granular entities. For the embedded entity's attribute information, see:

- Get Extension Attribute Entity, page 24-313.
- Source System Information Entity Attributes, page 24-358 (for the Person Source System Information entity)

### Related Topics

[Business Structure Attributes, page 24-174](#)

## Customer Account Site Use Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Customer Account Site Use business structure.

### Create Customer Account Site Use Business Structure

As part of creating a business object, you can provide data to create the Customer Account Site Use business structure.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: It is a lookup code in AR lookup type REGISTRY_STATUS.  Default: <i>A</i> .
site_use_code	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Cannot be updated to null.</li></ul>
primary_flag	VARCHAR2(1)	No	Validation: Primary_flag is lookup code in lookup type YES/NO.  Default: <i>N</i>
location	VARCHAR2(40)	No	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• It will be generated from sequence if user does not pass in and AUTO_SITE_NUMBERING in AR_SYSTEM_PARAMETERS is on.</li><li>• location must be unique within a customer account/ site_use_type.</li></ul>
bill_to_site_use_id	NUMBER	No	
sic_code	VARCHAR2(30)	No	
payment_term_id	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS.
gsa_indicator	VARCHAR2(1)	No	Validation: Primary_flag is lookup code in lookup type YES/NO.  Default: <i>N</i>
ship_partial	VARCHAR2(1)	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_via	VARCHAR2(30)	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type FOB.
order_type_id	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.
freight_term	VARCHAR2(30)	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups.
warehouse_id	NUMBER	No	Validation: Valid organization_id from ORG_ORGANIZATION_DEFINITIONS.
territory_id	NUMBER	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute25	VARCHAR2(150)	No	
tax_reference	VARCHAR2(50)	No	
sort_priority	NUMBER	No	
tax_code	VARCHAR2(50)	No	Validation: Must be a valid tax_code from AR_VAT_TAX.
demand_class_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type DEMAND_CLASS.
tax_header_level_flag	VARCHAR2(1)	No	
tax_rounding_rule	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
primary_salesrep_id	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_trx_id	NUMBER	No	Validation: Valid receivables trx_id from AR_RECEIVABLES_TRX.
dates_negative_tolerance	NUMBER	No	
dates_positive_tolerance	NUMBER	No	
date_type_preference	VARCHAR2(20)	No	Validation: Validated against OE lookup type REQUEST_DATE_TYPE.
over_shipment_tolerance	NUMBER	No	
under_shipment_tolerance	NUMBER	No	
item_cross_ref_pref	VARCHAR2(30)	No	Validation: Allowed values are INT, CUST, and cross_reference_type value from MTL_CROSS_REFERENCE_TYPES.
over_return_tolerance	NUMBER	No	
under_return_tolerance	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_sets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to <i>Y</i>, then arrivalsets_include_lines_flag is always <i>N</i>.</li> <li>• If arrivalsets_include_lines_flag is <i>Y</i>, then ship_sets_include_lines_flag is always <i>N</i>.</li> </ul> <p>Default: <i>N</i></p>
arrivalsets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to <i>Y</i>, then arrivalsets_include_lines_flag is always <i>N</i>.</li> <li>• If arrivalsets_include_lines_flag is <i>Y</i>, then ship_sets_include_lines_flag is always <i>N</i>.</li> </ul> <p>Default: <i>N</i></p>
sched_date_push_flag	VARCHAR2(1)	No	<p>Validation: sched_date_push_flag is lookup code in lookup type YES/NO.</p> <p>Default: <i>N</i></p>
invoice_quantity_rule	VARCHAR2(30)	No	
pricing_event	VARCHAR2(30)	No	Comment: This attribute is no longer used.
gl_id_rec	NUMBER	No	Validation: gl_id_rec is valid gl field.
gl_id_rev	NUMBER	No	Validation: gl_id_rev is valid gl field.
gl_id_tax	NUMBER	No	Validation: gl_id_tax is valid gl field.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gl_id_freight	NUMBER	No	Validation: gl_id_freight is valid gl field.
gl_id_clearing	NUMBER	No	Validation: gl_id_clearing is valid gl field.
gl_id_unbilled	NUMBER	No	Validation: gl_id_unbilled is valid gl field.
gl_id_unearned	NUMBER	No	Validation: gl_id_unearned is valid gl field.
gl_id_unpaid_rec	NUMBER	No	Validation: gl_id_unpaid_rec is valid gl field.
gl_id_remittance	NUMBER	No	Validation: gl_id_remittance is valid gl field.
gl_id_factor	NUMBER	No	Validation: gl_id_factor is valid gl field.
tax_classification	VARCHAR2(30)	No	
org_id	NUMBER	No	

#### Mapped Attributes

This procedure creates a mapping between the internally maintained PERSON\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The

attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Attributes of Embedded Entities

The Customer Account Site Use business structure has an embedded business structure and embedded granular entities. For the embedded structure and entity attribute information, see:

- Create Customer Profile Business Structure, page 24-200.
- Create Bank Account Use Entity, page 24-247
- Create Payment Method Entity, page 24-335

#### Update Customer Account Site Use Business Structure

As part of updating a business object, you can provide data to create or update the Customer Account Site Use business structure.

#### Identification Attributes

The API procedure uses the attributes in this table to uniquely identify the correct Customer Account Site Use business structure in the data model. You must provide sufficient information for these attributes to identify an existing Customer Account Site Use business structure.

Attribute	Data Type	Logical Key
site_use_id	NUMBER	No
orig_system	VARCHAR2(30)	No
orig_system_reference	VARCHAR2(255)	No

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• status cannot be set to null during update.</li> <li>• status is lookup code in lookup type CODE_STATUS.</li> </ul>
site_use_code	VARCHAR2(30)	No	Validation: Cannot be updated
primary_flag	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• primary_flag cannot be set to null during update.</li> <li>• primary_flag is lookup code in lookup type YES/NO.</li> </ul>
location	VARCHAR2(40)	No	Validation: Can only be updated if the AUTO_SITE_NUMBERING profile option in AR_SYSTEMS_PARAMETERS is unchecked.
bill_to_site_use_id	NUMBER	No	
sic_code	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
payment_term_id	NUMBER	No	Validation: Must be a valid term_id from RA_TERMS.
gsa_indicator	VARCHAR2(1)	No	Validation: gsa_indicator is lookup code in lookup type YES/NO
ship_partial	VARCHAR2(1)	No	Comment: This attribute is no longer used. Functionality replaced by ship_sets_include_lines_flag.
ship_via	VARCHAR2(30)	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	VARCHAR2(30)	No	Validation: Validated against AR lookup type FOB.
order_type_id	NUMBER	No	Validation: Valid order_type_id from OE_ORDER_TYPES_V.
price_list_id	NUMBER	No	Validation: Valid price_list_id from SO_PRICE_LISTS.
freight_term	VARCHAR2(30)	No	Validation: freight_term is lookup code in lookup type FREIGHT_TERMS in so_lookups.
warehouse_id	NUMBER	No	Validation: Valid organization_id from ORG_ORGANIZATION_DEFINITIONS.
territory_id	NUMBER	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute25	VARCHAR2(150)	No	
tax_reference	VARCHAR2(50)	No	
sort_priority	NUMBER	No	
tax_code	VARCHAR2(50)	No	Validation: Must be a valid tax_code from AR_VAT_TAX.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
demand_class_code	VARCHAR2(30)	No	Validation: Validated against AR lookup type DEMAND_CLASS.
tax_header_level_flag	VARCHAR2(1)	No	
tax_rounding_rule	VARCHAR2(30)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
primary_salesrep_id	NUMBER	No	Validation: Valid salesrep_id from RA_SALESREPS.
finchrg_receivables_trx_id	NUMBER	No	Validation: Valid receivables trx_id from AR_RECEIVABLES_TRX.
dates_negative_tolerance	NUMBER	No	
dates_positive_tolerance	NUMBER	No	
date_type_preference	VARCHAR2(20)	No	Validation: Validated against OE lookup type REQUEST_DATE_TYPE.
over_shipment_tolerance	NUMBER	No	
under_shipment_tolerance	NUMBER	No	
item_cross_ref_pref	VARCHAR2(30)	No	Validation: Allowed values are INT, CUST, and cross_reference_type value from MTL_CROSS_REFERENCE_TYPES.
over_return_tolerance	NUMBER	No	
under_return_tolerance	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
ship_sets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• shipsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
arrivalsets_include_lines_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• arrivalsets_include_lines_flag is lookup code in lookup type YES/NO.</li> <li>• If ship_sets_include_lines_flag is set to Y, then arrivalsets_include_lines_flag is always N.</li> <li>• If arrivalsets_include_lines_flag is Y, then ship_sets_include_lines_flag is always N.</li> </ul>
sched_date_push_flag	VARCHAR2(1)	No	Validation: sched_date_push_flag is lookup code in lookup type YES/NO.
invoice_quantity_rule	VARCHAR2(30)	No	
pricing_event	VARCHAR2(30)	No	Comment: This attribute is no longer used.
gl_id_rec	NUMBER	No	Validation: gl_id_rec is valid gl field.
gl_id_rev	NUMBER	No	Validation: gl_id_rev is valid gl field.
gl_id_tax	NUMBER	No	Validation: gl_id_tax is valid gl field.
gl_id_freight	NUMBER	No	Validation: gl_id_freight is valid gl field.
gl_id_clearing	NUMBER	No	Validation: gl_id_clearing is valid gl field.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
gl_id_unbilled	NUMBER	No	Validation: gl_id_unbilled is valid gl field.
gl_id_unearned	NUMBER	No	Validation: gl_id_unearned is valid gl field.
gl_id_unpaid_rec	NUMBER	No	Validation: gl_id_unpaid_rec is valid gl field.
gl_id_remittance	NUMBER	No	Validation: gl_id_remittance is valid gl field.
gl_id_factor	NUMBER	No	Validation: gl_id_factor is valid gl field.
tax_classification	VARCHAR2(30)	No	
org_id	NUMBER	No	

#### Attributes of Embedded Entities

The Customer Account Site Use business structure has an embedded business structure and embedded granular entities. For the embedded structure and entity attribute information, see:

- Update Customer Profile Business Structure, page 24-209.
- Update Bank Account Use Entity, page 24-248
- Update Payment Method Entity, page 24-336

#### Get Customer Account Site Use Business Structure

As part of getting a business object, embedded Customer Account Site Use business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Account Site Use business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
site_use_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_site_id	NUMBER
site_use_code	VARCHAR2(30)
primary_flag	VARCHAR2(1)
location	VARCHAR2(40)
bill_to_site_use_id	NUMBER
sic_code	VARCHAR2(30)

---

<b>Attribute</b>	<b>Data Type</b>
payment_term_id	NUMBER
gsa_indicator	VARCHAR2(1)
ship_partial	VARCHAR2(1)
ship_via	VARCHAR2(30)
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 to attribute25	VARCHAR2(150)
tax_reference	VARCHAR2(50)
sort_priority	NUMBER
tax_code	VARCHAR2(50)
demand_class_code	VARCHAR2(30)
tax_header_level_flag	VARCHAR2(1)
tax_rounding_rule	VARCHAR2(30)
global_attribute1 to global_attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)

---

---

<b>Attribute</b>	<b>Data Type</b>
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)
pricing_event	VARCHAR2(30)
gl_id_rec	NUMBER
gl_id_rev	NUMBER
gl_id_tax	NUMBER
gl_id_freight	NUMBER
gl_id_clearing	NUMBER

---

Attribute	Data Type
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)
org_id	NUMBER

### Attributes of Embedded Entities

The Customer Account Site Use business structure has an embedded business structure and embedded granular entities. For the embedded structure and entity attribute information, see:

- Get Customer Profile Business Structure, page 24-216.
- Get Bank Account Use Entity, page 24-249
- Get Payment Method Entity, page 24-337

### Related Topics

[Business Structure Attributes, page 24-174](#)

## Customer Profile Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Customer Profile business structures.

### Create Customer Profile Business Structure

As part of creating a business object, you can provide data to create the Customer Profile business structure.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the

corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type CODE_STATUS.  Default: A.
collector_id	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	NUMBER	No	
credit_checking	VARCHAR2(1)	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	DATE	No	
tolerance	NUMBER	No	Validation: Must be between -100 and 100.  Default: Defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
discount_terms	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If discount_terms = <i>Y</i>, then discount_grace_days should be greater than or equal to 0.</li> <li>• If discount_terms = <i>N</i>, then discount_grace_days should be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>
dunning_letters	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Dunning letters is lookup code in lookup type YES/NO.</li> <li>• If dunning_letters = <i>N</i>, then dunning_letter_set_id must be null.</li> <li>• If dunning_letters = <i>Y</i>, then dunning_letters_set_id must have a value.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>

Attribute	Data Type	Required	Validation, Default, Comment
interest_charges	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If interest_charges = <math>Y</math>, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>• If interest_charges = <math>N</math>, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>
send_statements	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If send_statements = <math>Y</math>, then statement_cycle_id must have a value and credit_balance_statements must have value.</li> <li>• If send_statements = <math>N</math>, then statement_cycle_id must be null and credit_balance_statements must be <math>N</math>.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
credit_balance_statements	VARCHAR2(1)	No	<p>Validation: It is a lookup code in lookup type YES/NO.</p> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>
credit_hold	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If passed in, profile_class_id should be positive.</li> <li>• Foreign key to hz_cust_profile_classes and the corresponding profile class should be active.</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>
profile_class_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If passed in, profile_class_id should be positive.</li> <li>• Foreign key to hz_cust_profile_classes and the corresponding profile class should be active</li> </ul> <p>Default: defaulted to the corresponding value of DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0).</p>

Attribute	Data Type	Required	Validation, Default, Comment
site_use_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Foreign key to hz_cust_site_uses.</li> <li>• One site use can only have one profile.</li> <li>• The customer which the site_use_id belongs to should have the same id as cust_account_id in this profile. In this case cust_account_id is mandatory.</li> </ul>
credit_rating	VARCHAR2(30)	No	Validation: Credit Rating is lookup code in AR lookup type CREDIT_RATING.
risk_code	VARCHAR2(30)	No	Validation: Risk Code is lookup code in AR lookup type RISK_CODE.
standard_terms	NUMBER	No	Validation: Must be a valid term_id in RA_TERMS.
override_terms	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
dunning_letter_set_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If dunning_letters = Y, then dunning_letter_set_id must have a value.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>• Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_TS.</li> </ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
interest_period_days	NUMBER	No	Validation: Can be entered when interest_charges is Y and it is mandatory.
payment_grace_days	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	NUMBER	No	Validation: Can be populated only if discount_terms is Y.
statement_cycle_id	NUMBER	No	Validation: <ul style="list-style-type: none"> <li>• Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES.</li> <li>• Can be populated if send_statements is Y.</li> </ul>
account_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type ACCOUNT_STATUS.
percent_collectable	NUMBER	No	Validation: Must be between 0 and 100.
autocash_hierarchy_id	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2(150)	No	
auto_rec_incl_disputed_flg	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type YES/NO.
ag			Default: Defaulted to the corresponding value of 'DEFAULT PROFILE CLASS (PROFILE_CLASS_ID = 0)

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
tax_printing_option	VARCHAR2(30)	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	VARCHAR2(1)	No	Validation: charge_on_finance_charge_flag is lookup code in lookup type YES/NO.
grouping_rule_id	NUMBER	No	Validation: Must be a valid grouping_rule_id from RA_GROUPING_RULES.
clearing_days	NUMBER	No	Validation: Must be greater than zero.
jgzz_attribute_category	VARCHAR2(30)	No	
jgzz_attribute1 to jgzz_attribute15	VARCHAR2(150)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
cons_inv_flag	VARCHAR2(1)	No	Validation: cons_inv_flag is lookup code in lookup type YES/NO.
cons_inv_type	VARCHAR2(30)	No	
autocash_hierarchy_id_for_addr	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
lockbox_matching_option	VARCHAR2(30)	No	Validation: Validated against AR lookup type ARLPLB_MATCHING_OPTION.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
review_cycle	VARCHAR2(30)	No	Validation: Validated against AR lookup type PERIODIC REVIEW CYCLE.
last_credit_review_date	DATE	No	
credit_classification	VARCHAR2(30)	No	Validation: This is a lookup code in lookup_type 'CREDIT_CLASSIFICATION'. Default: <i>NULL</i>
cons_bill_level	VARCHAR2(30)	No	
late_charge_calculation_tr x	VARCHAR2(30)	No	
credit_items_flag	VARCHAR2(1)	No	
disputed_transactions_flag	VARCHAR2(1)	No	
late_charge_type	VARCHAR2(30)	No	
late_charge_term_id	NUMBER	No	
interest_calculation_perio d	VARCHAR2(30)	No	
hold_charged_invoices_flag	VARCHAR2(1)	No	
message_text_id	NUMBER	No	
multiple_interest_rates_flag	VARCHAR2(1)	No	
charge_begin_date	DATE)	No	

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit

outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Attributes of Embedded Entities

The Customer Profile business structure has an embedded granular entity. For the embedded entity's attribute information, see: Create Customer Profile Amount Entity, page 24-298.

#### Update Customer Profile Business Structure

As part of updating a business object, you can provide data to create or update the Customer Profile business structure.

#### Identification Attributes

The API procedure uses the attributes in this table to uniquely identify the correct Customer Profile business structure in the data model. You must provide sufficient information for these attributes to identify an existing Customer Profile business structure.

**Note:** Some attributes are used as a logical key to identify the business structure if the CUST\_ACCT\_PROFILE\_ID attribute value is not provided. If that value is provided, then these attributes are not used for identification, but are accepted and stored in the data model.

Attribute	Data Type	Logical Key
cust_acct_profile_id	NUMBER	No
profile_class_id	NUMBER	Yes
site_use_id	NUMBER	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type CODE_STATUS.  Default: A.
collector_id	NUMBER	No	Validation: foreign key to ar_collectors.
credit_analyst_id	NUMBER	No	
credit_checking	VARCHAR2(1)	No	Validation: It is lookup code in lookup type YES/NO.
next_credit_review_date	DATE	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
tolerance	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Tolerance cannot be updated to null.</li> <li>• Must be between -100 and 100.</li> </ul>
discount_terms	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If discount_terms = Y, then discount_grace_days should be greater than or equal to 0.</li> <li>• If discount_terms = N, then discount_grace_days should be null.</li> </ul>
dunning_letters	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Dunning letters is lookup code in lookup type YES/NO.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> </ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
interest_charges	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If interest_charges = Y, then charge_on_finance_charge_flag must have a value and interest_period_days must have a value greater than 0.</li> <li>• If interest_charges = N, then charge_on_finance_charge_flag must be null and interest_period_days must be null.</li> </ul>
send_statements	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is lookup code in lookup YES/NO.</li> <li>• If send_statements = Y, then statement_cycle_id must have a value and credit_balance_statements must have value.</li> <li>• If send_statements = N, then statement_cycle_id must be null and credit_balance_statements must be N.</li> </ul>
credit_balance_statement_s	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• It is a lookup code in lookup type YES/NO.</li> <li>• Can be populated if send_statements is Y.</li> </ul>
credit_hold	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type YES/NO.

Attribute	Data Type	Required	Validation, Default, Comment
profile_class_id	NUMBER	No	Validation: Cannot be set to null.
site_use_id	NUMBER	No	Validation: Cannot be updated.
credit_rating	VARCHAR2(30)	No	Validation: Credit Rating is lookup code in lookup type CREDIT_RATING.
risk_code	VARCHAR2(30)	No	Validation: Risk Code is lookup code in AR lookup type RISK_CODE.
standard_terms	NUMBER	No	Validation: Must be a valid term_id in RA_TERMS.
override_terms	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
dunning_letter_set_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• If dunning_letters = Y, then dunning_letter_set_id must have a value.</li> <li>• If dunning_letters = N, then dunning_letter_set_id must be null.</li> <li>• Must be a valid dunning_letter_set_id from AR_DUNNING_LETTER_SETS.</li> </ul>
interest_period_days	NUMBER	No	Validation: Must be entered when interest_charges is Y and it is mandatory.
payment_grace_days	NUMBER	No	Validation: Must be greater than zero.
discount_grace_days	NUMBER	No	Validation: Can only be populated only if discount_terms is Y.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
statement_cycle_id	NUMBER	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Must be a valid statement_cycle_id from AR_STATEMENT_CYCLES.</li> <li>• Can be populated if send_statements is Y.</li> </ul>
account_status	VARCHAR2(30)	No	Validation: Validated against AR lookup type ACCOUNT_STATUS.
percent_collectable	NUMBER	No	Validation: Must be between 0 and 100.
autocash_hierarchy_id	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2(150)	No	
auto_rec_incl_disputed_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot be set to null.</li> <li>• It is a lookup code in lookup type YES/NO.</li> </ul>
tax_printing_option	VARCHAR2(30)	No	Validation: Validated against AR lookup type TAX_PRINTING_OPTION.
charge_on_finance_charge_flag	VARCHAR2(1)	No	Validation: It is lookup code in lookup type YES/NO.
grouping_rule_id	NUMBER	No	Validation: Must be a valid grouping_rule_id from RA_GROUPING_RULES.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
clearing_days	NUMBER	No	Validation: Must be greater than zero.
jgzz_attribute_category	VARCHAR2(30)	No	
jgzz_attribute1 to jgzz_attribute15	VARCHAR2(150)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
cons_inv_flag	VARCHAR2(1)	No	Validation: It is a lookup code in lookup type YES/NO
cons_inv_type	VARCHAR2(30)	No	
autocash_hierarchy_id_for _adr	NUMBER	No	Validation: Must be a valid autocash_hierarchy_id from AR_AUTOCASH_HIERARCHIES.
lockbox_matching_option	VARCHAR2(30)	No	Validation: Validated against AR lookup type ARLPLB_MATCHING_OPTION.
review_cycle	VARCHAR2(30)	No	
last_credit_review_date	DATE	No	
credit_classification	VARCHAR2(30)	No	Validation: This is a lookup code in lookup_type 'CREDIT_CLASSIFICATION'.
cons_bill_level	VARCHAR2(30)	No	
late_charge_calculation_tx	VARCHAR2(30)	No	
credit_items_flag	VARCHAR2(1)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
disputed_transactions_flag	VARCHAR2(1)	No	
late_charge_type	VARCHAR2(30)	No	
late_charge_term_id	NUMBER	No	
interest_calculation_period	VARCHAR2(30)	No	
hold_charged_invoices_flag	VARCHAR2(1)	No	
message_text_id	NUMBER	No	
multiple_interest_rates_flag	VARCHAR2(1)	No	
charge_begin_date	DATE)	No	

#### Attributes of Embedded Entities

The Customer Profile business structure has an embedded granular entity. For the embedded entity's attribute information, see: Update Customer Profile Amount Entity, page 24-300.

#### Get Customer Profile Business Structure

As part of getting a business object, embedded Customer Profile business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Profile business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE

- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
cust_acct_profile_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
collector_id	NUMBER
credit_analyst_id	NUMBER
credit_checking	VARCHAR2(1)
next_credit_review_date	DATE
tolerance	NUMBER

---

<b>Attribute</b>	<b>Data Type</b>
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

---

<b>Attribute</b>	<b>Data Type</b>
attribute_category	VARCHAR2(30)
attribute1..15	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..15	VARCHAR2(150)
global_attribute1..20	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)
review_cycle	VARCHAR2(30)
last_credit_review_date	DATE
credit_classification	VARCHAR2(30)
cons_bill_level	VARCHAR2(30)
late_charge_calculation_trx	VARCHAR2(30)

<b>Attribute</b>	<b>Data Type</b>
credit_items_flag	VARCHAR2(1)
disputed_transactions_flag	VARCHAR2(1)
late_charge_type	VARCHAR2(30)
late_charge_term_id	NUMBER
interest_calculation_period	VARCHAR2(30)
hold_charged_invoices_flag	VARCHAR2(1)
message_text_id	NUMBER
multiple_interest_rates_flag	VARCHAR2(1)
charge_begin_date	DATE)

#### Attributes of Embedded Entities

The Customer Profile business structure has an embedded granular entity. For the embedded entity's attribute information, see: Get Customer Profile Amount Entity, page 24-303.

#### Related Topics

[Business Structure Attributes, page 24-174](#)

## Employment History Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Employment History business structure.

#### Create Employment History Business Structure

As part of creating a business object, you can provide data to create the Employment History business structure.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.
begin_date	DATE	No	
end_date	DATE	No	Validation: If both begin_date and end_date are passed, then end_date must be greater than or equal to the begin_date.
employment_type_code	VARCHAR2(30)	No	Validation: Validated against the HZ_EMPLOYMENT_TYPE lookup type.
employed_as_title_code	VARCHAR2(30)	No	Validation: Validated against the RESPONSIBILITY lookup type.
employed_as_title	VARCHAR2(60)	No	Validation: if employed_as_title_code is supplied, then employed_as_title must be null.
employed_by_name_company	VARCHAR2(60)	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	NUMBER	No	Validation: Must exist in the HZ_PARTIES table.
employed_by_division_name	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
supervisor_name	VARCHAR2(60)	No	
branch	VARCHAR2(80)	No	
military_rank	VARCHAR2(240)	No	
served	VARCHAR2(240)	No	
station	VARCHAR2(240)	No	
responsibility	VARCHAR2(240)	No	
weekly_work_hours	NUMBER	No	Validation: If passed, then must be greater than zero and less than or equal to 168.
reason_for_leaving	VARCHAR2(240)	No	
faculty_position_flag	VARCHAR2(1)	Yes	Validation: validated against AR lookup type YES/NO. Default: N.
tenure_code	VARCHAR2(30)	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, else must be null. Validated against AR lookup type HZ_TENURE_CODE
fraction_of_tenure	NUMBER	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, else must be null. If passed, must be between 0 and 100 inclusive.
comments	VARCHAR2(2000)	No	

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Attributes of Embedded Entities

The Employment History business structure has an embedded granular entity. For the embedded entity's attribute information, see: Create Work Class Entity, page 24-359.

#### Update Employment History Business Structure

As part of updating a business object, you can provide data to create or update the Employment History business structure.

#### Identification Attributes

The API procedure uses the attributes in this table to uniquely identify the correct Employment History business structure in the data model. You must provide sufficient information for these attributes to identify an existing Employment History business structure.

**Note:** Some attributes are used as a logical key to identify the business structure if the EMPLOYMENT\_HISTORY\_ID attribute value is not provided. If that value is provided, then these attributes are not used for identification, but are accepted and stored in the data model.

Attribute	Data Type	Logical Key
employment_history_id	NUMBER	No
begin_date	DATE	Yes
employed_by_name_company	VARCHAR2(60)	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit

outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.
begin_date	DATE	No	
end_date	DATE	No	Validation: If both begin_date and end_date are passed, then end_date must be greater than or equal to the begin_date.
employment_type_code	VARCHAR2(30)	No	Validation: Validated against the HZ_EM PLOYMENT_TYPE lookup type.
employed_as_title_code	VARCHAR2(30)	No	Validation: Validated against the RESPONSIBILITY lookup type.
employed_as_title	VARCHAR2(60)	No	Validation: if employed_as_title_code is supplied, then employed_as_title must be null.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
employed_by_name_company	VARCHAR2(60)	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	NUMBER	No	Validation: Must exist in the HZ_PARTIES.
employed_by_division_name	VARCHAR2(60)	No	
supervisor_name	VARCHAR2(60)	No	
branch	VARCHAR2(80)	No	
military_rank	VARCHAR2(240)	No	
served	VARCHAR2(240)	No	
station	VARCHAR2(240)	No	
responsibility	VARCHAR2(240)	No	
weekly_work_hours	NUMBER	No	Validation: If passed, then must be greater than zero and less than or equal to 168.
reason_for_leaving	VARCHAR2(240)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
faculty_position_flag	VARCHAR2(1)	Yes	Validation: validated against AR lookup type YES/NO.  Default: N.
tenure_code	VARCHAR2(30)	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, else must be null. Validated against AR lookup type HZ_TENURE_CODE
fraction_of_tenure	NUMBER	No	Validation: Must only be passed if FACULTY_POSITION_IND = Y, else must be null. If passed, must be between 0 and 100 inclusive.
comments	VARCHAR2 (2000)	No	

#### Attributes of Embedded Entities

The Employment History business structure has an embedded granular entity. For the embedded entity's attribute information, see: Update Work Class Entity, page 24-360.

#### Get Employment History Business Structure

As part of getting a business object, embedded Employment History business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Employment History business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**

- *Event Independent:* Null
- *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
employment_history_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_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

<b>Attribute</b>	<b>Data Type</b>
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_flag	VARCHAR2(1)
tenure_code	VARCHAR2(30)
fraction_of_tenure	NUMBER
comments	VARCHAR2(2000)

#### Attributes of Embedded Entities

The Employment History business structure has an embedded granular entity. For the embedded entity's attribute information, see: Get Work Class Entity, page 24-361.

#### Related Topics

[Business Structure Attributes, page 24-174](#)

#### Financial Report Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Financial Report business structure.

## Create Financial Report Business Structure

As part of creating a business object, you can provide data to create the Financial Report business structure.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.
status	VARCHAR2(1)	No	Validation: Validated against AR lookup type REGISTRY_STATUS.
type_of_financial_report	VARCHAR2(60)	No	
document_reference	VARCHAR2(150)	No	
date_report_issued	DATE	No	
issued_period	VARCHAR2(60)	No	Validation: Either issued_period or report_start_date must be provided, but not both.
report_start_date	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	DATE	No	Validation: Must be provided if report_start_date is provided, else must be null.  If provided, then it must be greater than or equal to report_start_date.
requiring_authority	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
audit_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
estimated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
fiscal_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
final_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
forecast_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
opening_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
proforma_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
qualified_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
restated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
signed_by_principals_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
trial_balance_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
unbalanced_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit

outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Attributes of Embedded Entities

The Financial Report business structure has an embedded granular entity. For the embedded entity's attribute information, see: Create Financial Number Entity, page 24-315.

#### Update Financial Report Business Structure

As part of updating a business object, you can provide data to create or update the Financial Report business structure.

#### Identification Attributes

The API procedure uses the attributes in this table to uniquely identify the correct Financial Report business structure in the data model. You must provide sufficient information for these attributes to identify an existing Financial Report business structure.

**Note:** Some attributes are used as a logical key to identify the business structure if the FINANCIAL\_REPORT\_ID attribute value is not provided. If that value is provided, then these attributes are not used for identification, but are accepted and stored in the data model.

Attribute	Data Type	Logical Key
financial_report_id	NUMBER	No
type_of_financial_report	VARCHAR2(60)	Yes
document_reference	VARCHAR2(150)	Yes

Attribute	Data Type	Logical Key
date_report_issued	DATE	Yes
issued_period	VARCHAR2(60)	Yes
report_start_date	DATE	Yes
report_end_date	DATE	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	Yes	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.
status	VARCHAR2(1)	No	Validation: Validated against AR lookup type REGISTRY_STATUS.
type_of_financial_report	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
document_reference	VARCHAR2(150)	No	
date_report_issued	DATE	No	
issued_period	VARCHAR2(60)	No	Validation: Either issued_period or report_start_date must be provided, but not both.
report_start_date	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	DATE	No	Validation: Must be provided if report_start_date is provided, else must be null.  If provided, then it must be greater than or equal to report_start_date.
requiring_authority	VARCHAR2(60)	No	
audit_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
consolidated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
estimated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
fiscal_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
final_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
forecast_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
opening_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
proforma_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
qualified_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
restated_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
signed_by_principals_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
trial_balance_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.
unbalanced_ind	VARCHAR2(30)	No	Validation: Validated against AR lookup type YES/NO.

#### Attributes of Embedded Entities

The Financial Report business structure has an embedded granular entity. For the embedded entity's attribute information, see: Update Financial Number Entity, page 24-316.

#### Get Financial Report Business Structure

As part of getting a business object, embedded Financial Report business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Financial Report business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE

- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
financial_report_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
organization_id	NUMBER
type_of_financial_report	VARCHAR2(60)
document_reference	VARCHAR2(150)
date_report_issued	DATE
issued_period	VARCHAR2(60)

<b>Attribute</b>	<b>Data Type</b>
report_start_date	DATE
report_end_date	DATE
requiring_authority	VARCHAR2(60)
audit_ind	VARCHAR2(30)
consolidated_ind	VARCHAR2(30)
estimated_ind	VARCHAR2(30)
fiscal_ind	VARCHAR2(30)
final_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)

### Attributes of Embedded Entities

The Financial Report business structure has an embedded granular entity. For the embedded entity's attribute information, see: [Get Financial Number Entity](#), page 24-317.

### Related Topics

[Business Structure Attributes](#), page 24-174

## Location Business Structure Attributes

This section provides information about what seeded business object API procedures do with the attributes of embedded Location business structure.

### Create Location Business Structure

As part of creating a business object, you can provide data to create the Location business structure.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y. Default: <i>USER_ENTERED</i> .
country	VARCHAR2(60)	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Foreign Key to fnd_territories. territory_code</li></ul>
address1	VARCHAR2(240)	Yes	Validation: Mandatory attribute.
address2	VARCHAR2(240)	No	
address3	VARCHAR2(240)	No	
address4	VARCHAR2(240)	No	
city	VARCHAR2(60)	No	
postal_code	VARCHAR2(60)	No	
state	VARCHAR2(60)	No	
province	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
county	VARCHAR2(60)	No	
address_key	VARCHAR2(500)	No	
address_style	VARCHAR2(30)	No	
validated_flag	VARCHAR2(1)	No	
address_lines_phonetic	VARCHAR2(560)	No	
postal_plus4_code	VARCHAR2(10)	No	
position	VARCHAR2(50)	No	
location_directions	VARCHAR2(640)	No	
address_effective_date	DATE	No	
address_expiration_date	DATE	No	
clli_code	VARCHAR2(60)	No	
language	VARCHAR2(4)	No	Validation: Foreign key to fnd_languages.language_code (installed).
short_description	VARCHAR2(240)	No	
description	VARCHAR2(2000)	No	
loc_hierarchy_id	NUMBER	No	
sales_tax_geocode	VARCHAR2(30)	No	
sales_tax_inside_city_limits	VARCHAR2(30)	No	
fa_location_id	NUMBER	No	
attribute_category	VARCHAR2(30)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute1 to attribute20	VARCHAR2(150)	No	
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezones.timezone_id.
delivery_point_code	VARCHAR2(50)	No	
geometry_status_code	VARCHAR2(30)	No	
geometry	mdsys. sdo_geometry	No	

### Mapped Attributes

This procedure creates a mapping between the internally maintained PERSON\_ID and the attributes in this table. The created Source System Mapping is stored in the TCA SSM tables.

<b>Attribute</b>	<b>Data Type</b>
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### **Attributes of Embedded Entities**

The Location business structure has an embedded granular entity. For the embedded entity's attribute information, see: Create Extension Attribute Entity, page 24-311.

### **Update Location Business Structure**

As part of updating a business object, you can provide data to create or update the Location business structure.

### **Identification Attributes**

The API procedure uses the attributes in this table to uniquely identify the correct Location business structure in the data model. You must provide sufficient information for these attributes to identify an existing Location business structure.

Attribute	Data Type	Logical Key
location_id	NUMBER	No
orig_system	VARCHAR2(30)	No
orig_system_reference	VARCHAR2(255)	No

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### **Accepted Attributes**

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validation: Cannot be updated.
country	VARCHAR2(60)	No	Validation: Foreign key to fnd_territories.territory_code.
address1	VARCHAR2(240)	No	Validation: Cannot be set to null during update.
address2	VARCHAR2(240)	No	
address3	VARCHAR2(240)	No	
address4	VARCHAR2(240)	No	
city	VARCHAR2(60)	No	
postal_code	VARCHAR2(60)	No	
state	VARCHAR2(60)	No	
province	VARCHAR2(60)	No	
county	VARCHAR2(60)	No	
address_key	VARCHAR2(500)	No	
address_style	VARCHAR2(30)	No	
validated_flag	VARCHAR2(1)	No	
address_lines_phonetic	VARCHAR2(560)	No	
postal_plus4_code	VARCHAR2(10)	No	
position	VARCHAR2(50)	No	
location_directions	VARCHAR2(640)	No	
address_effective_date	DATE	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
address_expiration_date	DATE	No	
clli_code	VARCHAR2(60)	No	
language	VARCHAR2(4)	No	Validation: Foreign key to fnd_languages.language_code (installed).
short_description	VARCHAR2(240)	No	
description	VARCHAR2 (2000)	No	
loc_hierarchy_id	NUMBER	No	
sales_tax_geocode	VARCHAR2(30)	No	
sales_tax_inside_city_limi ts	VARCHAR2(30)	No	
fa_location_id	NUMBER	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
timezone_id	NUMBER	No	Validation: Foreign key to hz_timezones.timezone_id.
delivery_point_code	VARCHAR2(50)	No	
geometry_status_code	VARCHAR2(30)	No	
geometry	mdsys. sdo_geometry	No	

#### Attributes of Embedded Entities

The Location business structure has an embedded granular entity. For the embedded entity's attribute information, see: Update Extension Attribute Entity, page 24-312.

## Get Location Business Structure

As part of getting a business object, embedded Location business structure attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Location business structure. This attribute is only used when the business structure is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, UNCHANGED, or CHILD\_UPDATED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
location_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)

---

<b>Attribute</b>	<b>Data Type</b>
actual_content_source	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)
postal_plus4_code	VARCHAR2(10)
position	VARCHAR2(50)
location_directions	VARCHAR2(640)
address_effective_date	DATE
address_expiration_date	DATE

---

Attribute	Data Type
clli_code	VARCHAR2(60)
language	VARCHAR2(4)
short_description	VARCHAR2(240)
description	VARCHAR2(2000)
loc_hierarchy_id	NUMBER
sales_tax_geocode	VARCHAR2(30)
sales_tax_inside_city_limits	VARCHAR2(30)
fa_location_id	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
timezone_id	NUMBER
delivery_point_code	VARCHAR2(50)
geometry_status_code	VARCHAR2(30)
geometry	mdsys.sdo_geometry

#### Attributes of Embedded Entities

The Location business structure has an embedded granular entity. For the embedded entity's attribute information, see: Get Extension Attribute Entity, page 24-313.

#### Related Topics

[Business Structure Attributes, page 24-174](#)

## Granular Entity Attributes

Many seeded business objects have embedded granular entities. When you use the seeded business object APIs, the embedded entities are correspondingly created,

updated, or retrieved.

These sections provide information about what seeded business object API procedures do with the attributes of embedded granular entities.

- Bank Account Use Entity Attributes, page 24-247
- Certification Entity Attributes, page 24-251
- Citizenship Entity Attributes, page 24-254
- Classification Entity Attributes, page 24-259
- Contact Preference Entity Attributes, page 24-264
- Credit Rating Entity Attributes, page 24-276
- Customer Account Relationship Entity Attributes, page 24-293
- Customer Profile Amount Entity Attributes, page 24-298
- Education Entity Attributes, page 24-306
- Extension Attribute Entity Attributes, page 24-311
- Financial Number Entity Attributes, page 24-315
- Financial Profile Entity Attributes, page 24-318
- Organization Contact Role Entity Attributes, page 24-323
- Party Preference Entity Attributes, page 24-327
- Party Site Use Entity Attributes, page 24-331
- Payment Method Entity Attributes, page 24-335
- Person Interest Entity Attributes, page 24-339
- Person Language Entity Attributes, page 24-343
- Relationship Entity Attributes, page 24-348
- Role Responsibility Entity Attributes, page 24-354
- Source System Information Entity Attributes, page 24-358
- Work Class Entity Attributes, page 24-359

## Related Topics

[Business Object API Attributes Information Overview, page 24-2](#)

[Business Object APIs Overview, page 22-1](#)

## Bank Account Use Entity Attributes

### Create Bank Account Use Entity

As part of creating a business object, you can provide entity data to create the Bank Account Use entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
org_type	VARCHAR2(30)
org_id	NUMBER
instrument_type	VARCHAR2(30)
instrument_id	NUMBER
priority	NUMBER
start_date	DATE
end_date	DATE

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Update Bank Account Use Entity

As part of updating a business object, you can provide entity data to create or update the Bank Account Use entity.

### Identification Attributes

The API procedure uses the BANK\_ACCT\_USE\_ID attribute to uniquely identify the correct Bank Account Use entity in the data model. You must provide sufficient information for this attribute to identify an existing Bank Account Use entity.

- **Attribute:** BANK\_ACCT\_USE\_ID
- **Data Type:** NUMBER

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>
org_type	VARCHAR2(30)

---

Attribute	Data Type
org_id	NUMBER
instrument_type	VARCHAR2(30)
instrument_id	NUMBER
priority	NUMBER
start_date	DATE
end_date	DATE

## Get Bank Account Use Entity

As part of getting a business object, embedded Bank Account Use entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Bank Account Use entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
bank_acct_use_id	NUMBER
payment_function	VARCHAR2(30)
party_id	NUMBER
org_type	VARCHAR2(30)
org_id	NUMBER
cust_acct_id	NUMBER
site_use_id	NUMBER
instrument_type	VARCHAR2(30)
instrument_id	NUMBER
priority	NUMBER
start_date	DATE
end_date	DATE
created_by_name	VARCHAR2(100)
creation_date	DATE
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Certification Entity Attributes

### Create Certification Entity

As part of creating a business object, you can provide entity data to create the Certification entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
status	VARCHAR2(1)
certification_name	VARCHAR2(240)
current_status	VARCHAR2(30)
expires_on_date	DATE
grade	VARCHAR2(30)
issued_by_authority	VARCHAR2(60)
issued_on_date	DATE

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Certification Entity

As part of updating a business object, you can provide entity data to create or update the Certification entity.

### Identification Attributes

The API procedure uses the CERTIFICATION\_ID attribute to uniquely identify the correct Certification entity in the data model. You must provide sufficient information for this attribute to identify an existing Certification entity.

Attribute	Data Type	Logical Key
certification_id	NUMBER	No
certification_name	VARCHAR2(240)	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
status	VARCHAR2(1)

Attribute	Data Type
certification_name	VARCHAR2(240)
current_status	VARCHAR2(30)
expires_on_date	DATE
grade	VARCHAR2(30)
issued_by_authority	VARCHAR2(60)
issued_on_date	DATE

## Get Certification Entity

As part of getting a business object, embedded Certification entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Certification entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
certification_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
certification_name	VARCHAR2(240)
current_status	VARCHAR2(30)
expires_on_date	DATE
grade	VARCHAR2(30)
issued_by_authority	VARCHAR2(60)
issued_on_date	DATE

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Citizenship Entity Attributes

### Create Citizenship Entity

As part of creating a business object, you can provide entity data to create the Citizenship entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.
birth_or_selected	VARCHAR2(30)	No	Validation: Validated against the HZ_CITIZENSHIP_ACQUISITION lookup type.
country_code	VARCHAR2(2)	Yes	Validation: Must exist in FND_TERRITORIES.
date_recognized	DATE	No	
date_disowned	DATE	No	
end_date	DATE	No	
document_type	VARCHAR2(30)	No	
document_reference	VARCHAR2(60)	No	

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Citizenship Entity

As part of updating a business object, you can provide entity data to create or update the Citizenship entity.

### Identification Attributes

The API procedure uses the CITIZENSHIP\_ID attribute to uniquely identify the correct Citizenship entity in the data model. You must provide sufficient information for this attribute to identify an existing Citizenship entity.

Attribute	Data Type	Logical Key
citizenship_id	NUMBER	No
country_code	VARCHAR2(2)	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.

Attribute	Data Type	Required	Validation, Default, Comment
birth_or_selected	VARCHAR2(30)	No	Validation: Validated against the HZ_CITIZENSHIP_ACQUISITION lookup type.
country_code	VARCHAR2(2)	Yes	Validation: Must exist in FND_TERRITORIES.
date_recognized	DATE	No	
date_disowned	DATE	No	
end_date	DATE	No	
document_type	VARCHAR2(30)	No	
document_reference	VARCHAR2(60)	No	

## Get Citizenship Entity

As part of getting a business object, embedded Citizenship entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Citizenship entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
citizenship_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_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)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Classification Entity Attributes

### Create Classification Entity

As part of creating a business object, you can provide entity data to create the Classification entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS. orig_system with sst_flag value of Y.
status	VARCHAR2(1)	No	Validation: Validated against AR_LOOKUP type CODE_STATUS.
class_category	VARCHAR2(30)	No	Validation: Validated against HZ_CLASS_CATEGORIES.
class_code	VARCHAR2(30)	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.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
primary_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Validated against FND lookup values where lookup type YES/NO.</li> <li>• An owner_table_id can only have one primary (PRIMARY_FLAG = Y) assignment to a class_code of one class_category for one actual_content_source at one time.</li> </ul>
start_date_active	DATE	No	
end_date_active	DATE	No	Validation: must be null or greater than start_date_active.
rank	NUMBER	No	Comment: Provides the ability to rank classes if multiple classes have been assigned to a party.

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Update Classification Entity

As part of updating a business object, you can provide entity data to create or update

the Classification entity.

#### Identification Attributes

The API procedure uses the CODE\_ASSIGNMENT\_ID attribute to uniquely identify the correct Classification entity in the data model. You must provide sufficient information for this attribute to identify an existing Classification entity.

Attribute	Data Type	Logical Key
code_assignment_id	NUMBER	No
class_category	VARCHAR2(30)	Yes
class_code	VARCHAR2(30)	Yes
start_date_active	DATE	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)		

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
status	VARCHAR2(1)	No	Validation: Validated against AR_LOOKUP type CODE_STATUS.
class_category	VARCHAR2(30)	Yes	
class_code	VARCHAR2(30)	No	
primary_flag	VARCHAR2(1)	No	
start_date_active	DATE	No	
end_date_active	DATE	No	Comment: Cannot create classification with time period that overlaps an existing classification.
			Validation: Done in HZ_CLASS_VALIDATE_V2.validate_code_assignment.
rank	NUMBER	No	Comment: Provides the ability to rank classes if multiple classes have been assigned to a party.

## Get Classification Entity

As part of getting a business object, embedded Classification entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Classification entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**

- *Event Independent:* Null
- *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
code_assignment_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
class_category	VARCHAR2(30)
class_code	VARCHAR2(30)
primary_flagcontent_source_type	VARCHAR2(1)
start_date_active	DATE
end_date_active	DATE
rank	NUMBER

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Contact Preference Entity Attributes

### Create Contact Preference Entity

As part of creating a business object, you can provide entity data to create the Contact Preference entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"><li>Should be validated against the CODE_STATUS lookup type.</li><li>The PREFERENCE_END_DATE attribute should be set to the sysdate when STATUS has a value other than A.</li></ul>
contact_type	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"><li>Mandatory attribute.</li><li>Cannot be updated.</li><li>contact_type is lookup code in lookup type CONTACT_TYPE.</li></ul>

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_code	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Should be validated against the PREFERENCE_CODE lookup type.</li> </ul>
preference_topic_type	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Should be validated against the PREFERENCE_TOPIC_TYPE lookup type.</li> <li>• The lookup contains the following lookup_codes against which the PREFERENCE_TOPIC_TYPE will be validated:</li> </ul> <p>TABLES:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES,</li> <li>• AS_INTEREST_TYPES_B,</li> <li>• AS_INTEREST_CODES_B</li> <li>• LOOKUP_TYPE: CONTACT_USAGE</li> </ul>

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_topic_type_id	NUMBER	No	<p>Validation:</p> <p>Valid if the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES,</li> <li>• AS_INTEREST_TYPES_B,</li> <li>• AS_INTEREST_CODES_B</li> </ul> <p>The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected PREFERENCE_TOPIC_TYPE.</p>
preference_topic_type_code	VARCHAR2(30)	No	<p>Validation:</p> <p>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.</p> <p>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.</p>
preference_start_date	DATE	Yes	Validation: Mandatory attribute.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_end_date	DATE	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Cannot be updated to a day before the sysdate.</li> <li>• PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE.</li> <li>• If the STATUS column in the HZ_CONTACT_POINTS table is set to a value other than A for Active (such as I for Inactive, M for Merged, and D for Deleted), then PREFERENCE_END_DATE is not passed in these situations, it should default to the system date. If a value other than the system date is passed, it should fail.</li> </ul>
preference_start_time_hr	NUMBER	No	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>
preference_end_time_hr	NUMBER	No	<p>Validation:</p> <p>In 0 to 24 hour format.</p> <p>PREFERENCE_END_TIME_HR: PREFERENCE_END_TIME_MI should be greater than or equal to PREFERENCE_START_TIME_MI: PREFERENCE_START_TIME_MI</p>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_start_time_mi	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	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	NUMBER	No	Validation: none.
max_no_of_interact_uom_code	VARCHAR2(30)	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	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Should be validated against the REQUESTED_BY lookup type.</li> </ul>
reason_code	VARCHAR2(30)	No	Validation: Validated against the REASON_CODE lookup type.

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Contact Preference Entity

As part of updating a business object, you can provide entity data to create or update the Contact Preference entity.

### Identification Attributes

The API procedure uses the CONTACT\_PREFERENCE\_ID attribute to uniquely identify the correct Contact Preference entity in the data model. You must provide sufficient information for this attribute to identify an existing Contact Preference entity.

Attribute	Data Type	Logical Key
contact_preference_id	NUMBER	No
contact_type	VARCHAR2(30)	Yes
preference_topic_type	VARCHAR2(30)	Yes
preference_topic_type_id	NUMBER	Yes
preference_topic_type_code	VARCHAR2(30)	Yes
preference_start_date	DATE	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return

object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"><li>• Should be validated against the CODE_STATUS lookup type.</li><li>• The PREFERENCE_END_DATE attribute should be set to the sysdate when STATUS has a value other than A.</li><li>• Cannot be set to NULL during update.</li></ul>
contact_type	VARCHAR2(30)	No	Validation: Cannot be updated.
preference_code	VARCHAR2(30)	No	Validation: Should be validated against the PREFERENCE_CODE AR lookup type.

Attribute	Data Type	Required	Validation, Default, Comment
preference_topic_type	VARCHAR2(30)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Should be validated against the PREFERENCE_TOPIC_TYPE lookup type.</li> <li>• The lookup contains the following lookup_codes against which the PREFERENCE_TOPIC_TYPE will be validated:</li> </ul> <p>TABLES:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES,</li> <li>• AS_INTEREST_TYPES_B,</li> <li>• AS_INTEREST_CODES_B</li> <li>• LOOKUP_TYPE</li> </ul>
preference_topic_type_id	NUMBER	No	<p>Validation:</p> <p>If the value in the PREFERENCE_TOPIC_TYPE attribute is one of these values:</p> <ul style="list-style-type: none"> <li>• AMS_SOURCE_CODES,</li> <li>• AS_INTEREST_TYPES_B,</li> <li>• AS_INTEREST_CODES_B</li> </ul> <p>The PREFERENCE_TOPIC_TYPE attribute is the foreign key of table selected in the attribute PREFERENCE_TOPIC_TYPE.</p>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_topic_type_code	VARCHAR2(30)	No	<p>Validation:</p> <p>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.</p> <p>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.</p>
preference_start_date	DATE	Yes	<p>Validation:</p> <p>The value of the PREFERENCE_END_DATE attribute should be greater than or equal to the value of the PREFERENCE_START_DATE attribute.</p>
preference_end_date	DATE	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>• PREFERENCE_END_DATE cannot be updated to a day before the sysdate.</li> <li>• PREFERENCE_END_DATE should be greater than or equal to PREFERENCE_START_DATE.</li> </ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
preference_start_time_hr	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	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	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	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	NUMBER	No	Validation: none.

Attribute	Data Type	Required	Validation, Default, Comment
max_no_of_interact_uom_code	VARCHAR2(30)	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	VARCHAR2(30)	Yes	Validation: REQUESTED_BY should be validated against the REQUESTED_BY AR lookup type
reason_code	VARCHAR2(30)	No	Validation: Should be validated against the REASON_CODE lookup type.

## Get Contact Preference Entity

As part of getting a business object, embedded Contact Preference entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Contact Preference entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
contact_preference_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_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

---

<b>Attribute</b>	<b>Data Type</b>
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)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Credit Rating Entity Attributes

### Create Credit Rating Entity

As part of creating a business object, you can provide entity data to create the Credit Rating entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validate: HZ_MIXNM.Utility.ValidateContentSource.
status	VARCHAR2(1)	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS.
description	VARCHAR2 (2000)	No	
rating	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
rated_as_of_date	DATE	No	
rating_organization	VARCHAR2(240)	No	
comments	VARCHAR2(240)	No	
det_history_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
fincl_embt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
criminal_proceeding_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
claims_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
secured_flgng_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
fincl_lgl_event_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
disaster_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
oprg_spec_evnt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
other_spec_evnt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
avg_high_credit	NUMBER	No	
credit_score	VARCHAR2(30)	No	
credit_score_age	NUMBER	No	
credit_score_class	NUMBER	No	
credit_score_commentary	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTAR Y.
credit_score_commentary 2 to credit_score_commentary 10	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTAR Y.
credit_score_date	DATE	No	
credit_score_incd_default	NUMBER	No	
credit_score_natl_percentile	NUMBER	No	
failure_score	VARCHAR2(30)	No	
failure_score_age	NUMBER	No	
failure_score_class	NUMBER	No	
failure_score_commentary	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTA RY.
failure_score_commentary y2 to failure_score_commentary y10	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTA RY.
failure_score_date	DATE	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
failure_score_incd_default	NUMBER	No	
failure_score_natnl_perce_ntile	NUMBER	No	
failure_score_override_code	VARCHAR2(30)	No	
global_failure_score	VARCHAR2(30)	No	
debarment_ind	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup YES/NO.
debarments_count	NUMBER	No	
debarments_date	DATE	No	
high_credit	NUMBER	No	
maximum_credit_currency_code	VARCHAR2(240)	No	Validation: Foreign key to fnd_currencies.currency_code.
maximum_credit_rcmd	NUMBER	No	
paydex_norm	VARCHAR2(3)	No	
paydex_score	VARCHAR2(3)	No	
paydex_three_months ago	VARCHAR2(3)	No	
credit_score_override_code	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE.
cr_scr_clas_expl	VARCHAR2(30)	No	
low_rng_delq_scr	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
high_rng_delq_scr	NUMBER	No	
delq_pmt_rng_prcnt	NUMBER	No	
delq_pmt_pctg_for_all_firms	NUMBER	No	
num_trade_experiences	NUMBER	No	
paydex_firm_days	VARCHAR2(15)	No	
paydex_firm_comment	VARCHAR2(60)	No	
paydex_industry_days	VARCHAR2(15)	No	
paydex_industry_comment	VARCHAR2(50)	No	
paydex_comment	VARCHAR2(240)	No	
suit_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
lien_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
judgement_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
bankruptcy_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
no_trade_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.

Attribute	Data Type	Required	Validation, Default, Comment
prnt_hq_bkcy_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
num_prnt_bkcy_filing	NUMBER	No	
prnt_bkcy_filg_type	VARCHAR2(20)	No	
prnt_bkcy_filg_chapter	NUMBER	No	
prnt_bkcy_filg_date	DATE	No	
num_prnt_bkcy_convvs	NUMBER	No	
prnt_bkcy_conv_date	DATE	No	
prnt_bkcy_chapter_conv	VARCHAR2(60)	No	
slow_trade_expl	VARCHAR2(100)	No	
negv_pmt_expl	VARCHAR2(150)	No	
pub_rec_expl	VARCHAR2(150)	No	
business_discontinued	VARCHAR2(240)	No	
spcl_event_comment	VARCHAR2(150)	No	
num_spcl_event	NUMBER	No	
spcl_event_update_date	DATE	No	
spcl_evnt_txt	VARCHAR2 (2000)	No	

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Credit Rating Entity

As part of updating a business object, you can provide entity data to create or update the Credit Rating entity.

### Identification Attributes

The API procedure uses the CREDIT\_RATING\_ID attribute to uniquely identify the correct Credit Rating entity in the data model. You must provide sufficient information for this attribute to identify an existing Credit Rating entity.

Attribute	Data Type	Logical Key
credit_rating_id	NUMBER	No
rated_as_of_date	DATE	Yes
rating_organization	VARCHAR2(240)	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
actual_content_source	VARCHAR2(30)	No	Validate: HZ_MIXNM.Utility.ValidateContentSource.
status	VARCHAR2(1)	No	Validation: Must be a valid lookup value from lookup REGISTRY_STATUS
description	VARCHAR2(2000)	No	
rating	VARCHAR2(60)	No	
rated_as_of_date	DATE	No	
rating_organization	VARCHAR2(240)	No	
comments	VARCHAR2(240)	No	
det_history_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
fincl_embt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
criminal_proceeding_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
claims_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
secured_flgng_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
fincl_lgl_event_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
disaster_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
oprg_spec_evnt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
other_spec_evnt_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
avg_high_credit	NUMBER	No	
credit_score	VARCHAR2(30)	No	
credit_score_age	NUMBER	No	
credit_score_class	NUMBER	No	
credit_score_commentary	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY.
credit_score_commentary 2 to credit_score_commentary 10	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup CREDIT_SCORE_COMMENTARY.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
credit_score_date	DATE	No	
credit_score_incd_default	NUMBER	No	
credit_score_natl_percentile	NUMBER	No	
failure_score	VARCHAR2(30)	No	
failure_score_age	NUMBER	No	
failure_score_class	NUMBER	No	
failure_score_commentary	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY.
failure_score_commentary2 to failure_score_commentary10	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_COMMENTARY.
failure_score_date	DATE	No	
failure_score_incd_default	NUMBER	No	
failure_score_natnl_perce ntile	NUMBER	No	
failure_score_override_co de	VARCHAR2(30)	No	
global_failure_score	VARCHAR2(30)	No	
debarment_ind	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup YES/NO.
debarments_count	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
debarments_date	DATE	No	
high_credit	NUMBER	No	
maximum_credit_currency_code	VARCHAR2(240)	No	Validation: Foreign key to fnd_currencies.currency_code.
maximum_credit_rcmd	NUMBER	No	
paydex_norm	VARCHAR2(3)	No	
paydex_score	VARCHAR2(3)	No	
paydex_three_monthsago	VARCHAR2(3)	No	
credit_score_override_code	VARCHAR2(30)	No	Validation: Must be a valid lookup value from lookup FAILURE_SCORE_OVERRIDE_CODE.
cr_scr_clas_expl	VARCHAR2(30)	No	
low_rng_delq_scr	NUMBER	No	
high_rng_delq_scr	NUMBER	No	
delq_pmt_rng_prct	NUMBER	No	
delq_pmt_pctg_for_all_firms	NUMBER	No	
num_trade_experiences	NUMBER	No	
paydex_firm_days	VARCHAR2(15)	No	
paydex_firm_comment	VARCHAR2(60)	No	
paydex_industry_days	VARCHAR2(15)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
paydex_industry_comme nt	VARCHAR2(50)	No	
paydex_comment	VARCHAR2(240)	No	
suit_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
lien_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
judgement_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
bankruptcy_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
no_trade_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
prnt_hq_bkcy_ind	VARCHAR2(5)	No	Validation: Must be a valid lookup value from lookup YES/NO.
num_prnt_bkcy_filing	NUMBER	No	
prnt_bkcy_filg_type	VARCHAR2(20)	No	
prnt_bkcy_filg_chapter	NUMBER	No	
prnt_bkcy_filg_date	DATE	No	
num_prnt_bkcy_convvs	NUMBER	No	
prnt_bkcy_conv_date	DATE	No	
prnt_bkcy_chapter_conv	VARCHAR2(60)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
slow_trade_expl	VARCHAR2(100)	No	
negv_pmt_expl	VARCHAR2(150)	No	
pub_rec_expl	VARCHAR2(150)	No	
business_discontinued	VARCHAR2(240)	No	
spcl_event_comment	VARCHAR2(150)	No	
num_spcl_event	NUMBER	No	
spcl_event_update_date	DATE	No	
spcl_evnt_txt	VARCHAR2 (2000)	No	

## Get Credit Rating Entity

As part of getting a business object, embedded Credit Rating entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Credit Rating entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
credit_rating_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
organization_id	NUMBER
description	VARCHAR2(2000)
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_flgng_ind	VARCHAR2(5)

---

<b>Attribute</b>	<b>Data Type</b>
fincl_lgl_event_ind	VARCHAR2(5)
disaster_ind	VARCHAR2(5)
oprg_spec_evnt_ind	VARCHAR2(5)
other_spec_evnt_ind	VARCHAR2(5)
avg_high_credit	NUMBER
credit_score	VARCHAR2(30)
credit_score_age	NUMBER
credit_score_class	NUMBER
credit_score_commentary	VARCHAR2(30)
credit_score_commentary2 to 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 to failure_score_commentary10	VARCHAR2(30)
failure_score_date	DATE

---

<b>Attribute</b>	<b>Data Type</b>
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

---

---

<b>Attribute</b>	<b>Data Type</b>
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_convvs	NUMBER
prnt_bkcy_conv_date	DATE
prnt_bkcy_chapter_conv	VARCHAR2(60)
slow_trade_expl	VARCHAR2(100)
negv_pmt_expl	VARCHAR2(150)

---

Attribute	Data Type
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)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Customer Account Relationship Entity Attributes

### Create Customer Account Relationship Entity

As part of creating a business object, you can provide entity data to create the Customer Account Relationship entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: Status is lookup code in lookup type CODE_STATUS. Default: A.
relationship_type	VARCHAR2(30)	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE.
comments	VARCHAR2(240)	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2(150)	No	
customer_reciprocal_flag	VARCHAR2(1)	No	Validation: customer_reciprocal_flag is lookup code in lookup type YES/NO.
			Default: N.
bill_to_flag	VARCHAR2(1)	No	
ship_to_flag	VARCHAR2(1)	No	
org_id	NUMBER	No	

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Update Customer Account Relationship Entity

As part of updating a business object, you can provide entity data to create or update the Customer Account Relationship entity.

#### Identification Attributes

The API procedure uses the CUST\_ACCT\_ID attribute to uniquely identify the correct Customer Account Relationship entity in the data model. You must provide sufficient

information for this attribute to identify an existing Customer Account Relationship entity.

- **Attribute:** cust\_acct\_id
- **Data Type:** NUMBER

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• Status cannot be set to null during update.</li><li>• Status is lookup code in lookup type CODE_STATUS.</li></ul>
relationship_type	VARCHAR2(30)	No	Validation: relationship_type is lookup code in lookup type RELATIONSHIP_TYPE.
comments	VARCHAR2(240)	No	

Attribute	Data Type	Required	Validation, Default, Comment
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2(150)	No	
customer_reciprocal_flag	VARCHAR2(1)	No	Validation: Cannot be updated.
bill_to_flag	VARCHAR2(1)	No	
ship_to_flag	VARCHAR2(1)	No	
org_id	NUMBER	No	

## Get Customer Account Relationship Entity

As part of getting a business object, embedded Customer Account Relationship entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Account Relationship entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
cust_acct_id	NUMBER
related_cust_acct_id	NUMBER
relationship_type	VARCHAR2(30)
comments	VARCHAR2(240)
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)
customer_reciprocal_flag	VARCHAR2(1)
bill_to_flag	VARCHAR2(1)
ship_to_flag	VARCHAR2(1)
org_id	NUMBER

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Customer Profile Amount Entity Attributes

### Create Customer Profile Amount Entity

As part of creating a business object, you can provide entity data to create the Customer Profile Amount entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
currency_code	VARCHAR2(15)	Yes	<p>Validation:</p> <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Foreign key to fnd_currencies.</li><li>• For a given cust_account_profile_id and currency_code, only one record of the profile amount is allowed.</li></ul>
trx_credit_limit	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
overall_credit_limit	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
min_dunning_amount	NUMBER	No	
min_dunning_invoice_amount	NUMBER	No	
max_interest_charge	NUMBER	No	
min_statement_amount	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
auto_rec_min_receipt_amount	NUMBER	No	
interest_rate	NUMBER	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2(150)	No	
min_fc_balance_amount	NUMBER	No	
min_fc_invoice_amount	NUMBER	No	
expiration_date	DATE	No	
jgzz_attribute_category	VARCHAR2(30)	No	
jgzz_attribute1 to jgzz_attribute15	VARCHAR2(150)	No	
global_attribute1 to global_attribute20	VARCHAR2(150)	No	
global_attribute_category	VARCHAR2(30)	No	
exchange_rate_type	VARCHAR2(30)	No	
min_fc_invoice_overdue_type	VARCHAR2(30)	No	
min_fc_invoice_percent	NUMBER	No	
min_fc_balance_overdue_type	VARCHAR2(30)	No	
min_fc_balance_percent	NUMBER	No	
interest_type	VARCHAR2(30)	No	
interest_fixed_amount	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
interest_schedule_id	NUMBER	No	
penalty_type	VARCHAR2(30)	No	
penalty_rate	NUMBER	No	
min_interest_charge	NUMBER	No	
penalty_fixed_amount	NUMBER	No	
penalty_schedule_id	NUMBER	No	

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Update Customer Profile Amount Entity

As part of updating a business object, you can provide entity data to create or update the Customer Profile Amount entity.

#### Identification Attributes

The API procedure uses the CUST\_ACCT\_PROFILE\_AMT\_ID attribute to uniquely identify the correct Customer Profile Amount entity in the data model. You must provide sufficient information for this attribute to identify an existing Customer Profile Amount entity.

Attribute	Data Type	Logical Key
cust_acct_profile_amt_id	NUMBER	No
currency_code	VARCHAR2(15)	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
currency_code	VARCHAR2(15)	No	Validation: Cannot be updated.
trx_credit_limit	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
overall_credit_limit	NUMBER	No	Validation: trx_credit_limit must be less than or equal to overall_credit_limit.
min_dunning_amount	NUMBER	No	

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
min_dunning_invoice_am ount	NUMBER	No	
max_interest_charge	NUMBER	No	
min_statement_amount	NUMBER	No	
auto_rec_min_receipt_am ount	NUMBER	No	
interest_rate	NUMBER	No	
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute15	VARCHAR2 (150)	No	
min_fc_balance_amount	NUMBER	No	
min_fc_invoice_amount	NUMBER	No	
expiration_date	DATE	No	
jgzz_attribute_category	VARCHAR2(30)	No	
jgzz_attribute1 to jgzz_attribute15	VARCHAR2 (150)	No	
global_attribute1 to global_attribute20	VARCHAR2 (150)	No	
global_attribute_category	VARCHAR2(30)	No	
exchange_rate_type	VARCHAR2(30)	No	
min_fc_invoice_overdue_t ype	VARCHAR2(30)	No	
min_fc_invoice_percent	NUMBER	No	

Attribute	Data Type	Required	Validation, Default, Comment
min_fc_balance_overdue_type	VARCHAR2(30)	No	
min_fc_balance_percent	NUMBER	No	
interest_type	VARCHAR2(30)	No	
interest_fixed_amount	NUMBER	No	
interest_schedule_id	NUMBER	No	
penalty_type	VARCHAR2(30)	No	
penalty_rate	NUMBER	No	
min_interest_charge	NUMBER	No	
penalty_fixed_amount	NUMBER	No	
penalty_schedule_id	NUMBER	No	

## Get Customer Profile Amount Entity

As part of getting a business object, embedded Customer Profile Amount entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Customer Profile Amount entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
cust_acct_profile_amt_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
cust_acct_profile_id	NUMBER
cust_acct_id	NUMBER
cust_acct_site_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

---

<b>Attribute</b>	<b>Data Type</b>
auto_rec_min_receipt_amount	NUMBER
interest_rate	NUMBER
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)
min_fc_balance_amount	NUMBER
min_fc_invoice_amount	NUMBER
expiration_date	DATE
jgzz_attribute_category	VARCHAR2(30)
jgzz_attribute1 to jgzz_attribute15	VARCHAR2(150)
global_attribute1 to global_attribute20	VARCHAR2(150)
global_attribute_category	VARCHAR2(30)
exchange_rate_type	VARCHAR2(30)
min_fc_invoice_overdue_type	VARCHAR2(30)
min_fc_invoice_percent	NUMBER
min_fc_balance_overdue_type	VARCHAR2(30)
min_fc_balance_percent	NUMBER
interest_type	VARCHAR2(30)
interest_fixed_amount	NUMBER
interest_schedule_id	NUMBER
penalty_type	VARCHAR2(30)

---

Attribute	Data Type
penalty_rate	NUMBER
min_interest_charge	NUMBER
penalty_fixed_amount	NUMBER
penalty_schedule_id	NUMBER

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Education Entity Attributes

### Create Education Entity

As part of creating a business object, you can provide entity data to create the Education entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: Validated against the REGISTRY_STATUS lookup type.
course_major	VARCHAR2(60)	No	
degree_received	VARCHAR2(60)	No	
start_date_attended	DATE	No	

Attribute	Data Type	Required	Validation, Default, Comment
last_date_attended	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	VARCHAR2(60)	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	NUMBER	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	VARCHAR2(30)	No	Validation: If the value is modified, then the value is validated against the HZ_TYPE_OF_SCHOOL lookup type.

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Update Education Entity

As part of updating a business object, you can provide entity data to create or update the Education entity.

### Identification Attributes

The API procedure uses the EDUCATION\_ID attribute to uniquely identify the correct Education entity in the data model. You must provide sufficient information for this attribute to identify an existing Education entity.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Logical Key</b>
education_id	NUMBER	No
course_major	VARCHAR2(60)	Yes
degree_received	VARCHAR2(60)	Yes
school_attended_name	VARCHAR2(60)	Yes

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: Validated against the REGISTRY_STATUS lookup type.
course_major	VARCHAR2(60)	No	
degree_received	VARCHAR2(60)	No	
start_date_attended	DATE	No	
last_date_attended	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	VARCHAR2(60)	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	NUMBER	No	Validation: Must exist in the HZ_PARTIES table.
type_of_school	VARCHAR2(30)	No	

## Get Education Entity

As part of getting a business object, embedded Education entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Education entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
education_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_id	NUMBER
course_major	VARCHAR2(60)
degree_received	VARCHAR2(60)

Attribute	Data Type
start_date_attended	DATE
last_date_attended	DATE
school_attended_name	VARCHAR2(60)
school_party_id	NUMBER
type_of_school	VARCHAR2(30)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Extension Entity Attributes

### Create Extension Attribute Entity

As part of creating a business object, you can provide entity data to create the Extension Attribute entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
attr_value_str	VARCHAR2(1000)	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.
attr_value_num	NUMBER	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.
attr_value_date	DATE	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### **Update Extension Attribute Entity**

As part of updating a business object, you can provide entity data to create or update the Extension Attribute entity.

#### **Identification Attributes**

The API procedure uses the EXTENSION\_ID attribute to uniquely identify the correct Extension Attribute entity in the data model. You must provide sufficient information for this attribute to identify an existing Extension Attribute entity.

<b>Attribute</b>	<b>Data Type</b>	<b>Logical Key</b>
extension_id	NUMBER	No
attr_group_name	VARCHAR2(30)	No
attr_name	VARCHAR2(30)	No
attr_group_type	VARCHAR2(30)	No

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common

object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

Attribute	Data Type
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

Attribute	Data Type	Required	Validation, Default, Comment
attr_value_str	VARCHAR2(1000)	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.
attr_value_num	NUMBER	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.
attr_value_date	DATE	Yes or No	Validation: One of attr_value_str, attr_value_date, or attr_value_num is mandatory.

---

### Get Extension Attribute Entity

As part of getting a business object, embedded Extension Attribute entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Extension Attribute entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)

- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
attr_group_name	VARCHAR2(30)
attr_name	VARCHAR2(30)
attr_value_str	VARCHAR2(1000)
attr_value_num	NUMBER
attr_value_date	DATE
attr_display_value	VARCHAR2(1000)
attr_group_type	VARCHAR2(30)
attr_group_id	NUMBER
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER

### Related Topics

[Granular Entity Attributes, page 24-245](#)

## Financial Number Entity Attributes

### Create Financial Number Entity

As part of creating a business object, you can provide entity data to create the Financial Number entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	Yes	Validation: validated against AR lookup type REGISTRY_STATUS.
financial_number	NUMBER	No	
financial_number_name	VARCHAR2(60)	No	Validation: Must be a valid lookup under FIN_NUM_NAME lookup type.
financial_units_applied	NUMBER	No	
financial_number_currenc y	VARCHAR2(240)	No	
projected_actual_flag	VARCHAR2(1)	No	

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Update Financial Number Entity

As part of updating a business object, you can provide entity data to create or update the Financial Number entity.

### Identification Attributes

The API procedure uses the FINANCIAL\_NUMBER\_ID attribute to uniquely identify the correct Financial Number entity in the data model. You must provide sufficient information for this attribute to identify an existing Financial Number entity.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Logical Key</b>
financial_number_id	NUMBER	No
financial_number_name	VARCHAR2(60)	Yes

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	Yes	Validation: validated against AR lookup type REGISTRY_STATUS.
financial_number	NUMBER	No	
financial_number_name	VARCHAR2(60)	No	Validation: Must be a valid lookup under FIN_NUM_NAME lookup type.
financial_units_applied	NUMBER	No	
financial_number_currenc y	VARCHAR2(240)	No	
projected_actual_flag	VARCHAR2(1)	No	

## Get Financial Number Entity

As part of getting a business object, embedded Financial Number entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Financial Number entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
financial_number_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
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)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Financial Profile Entity Attributes

### Create Financial Profile Entity

As part of creating a business object, you can provide entity data to create the Financial Profile entity.

### **Accepted Attributes**

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>
status	VARCHAR2(1)
access_authority_date	DATE
access_authority_granted	VARCHAR2(1)
balance_amount	NUMBER
balance_verified_on_date	DATE
financial_account_number	VARCHAR2(60)
financial_account_type	VARCHAR2(30)
financial_org_type	VARCHAR2(30)
financial_organization_name	VARCHAR2(240)

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

## Update Financial Profile Entity

As part of updating a business object, you can provide entity data to create or update the Financial Profile entity.

### Identification Attributes

The API procedure uses the FINANCIAL\_PROFILE\_ID attribute to uniquely identify the correct Financial Profile entity in the data model. You must provide sufficient information for this attribute to identify an existing Financial Profile entity.

- **Attribute:** financial\_profile\_id
- **Data Type:** NUMBER

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
status	VARCHAR2(1)
access_authority_date	DATE
access_authority_granted	VARCHAR2(1)

Attribute	Data Type
balance_amount	NUMBER
balance_verified_on_date	DATE
financial_account_number	VARCHAR2(60)
financial_account_type	VARCHAR2(30)
financial_org_type	VARCHAR2(30)
financial_organization_name	VARCHAR2(240)

## Get Financial Profile Entity

As part of getting a business object, embedded Financial Profile entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Financial Profile entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

---

<b>Attribute</b>	<b>Data Type</b>
financial_profile_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
access_authority_date	DATE
access_authority_granted	VARCHAR2(1)
balance_amount	NUMBER
balance_verified_on_date	DATE
financial_account_number	VARCHAR2(60)
financial_account_type	VARCHAR2(30)
financial_org_type	VARCHAR2(30)
financial_organization_name	VARCHAR2(240)

---

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Organization Contact Role Entity Attributes

### Create Organization Contact Role Entity

As part of creating a business object, you can provide entity data to create the Organization Contact Role entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.  Default: A.
role_type	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Validated against AR lookup type CONTACT_ROLE_TYPE.</li></ul>
primary_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
role_level	VARCHAR2(30)	No	
primary_contact_per_role_type	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

## Update Organization Contact Role Entity

As part of updating a business object, you can provide entity data to create or update the Organization Contact Role entity.

### Identification Attributes

The API procedure uses the ORG\_CONTACT\_ROLE\_ID attribute to uniquely identify the correct Organization Contact Role entity in the data model. You must provide sufficient information for this attribute to identify an existing Organization Contact Role entity.

---

<b>Attribute</b>	<b>Data Type</b>	<b>Logical Key</b>
org_contact_role_id	NUMBER	No
orig_system	VARCHAR2(30)	No
role_type	VARCHAR2(30)	Yes

---

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS.</li><li>• Cannot be set to null during update.</li></ul>
role_type	VARCHAR2(30)	No	Validation: <ul style="list-style-type: none"><li>• Cannot be set to null during update.</li><li>• Validated against AR lookup type CONTACT_ROLE_TYPE.</li></ul>
primary_flag	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
role_level	VARCHAR2(30)	No	
primary_contact_per_role_type	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

### Get Organization Contact Role Entity

As part of getting a business object, embedded Organization Contact Role entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Organization Contact Role entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
org_contact_role_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
org_contact_id	NUMBER
role_type	VARCHAR2(30)
primary_flag	VARCHAR2(1)
role_level	VARCHAR2(30)
primary_contact_per_role_type	VARCHAR2(1)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Party Preference Entity Attributes

### Create Party Preference Entity

As part of creating a business object, you can provide entity data to create the Party Preference entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
category	VARCHAR2(30)
preference_code	VARCHAR2(20)
value_varchar2	VARCHAR2(240)
value_number	NUMBER
value_date	DATE
value_name	VARCHAR2(50)
module	VARCHAR2(50)
additional_value1	VARCHAR2(150)
additional_value2	VARCHAR2(150)
additional_value3	VARCHAR2(150)
additional_value4	VARCHAR2(150)
additional_value5	VARCHAR2(150)

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### **Update Party Preference Entity**

As part of updating a business object, you can provide entity data to create or update the Party Preference entity.

#### **Identification Attributes**

The API procedure uses the PARTY\_PREFERENCE\_ID attribute to uniquely identify the correct Party Preference entity in the data model. You must provide sufficient information for this attribute to identify an existing Party Preference entity.

<b>Attribute</b>	<b>Data Type</b>	<b>Logical Key</b>
party_preference_id	NUMBER	No
category	VARCHAR2(30)	Yes
preference_code	VARCHAR2(20)	Yes
module	VARCHAR2(50)	Yes

### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common

object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
category	VARCHAR2(30)
preference_code	VARCHAR2(20)
value_varchar2	VARCHAR2(240)
value_number	NUMBER
value_date	DATE
value_name	VARCHAR2(50)
module	VARCHAR2(50)
additional_value1	VARCHAR2(150)
additional_value2	VARCHAR2(150)
additional_value3	VARCHAR2(150)
additional_value4	VARCHAR2(150)
additional_value5	VARCHAR2(150)

#### Get Party Preference Entity

As part of getting a business object, embedded Party Preference entity attributes are

also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Party Preference entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
party_preference_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
category	VARCHAR2(30)
preference_code	VARCHAR2(20)

Attribute	Data Type
value_varchar2	VARCHAR2(240)
value_number	NUMBER
value_date	DATE
value_name	VARCHAR2(50)
module	VARCHAR2(50)
additional_value1	VARCHAR2(150)
additional_value2	VARCHAR2(150)
additional_value3	VARCHAR2(150)
additional_value4	VARCHAR2(150)
additional_value5	VARCHAR2(150)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Party Site Use Entity Attributes

### Create Party Site Use Entity

As part of creating a business object, you can provide entity data to create the Party Site Use entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: Validated against AR lookup type REGISTRY_STATUS.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
comments	VARCHAR2(240)	No	
site_use_type	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Validated against AR lookup type PARTY_SITE_USE_CODE.</li> </ul>
primary_per_type	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO. If the status is set to Inactive ( <i>I</i> ), then this value cannot be set to Yes ( <i>Y</i> ).

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Update Party Site Use Entity

As part of updating a business object, you can provide entity data to create or update the Party Site Use entity.

#### Identification Attributes

The API procedure uses the PARTY\_SITE\_USE\_ID attribute to uniquely identify the correct Party Site Use entity in the data model. You must provide sufficient information for this attribute to identify an existing Party Site Use entity.

Attribute	Data Type	Logical Key
party_site_use_id	NUMBER	No
site_use_type	VARCHAR2(30)	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• Validated against AR lookup type REGISTRY_STATUS.</li> <li>• Cannot be updated to null.</li> </ul>
comments	VARCHAR2(240)	No	

Attribute	Data Type	Required	Validation, Default, Comment
site_use_type	VARCHAR2(30)	No	Validation: Cannot be updated.
primary_per_type	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

## Get Party Site Use Entity

As part of getting a business object, embedded Party Site Use entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Party Site Use entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
party_site_use_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)

Attribute	Data Type
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
party_site_id	NUMBER
comments	VARCHAR2(240)
site_use_type	VARCHAR2(30)
primary_per_type	VARCHAR2(1)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Payment Method Entity Attributes

### Create Payment Method Entity

As part of creating a business object, you can provide entity data to create the Payment Method entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
receipt_method_id	NUMBER
primary_flag	VARCHAR2(1)

<b>Attribute</b>	<b>Data Type</b>
start_date	DATE
end_date	DATE
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)

#### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

#### **Update Payment Method Entity**

As part of updating a business object, you can provide entity data to create or update the Payment Method entity.

##### **Identification Attributes**

The API procedure uses the PAYMENT\_METHOD\_ID attribute to uniquely identify the correct Payment Method entity in the data model. You must provide sufficient information for this attribute to identify an existing Payment Method entity.

- **Attribute:** payment\_method\_id
- **Data Type:** NUMBER

#### **Common Object Attribute**

In certain cases, parties may be integrated between TCA and other systems using a

"common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
receipt_method_id	NUMBER
primary_flag	VARCHAR2(1)
start_date	DATE
end_date	DATE
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)

### Get Payment Method Entity

As part of getting a business object, embedded Payment Method entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Payment Method entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure

call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
payment_method_id	NUMBER
cust_acct_id	NUMBER
site_use_id	NUMBER
receipt_method_id	NUMBER
primary_flag	VARCHAR2(1)
start_date	DATE
end_date	DATE
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)
created_by_name	VARCHAR2(100)
creation_date	DATE
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Person Interest Entity Attributes

### Create Person Interest Entity

As part of creating a business object, you can provide entity data to create the Person Interest entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type
status	VARCHAR2(1)
level_of_interest	VARCHAR2(30)
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

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Person Interest Entity

As part of updating a business object, you can provide entity data to create or update the Person Interest entity.

### Identification Attributes

The API procedure uses the PERSON\_INTEREST\_ID attribute to uniquely identify the correct Person Interest entity in the data model. You must provide sufficient information for this attribute to identify an existing Person Interest entity.

Attribute	Data Type	Logical Key
person_interest_id	NUMBER	No
interest_type_code	VARCHAR2(30)	Yes
sub_interest_type_code	VARCHAR2(30)	Yes
interest_name	VARCHAR2(240)	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

---

Attribute	Data Type
common_obj_id	VARCHAR2(255)

---

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

---

Attribute	Data Type
status	VARCHAR2(1)
level_of_interest	VARCHAR2(30)
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

---

### Get Person Interest Entity

As part of getting a business object, embedded Person Interest entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Person Interest entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call.

See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
person_interest_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_id	NUMBER
level_of_interest	VARCHAR2(30)
level_of_participation	VARCHAR2(30)
interest_type_code	VARCHAR2(30)
comments	VARCHAR2(240)

Attribute	Data Type
sport_indicator	VARCHAR2(1)
sub_interest_type_code	VARCHAR2(30)
interest_name	VARCHAR2(240)
team	VARCHAR2(240)
since	DATE

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Person Language Entity Attributes

### Create Person Language Entity

As part of creating a business object, you can provide entity data to create the Person Language entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: Validated against AR lookup type REGISTRY_STATUS. Default: A.
language_name	VARCHAR2(4)	Yes	Validation: <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Foreign key to fnd_languages. language_code</li> </ul>

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
native_language	VARCHAR2(1)	No	
primary_language_indicat or	VARCHAR2(1)	No	
reads_level	VARCHAR2(30)	No	
speaks_level	VARCHAR2(30)	No	
writes_level	VARCHAR2(30)	No	
spoken_comprehension_le vel	VARCHAR2(30)	No	Validation: Validated against the HZ_LANGUAGE_PROFICIENC Y lookup type.

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

<b>Attribute</b>	<b>Data Type</b>
common_obj_id	VARCHAR2(255)

### Update Person Language Entity

As part of updating a business object, you can provide entity data to create or update the Person Language entity.

#### Identification Attributes

The API procedure uses the LANGUAGE\_USE\_REFERENCE\_ID attribute to uniquely identify the correct Person Language entity in the data model. You must provide sufficient information for this attribute to identify an existing Person Language entity.

Attribute	Data Type	Logical Key
language_use_reference_id	NUMBER	No
language_name	VARCHAR2(4)	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS.</li><li>• Cannot be set to null during update.</li></ul>
language_name	VARCHAR2(4)	No	Validation: Cannot be updated.
native_language	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.

Attribute	Data Type	Required	Validation, Default, Comment
primary_language_indicat or	VARCHAR2(1)	No	Validation: Validated against AR lookup type YES/NO.
reads_level	VARCHAR2(30)	No	
speaks_level	VARCHAR2(30)	No	
writes_level	VARCHAR2(30)	No	
spoken_comprehension_l evel	VARCHAR2(30)	No	

## Get Person Language Entity

As part of getting a business object, embedded Person Language entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Person Language entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
language_use_reference_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
person_id	NUMBER
language_name	VARCHAR2(4)
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)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Relationship Entity Attributes

### Create Relationship Entity

As part of creating a business object, you can provide entity data to create the Relationship entity.

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Foreign key to HZ_ORIG_SYSTEMS_B.orig_system with sst_flag value of Y.  Default: USER_ENTERED.
status	VARCHAR2(1)	No	Validation: Validated against AR lookup type REGISTRY_STATUS.  Default: A.
related_object_type	VARCHAR2(30)	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• Foreign key to fnd_object_instance_sets. instance_set_name.</li></ul>
related_object_id	NUMBER	Yes	Validation: <ul style="list-style-type: none"><li>• Mandatory attribute.</li><li>• 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.</li></ul>
related_object_orig_system	VARCHAR2(30)		

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
related_object_osr	VARCHAR2(255)		
relationship_code	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Validated against AR lookup type PARTY_RELATIONS_TYPE.</li> <li>• Required to be a valid relationship code for the particular relationship type requested.</li> </ul>
relationship_type	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• Foreign key to hz_relationship_types. relationship_type.</li> </ul>
comments	VARCHAR2(240)	No	
start_date	DATE	Yes	<p>Validation: Must be less than end_date if end_date is passed.</p> <p>Default: sysdate.</p>
end_date	DATE	No	<p>Default: 31-DEC-4712</p> <p>Validation: Must be greater than start_date.</p>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
additional_information1 to additional_information30	VARCHAR2(150)	No	
percentage_ownership	NUMBER	No	

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a

"common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

## Update Relationship Entity

As part of updating a business object, you can provide entity data to create or update the Relationship entity.

### Identification Attributes

The API procedure uses the RELATIONSHIP\_ID attribute to uniquely identify the correct Relationship entity in the data model. You must provide sufficient information for this attribute to identify an existing Relationship entity.

Attribute	Data Type	Logical Key
relationship_id	NUMBER	No
related_object_type	VARCHAR2(30)	Yes
related_object_id	NUMBER	Yes
related_object_orig_system	VARCHAR2(30)	Yes
related_object_osr	VARCHAR2(255)	Yes
relationship_code	VARCHAR2(30)	Yes
relationship_type	VARCHAR2(30)	Yes
start_date	DATE	Yes

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
actual_content_source	VARCHAR2(30)	No	Validation: Value will not be updated in the database.
status	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"><li>• Validated against AR lookup type REGISTRY_STATUS.</li><li>• Cannot be updated to null.</li></ul>
related_object_type	VARCHAR2(30)	No	Validation: Cannot be updated.
related_object_id	NUMBER	No	Validation: Cannot be updated.
related_object_orig_system	VARCHAR2(30)		
related_object_osr	VARCHAR2(255)		
relationship_code	VARCHAR2(30)	No	Validation: Cannot be updated.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
relationship_type	VARCHAR2(30)	No	Validation: Cannot be updated.
comments	VARCHAR2(240)	No	
start_date	DATE	Yes	Validation: Cannot be set to null during update. Cannot be greater than end_date.
end_date	DATE	No	Validation: Must be greater than start date.
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	
additional_information1 to additional_information30	VARCHAR2(150)	No	
percentage_ownership	NUMBER	No	

## Get Relationship Entity

As part of getting a business object, embedded Relationship entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Relationship entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null

- *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
relationship_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
actual_content_source	VARCHAR2(30)
status	VARCHAR2(1)
parent_object_type	VARCHAR2(30)
parent_object_id	NUMBER
related_object_type	VARCHAR2(30)
related_object_id	NUMBER
related_object_orig_system	VARCHAR2(30)
related_object_osr	VARCHAR2(255)
relationship_code	VARCHAR2(30)
relationship_type	VARCHAR2(30)

<b>Attribute</b>	<b>Data Type</b>
comments	VARCHAR2(240)
start_date	DATE
end_date	DATE
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)
additional_information1 to additional_information30	VARCHAR2(150)
percentage_ownership	NUMBER

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Account Contact Role Entity Attributes

### Create Account Contact Role Entity

As part of creating a business object, you can provide entity data to create the Account Contact Role entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

<b>Attribute</b>	<b>Data Type</b>	<b>Required</b>	<b>Validation, Default, Comment</b>
responsibility_type	VARCHAR2(30)	Yes	<p>Validation:</p> <ul style="list-style-type: none"> <li>• Mandatory attribute.</li> <li>• It is a lookup code in lookup type SITE_USE_CODE.</li> </ul>

Attribute	Data Type	Required	Validation, Default, Comment
primary_flag	VARCHAR2(1)	No	<p>Validation:</p> <ul style="list-style-type: none"> <li>Primary Flag is lookup code in lookup type YES/NO.</li> <li>It is unique per cust_account_role_id.</li> </ul> <p>Default: N.</p>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Update Account Contact Role Entity

As part of updating a business object, you can provide entity data to create or update the Account Contact Role entity.

#### Identification Attributes

The API procedure uses the RESPONSIBILITY\_ID attribute to uniquely identify the correct Account Contact Role entity in the data model. You must provide sufficient information for this attribute to identify an existing Account Contact Role entity.

Attribute	Data Type	Logical Key
responsibility_id	NUMBER	No
responsibility_type	VARCHAR2(30)	Yes

#### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

#### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
responsibility_type	VARCHAR2(30)	No	Validation: Cannot be updated.
primary_flag	VARCHAR2(1)	No	Validation: <ul style="list-style-type: none"> <li>• Primary Flag is lookup code in lookup type YES/NO.</li> <li>• It is unique per cust_account_role_id.</li> </ul>
attribute_category	VARCHAR2(30)	No	
attribute1 to attribute20	VARCHAR2(150)	No	

## Get Account Contact Role Entity

As part of getting a business object, embedded Account Contact Role entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Account Contact Role entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

Attribute	Data Type
responsibility_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
cust_acct_contact_id	NUMBER

<b>Attribute</b>	<b>Data Type</b>
responsibility_type	VARCHAR2(30)
primary_flag	VARCHAR2(1)
attribute_category	VARCHAR2(30)
attribute1 to attribute15	VARCHAR2(150)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Source System Information Entity Attributes

As part of getting a business object, embedded Source System Information entity attributes are also retrieved. The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
orig_system_ref_id	NUMBER
orig_system	VARCHAR2(30)
orig_system_reference	VARCHAR2(255)
object_type	VARCHAR2(30)
object_id	NUMBER
status	VARCHAR2(1)
reason_code	VARCHAR2(30)
old_orig_system_reference	VARCHAR2(255)
start_date_active	DATE
end_date_active	DATE

Attribute	Data Type
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
attribute_category	VARCHAR2(30)
attribute1 to attribute20	VARCHAR2(150)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Work Class Entity Attributes

### Create Work Class Entity

As part of creating a business object, you can provide entity data to create the Work Class entity.

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.
level_of_experience	VARCHAR2(60)	No	
work_class_name	VARCHAR2(240)	Yes	

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Update Work Class Entity

As part of updating a business object, you can provide entity data to create or update the Work Class entity.

#### Identification Attributes

The API procedure uses the WORK\_CLASS\_ID attribute to uniquely identify the correct Work Class entity in the data model. You must provide sufficient information for this attribute to identify an existing Work Class entity.

- **Attribute:** work\_class\_id
- **Data Type:** NUMBER

### Common Object Attribute

In certain cases, parties may be integrated between TCA and other systems using a "common object" model. In such a model, a common identifier for the object may sit outside of the TCA data model and a caller of the procedure will require the common object ID as a return parameter of the API call.

This attribute is returned from the API so that the caller may identify the common object that corresponds to the TCA object that has been created or updated. The attribute is not stored in TCA tables and is "passed through" the API in the return object.

Attribute	Data Type
common_obj_id	VARCHAR2(255)

### Accepted Attributes

The API procedure accepts the attributes shown in this table, and stores the corresponding data in the data model.

Attribute	Data Type	Required	Validation, Default, Comment
status	VARCHAR2(1)	No	Validation: validated against AR lookup type REGISTRY_STATUS.
level_of_experience	VARCHAR2(60)	No	
work_class_name	VARCHAR2(240)	Yes	

### Get Work Class Entity

As part of getting a business object, embedded Work Class entity attributes are also retrieved. These attributes are similarly extracted for both event independent and event dependent Get business object API procedures, except for the ACTION\_TYPE attribute.

#### Action Type Attribute

The ACTION\_TYPE attribute provides information about how an Update business event affected the retrieved Work Class entity. This attribute is only used when the entity is retrieved as part of an event dependent Get business object API procedure call. See: Get Updated Business Object Procedures, page 22-16.

- **Attribute:** ACTION\_TYPE
- **Data Type:** VARCHAR2(30)
- **Possible Returned Values:**
  - *Event Independent:* Null
  - *Event Dependent:* CREATED, UPDATED, or UNCHANGED

#### Retrieved Attributes

The API procedure retrieves values for the attributes shown in this table.

<b>Attribute</b>	<b>Data Type</b>
work_class_id	NUMBER
last_update_date	DATE
last_updated_by_name	VARCHAR2(100)
creation_date	DATE
created_by_name	VARCHAR2(100)
program_update_date	DATE
created_by_module	VARCHAR2(150)
status	VARCHAR2(1)
employment_history_id	NUMBER
level_of_experience	VARCHAR2(60)
work_class_name	VARCHAR2(240)

## Related Topics

[Granular Entity Attributes, page 24-245](#)

## Business Events

This chapter covers the following topics:

- Overview
- Installation and Setup
- Trading Community Architecture Granular Events
- Trading Community Architecture Business Object 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.

#### Setup

The following procedure describes how to setup event subscription.

## Event Subscription

1. Log on to Oracle Applications using the Workflow Administrator Web Application responsibility.
2. Navigate to Business Events and search for a business event.
3. Click Create Subscription.
4. Enter the action and other parameters for the subscription.

For granular events, TCA uses the same naming convention:

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

For business object events, TCA uses the following naming convention:

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

For more information on subscribing to events, see: *Defining Event Subscriptions, Oracle Workflow Developer's Guide*.

## Workflow Agent Listener Concurrent Program

You should run subscriptions in deferred mode using the Oracle Workflow Business Event System, so that no overhead is added to the process that raised the event. The phase number of a 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.

For more information, see: *Deferring Subscription Processing Using Subscription Phase Numbers, and Managing Business Events in Oracle Workflow Developer's Guide*.

## Related Topics

[Setting Up Business Events, Oracle Trading Community Architecture Administration Guide](#)

[Business Object Overview, page 20-2](#)

[Seeded Business Object APIs, page 22-2](#)

[Business Object API Features, page 22-5](#)

[Business Object API Attributes Information Overview, page 24-2](#)

## Trading Community Architecture Granular 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: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.Person.create</li><li>• oracle.apps.ar.hz.Person.update</li></ul>	<p>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.</p> <p>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.</p> <p>HZ_PARTIES table contains denormalized information from the HZ_PERSON_PROFILES, HZ_ORGANIZATION_PROFILES, HZ_LOCATIONS, HZ_CONTACT_POINTS and HZ_PERSON_LANGUAGE tables.</p>	PARTY_ID

Name	Description of the Physical Entity	ID Parameter Name
Organization:	An organization is a party of the Organization type.	PARTY_ID
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.Organization.create</li> <li>• oracle.apps.ar.hz.Organization.update</li> </ul>	<p>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.</p>	
Group:	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
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.Group.create</li> <li>• oracle.apps.ar.hz.Group.update</li> </ul>		
Location:	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
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.Location.create</li> <li>• oracle.apps.ar.hz.Location.update</li> </ul>		
Party Site:	<p>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.</p> <p>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.</p> <p>For example, 500 Oracle Parkway can be specified as a party site for Oracle. This party site can be used for multiple customer accounts with the same party.</p>	PARTY_SITE_ID
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.PartySite.create</li> <li>• oracle.apps.ar.hz.PartySite.update</li> </ul>		

Name	Description of the Physical Entity	ID Parameter Name
Party Site Use: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.PartySiteUse.create</li><li>• oracle.apps.ar.hz.PartySiteUse.update</li></ul>	A relation between a party and a location can have several uses such as delivery or billing. The HZ_PARTY_SITEUSES 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
Relationship: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.Relationship.create</li><li>• oracle.apps.ar.hz.Relationship.update</li></ul>	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: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.OrgContact.create</li><li>• oracle.apps.ar.hz.OrgContact.update</li></ul>	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

Name	Description of the Physical Entity	ID Parameter Name
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: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.RoleResponsibility.create</li><li>• oracle.apps.ar.hz.RoleResponsibility.update</li></ul>	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: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.CustAcctRelate.create</li><li>• oracle.apps.ar.hz.CustAcctRelate.update</li></ul>	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: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.CustAcctSite.create</li><li>• oracle.apps.ar.hz.CustAcctSite.update</li></ul>	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

Name	Description of the Physical Entity	ID Parameter Name
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_SITEUSES_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 LE  P_CREATE_PROFILE_AMT
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

Name	Description of the Physical Entity	ID Parameter Name
Customer Profile: <ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.CustomerProfile.create</li> <li>• oracle.apps.ar.hz.CustomerProfile.update</li> </ul>	<p>A customer account has a customer profile class that provides default customer account attributes.</p> <p>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.</p> <p>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.</p>	
Contact Point: <ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.ContactPoint.create</li> <li>• oracle.apps.ar.hz.ContactPoint.update</li> </ul>	<p>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.</p> <p>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.</p>	CONTACT_POINT_ID

Name	Description of the Physical Entity	ID Parameter Name
Contact Preference:  • oracle.apps.ar.hz.ContactPreference.create  • oracle.apps.ar.hz.ContactPreference.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_PREF ERENCE_ID
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

Name	Description of the Physical Entity	ID Parameter Name
Citizenship:	<p>Citizenship is part of a person's information.</p> <ul style="list-style-type: none"> <li>oracle.apps.ar.hz.Citizenship.create</li> <li>oracle.apps.ar.hz.Citizenship.update</li> </ul> <p>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.</p>	CITIZENSHIP_ID
Class Category:	<p>Class Categories provides a way to classify parties and party sites.</p> <ul style="list-style-type: none"> <li>oracle.apps.ar.hz.ClassCategory.create</li> <li>oracle.apps.ar.hz.ClassCategory.update</li> </ul> <p>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.</p>	CLASS_CATEGORY
Class Category Use:	<p>Classification is an open structure. Class Category Use indicates which table uses which class category.</p> <ul style="list-style-type: none"> <li>oracle.apps.ar.hz.ClassCategoryUse.create</li> <li>oracle.apps.ar.hz.ClassCategoryUse.update</li> </ul> <p>The HZ_CLASS_CATEGORYUSES table stores information about the tables that use a particular class category.</p> <p>For example, the SIC 1987 class category can be used to classify the parties that have the Organization party type.</p>	CLASS_CATEGORY OWNER_TABLE COLUMN_NAME
Class Code Relationship:	<p>Class Code Relationships provides a method to build hierarchy structures within Class Codes.</p> <ul style="list-style-type: none"> <li>oracle.apps.ar.hz.ClassCodeRelation.create</li> <li>oracle.apps.ar.hz.ClassCodeRelation.update</li> </ul> <p>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.</p> <p>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).</p>	CLASS_CATEGORY CLASS_CODE SUB_CLASS_CODE

Name	Description of the Physical Entity	ID Parameter Name
Code Assignment: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.CodeAssignment.create</li><li>• oracle.apps.ar.hz.CodeAssignment.update</li></ul>	<p>Code Assignment links an instance of the class code to an instance of the classified table.</p> <p>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.</p>	CODE_ASSIGNMENT_ID
Education: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.Education.create</li><li>• oracle.apps.ar.hz.Education.update</li></ul>	<p>A party of the Person type can have an education history.</p> <p>The HZ_EDUCATION table stores general information about the educational history of a party of the Person type.</p> <p>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.</p>	EDUCATION_ID
Employment History: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.EmploymentHistory.create</li><li>• oracle.apps.ar.hz.EmploymentHistory.update</li></ul>	<p>A party of the Person type can have an employment history.</p> <p>The HZ_EMPLOYMENT_HISTORY table stores information about the employment and military service history of a party of the Person type.</p> <p>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.</p>	EMPLOYMENT_HISTORY_ID
Financial Profile: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.FinancialProfile.create</li><li>• oracle.apps.ar.hz.FinancialProfile.update</li></ul>	<p>The HZ_FINANCIAL_PROFILE table stores information about the financial accounts owned by a party.</p> <p>For example, when you perform a credit evaluation of a prospective customer, data may be entered about the balance in the prospective customer's bank account.</p>	FINANCIAL_PROFILE_ID

Name	Description of the Physical Entity	ID Parameter Name
Financial Report: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.FinancialReport.create</li><li>• oracle.apps.ar.hz.FinancialReport.update</li></ul>	<p>The HZ_FINANCIAL_REPORTS table stores information about the reports on the financial status of a party. Detailed numerical financial data is available in the HZ_FINANCIAL_NUMBERS table.</p> <p>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.</p>	FINANCIAL_REPORT_ID
Financial Number: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.FinancialNumber.create</li><li>• oracle.apps.ar.hz.FinancialNumber.update</li></ul>	<p>The HZ_FINANCIAL_NUMBERS table stores detailed financial information for the HZ_FINANCIAL_REPORTS table.</p> <p>For example, a record could store the information that projected leasing revenue includes 1,000,000 Italian lira.</p>	FINANCIAL_NUMBER_ID
Person Interest: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.PersonInterest.create</li><li>• oracle.apps.ar.hz.PersonInterest.update</li></ul>	<p>The HZ_PERSON_INTEREST table stores information about sports, hobbies and personal interests of a party of the Person type.</p> <p>This information could be helpful in developing a more personalized relationship with a business contact.</p> <p>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.</p>	PERSON_INTEREST_ID
Person Language: <ul style="list-style-type: none"><li>• oracle.apps.ar.hz.PersonLanguage.create</li><li>• oracle.apps.ar.hz.PersonLanguage.update</li></ul>	<p>The HZ_PERSON_LANGUAGE table stores information about a language spoken by a party of the Person type.</p> <p>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.</p>	LANGUAGE_USE_REFERENCE_ID

Name	Description of the Physical Entity	ID Parameter Name
Work Class:		
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.WorkClass.create</li> <li>• oracle.apps.ar.hz.WorkClass.update</li> </ul>	<p>The HZ_WORK_CLASS table stores custom-defined classification information about a person's job title and work experience.</p> <p>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.</p>	WORK_CLASS_ID
Original System Reference:		
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.origSystemRef.create</li> <li>• oracle.apps.ar.hz.origSystemRef.update</li> </ul>	<p>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.</p>	ORIG_SYSTEM_REF_ID
Account Merge:		
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.CustAccount.merge</li> </ul>		
Party Merge:		
<ul style="list-style-type: none"> <li>• oracle.apps.ar.hz.Party.merge</li> </ul>		

## Related Topics

- Business Object Overview, page 20-2
- Seeded Business Object APIs, page 22-2
- Business Object API Features, page 22-5
- Business Object API Attributes Information Overview, page 24-2

## Trading Community Architecture Business Object Events

This table provides the following information about the TCA Business Objects Events:

- Name of event
- Description

Event Name	Event Description
Persons Created oracle.apps.ar.hz.personBO.create	Raised when one or more Person business objects are completely created for all mandatory embedded objects, structures, and entities.
Persons Updated oracle.apps.ar.hz.personBO.update	Raised when one or more TCA entities embedded within a complete Person business object are created or modified.
Person Customers Created oracle.apps.ar.hz.CustBO.create	Raised when one or more Person Customer business objects are completely created for all mandatory embedded objects, structures, and entities. This business object combines TCA Party and Account layers and is created when a Customer Account business object is created for an existing Person business object.
Person Customers Updated oracle.apps.ar.hz.CustBO.update	Raised when one or more TCA embedded entities within a complete Person Customer business object are created or modified.
Organizations Created oracle.apps.ar.hz.orgBO.create	Raised when one or more Organization business objects are completely created for all mandatory embedded objects, structures, and entities.
Organizations Updated oracle.apps.ar.hz.orgBO.update	Raised when one or more TCA entities embedded within a complete Organization business object are created or modified.
Organization Customers Created oracle.apps.ar.hz.orgCustBO.create	Raised when one or more Organization Customer business objects are completely created for all mandatory embedded objects, structures, and entities. This business object combines TCA Party and Account layers and is created when a Customer Account business object is created for an existing Organization business object.

---

Event Name	Event Description
Organization Customers Updated oracle.apps.ar.hz.orgCustBO.update	Raised when one or more TCA embedded entities within a complete Organization Customer business object are created or modified.

---

## Related Topics

- Business Object Overview, page 20-2
- Seeded Business Object APIs, page 22-2
- Business Object API Features, page 22-5
- Business Object API Attributes Information Overview, page 24-2

---

## Resource Manager APIs Overview and Features

This chapter covers the following topics:

- Resource Manager APIs
- Private APIs
- Public APIs
- Public, published APIs
- Parameter Specifications
- Standard IN Parameters
- Standard OUT Parameters
- Parameter Size
- Missing Parameter Attributes
- Parameter Validations
- Invalid Parameters
- Version Information
- Status Messages

### Resource Manager APIs

The Resource Manager contains the following types of APIs:

#### Private APIs

Private APIs are for internal, development use only. Details are not provided to anyone outside of the immediate development environment, nor are they intended for use by anyone outside of the e-Business Suite development environment.

## Public APIs

Public APIs are designed for customers and Oracle consultants to integrate non-Oracle systems into Oracle e-Business Suite or to extend the functionality of the base products. Oracle does not support public APIs unless they are published in a reference manual such as this one. The user accepts all risk and responsibility for working with non-published public APIs.

## Public, published APIs

Public Published APIs are guaranteed by Oracle to remain valid and that patches will not alter the API behavior. Public, published APIs are supported by Oracle to the same extent as released software.

For non-published APIs, Oracle expressly does not provide any guarantees regarding consistency of naming, usage, or behavior of any API (public or private). It is also possible that a patch could alter any characteristic of any non-published e-Business Suite API. As such, those who choose to use these APIs do so at their own risk. However, Oracle does attempt to minimize all changes to public APIs, even if not published.

**Note:** Earlier, many of the Oracle E-Business Suite's' PL/SQL server side APIs have been enhanced to utilize the pass by reference semantics of PL/SQL. This improves performance considerably and reduces memory consumption. In the normal processing case (i.e. success), there is no change of behavior, and callers of these APIs are not impacted. However, in the case of exceptions, there is a behavior change which results in assignments being exposed to the caller, which are made in the API prior to any exceptions being raised. The previous behavior would rollback these assignments made by the API if an exception occurred in this API. Developers writing custom extensions to Oracle E-Business Suite, or third party integrators which use the standard Oracle E-Business Suite's APIs should be aware of this change in semantics.

Each published API provides an API specification, and definitions as for its parameters, data structures, and status messages. Sample scripts and documented process flow diagrams are included where applicable.

**Note:** The words *procedure* and *API* are used interchangeably in this document.

## Parameter Specifications

The specifications for the public APIs provided by the Resource Manager define four categories of parameters:

- Standard IN
- Standard OUT
- Procedure specific IN
- Procedure specific OUT

Standard IN and OUT parameters are specified by the Oracle Applications business object API Coding Standards, and are discussed in the following sections.

Procedure specific IN and OUT parameter are related to the API being specified, and are discussed with that individual API.

## Standard IN Parameters

The following table describes standard IN parameters, which are common to all public APIs provided by Resource Manager.

### ***Standard IN Parameters***

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number (provided in the documentation).

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	Yes	<p>The valid values for this parameter are:</p> <ul style="list-style-type: none"> <li>• True = FND_API.G_TRUE</li> <li>• False = FND_API.G_FALSE</li> <li>• Default = FND_API.G_FALSE</li> </ul> <p>If set to true, then the API makes a call to <i>fnd_msg_pub.initialize</i> to initialize the message stack. To set to true, use the value, "T".</p> <p>If set to false then the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called. To set to false, use the value, "F".</p>
p_commit	VARCHAR2(1)	No	<p>The valid values for this parameter are:</p> <ul style="list-style-type: none"> <li>• True = FND_API.G_TRUE</li> <li>• False = FND_API.G_FALSE</li> <li>• Default = FND_API.G_FALSE</li> </ul> <p>If set to true, then the API commits before returning to the calling program. To set to true, use the value, "T".</p> <p>If set to false, then it is the calling program's responsibility to commit the transaction. To set to false, use the value, "F".</p>

## Standard OUT Parameters

The following table describes standard OUT parameters, which are common to all public APIs provided by Resource Manager.

**Note:** All standard OUT parameters are required.

#### **Standard OUT Parameters**

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: <ul style="list-style-type: none"><li>• FND_API. G_RET_STS_SUCCESS  Success: Indicates the API call was successful</li><li>• FND_API. G_RET_STS_ERROR  Expected Error: There is a validation error, or missing data error.</li><li>• FND_API. G_RET_STS_UNEXP_ER ROR  Unexpected Error: The calling program can not correct the error.</li></ul>
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Holds the encoded message if <i>x_msg_count</i> is equal to one.

## **Parameter Size**

Verify the size of the column from the base table for that column when passing a parameter of a specific length. For example, if you pass a NUMBER value, first query to find the exact value to pass. An incorrect value can cause the API call to fail.

## **Missing Parameter Attributes**

The following table describes optional IN parameters which are initialized to pre-defined values representing missing constants. These constants are defined for the

common PL/SQL data types and should be used in the initialization of the API formal parameters.

#### ***Initialized IN Parameters***

Parameter	Type	Initialized Value
g_miss_num	CONSTANT	NUMBER:= 9.99E125
g_miss_char	CONSTANT	VARCHAR2(1):= chr(0)
g_miss_date	CONSTANT	DATE:= TO_DATE('1','j');

These constants are defined in the package FND\_API in the file *fndpapis.pls*. All columns in a record definition are set to the G\_MISS\_X constant as defined for the data type.

## **Parameter Validations**

The following types of parameters are always validated during the API call:

- Standard IN
- Standard OUT
- Mandatory procedure specific IN
- Procedure specific OUT

## **Invalid Parameters**

If the API encounters any invalid parameters during the API call, then one of the following actions will occur:

- An exception is raised.
- An error message identifying the invalid parameter is generated.
- All API actions are cancelled.

## **Version Information**

It is mandatory that every API call pass a version number for that API as its first parameter (*p\_api\_version*).

This version number must match the internal version number of that API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

**Warning:** The currently supported version at this time is 1.0. Use only this for the API version number.

In addition, the object version number **must** be input for all update and delete APIs.

- If the *object\_version\_number* passed by the API matches that of the object in the database, then the update is completed.
- If the *object\_version\_number* passed by the API does not match that of the object in the database, then an error condition is generated.

## Status Messages

Every API must return one of the following states as parameter *x\_return\_status* after the API is called:

- S (Success)
- E (Error)
- U (Unexpected error)

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

Each state can be associated with a status message. The following table describes each state.

#### ***Status Message and Description***

---

<b>Status</b>	<b>Description</b>
S	<p>Indicates that the API performed all the operations requested by its caller.</p> <ul style="list-style-type: none"><li>• A success return status may or may not be accompanied by messages in the API message list.</li><li>• Currently, the Resource Manager APIs do not provide a message for a return status of success.</li></ul>
E	<p>Indicates that the API failed to perform one or more of the operations requested by its caller.</p> <p>An error return status is accompanied by one or more messages describing the error.</p>
U	<p>Indicates that the API encountered an error condition it did not expect, or could not handle, and that it is unable to continue with its regular processing.</p> <ul style="list-style-type: none"><li>• For example, certain programming errors such as attempting to divide by zero causes this error.</li><li>• These types of errors usually cannot be corrected by the user and requires a system administrator or application developer to correct.</li></ul>

---

#### **Warning and Information Messages**

In addition to these three types of possible status messages, you can also code the following additional message types:

- Warnings
- Information

To create a warning message, perform the following steps:

1. Create a global variable to be used to signal a warning condition. For example, this

could be similar to the following:

```
G_RET_STS_WARNING := 'W'
```

This global variable is not part of the FND\_API package.

2. Return this value if the warning condition is encountered. For example, using the same example as in step one, set up the following code in the API to process the warning condition:

```
x_return_status := G_RET_STS_WARNING
```

This code replaces the more usual:

```
x_return_status := fnd_api.g_ret_sts_unexp_error for "U"
```

3. If desired, perform a similar procedure to create Information messages.



## Resource Manager Public APIs

This chapter covers the following topics:

- Resource Manager Public Packages
- Resource Manager Public APIs
- Package JTF\_RS\_RESOURCE\_PUB
- Create\_Resource
- Update\_Resource
- Package JTF\_RS\_GROUPS\_PUB
- Create\_Resource\_Group
- Update\_Resource\_Group
- Package JTF\_RS\_SALESREPS\_PUB
- Create\_SalesRep
- Update\_SalesRep
- Messages and Notifications
- JTF\_RS\_RESOURCE\_PUB
- JTF\_RS\_GROUPS\_PUB
- JTF\_RS\_SALESREPS\_PUB
- Sample Code
- JTF\_RS\_RESOURCE\_PUB
- JTF\_RS\_GROUPS\_PUB
- JTF\_RS\_SALESREPS\_PUB

## Resource Manager Public Packages

There are three public Resource Manager packages:

- Package JTF\_RS\_RESOURCE\_PUB, page 27-3
- Package JTF\_RS\_GROUPS\_PUB, page 27-35
- Package JTF\_RS\_SALESREPS\_PUB, page 27-46

### Package JTF\_RS\_RESOURCE\_PUB

All public procedures (APIs) relating to creating or updating a resource are stored in package JTF\_RS\_RESOURCE\_PUB. This package contains the following APIs:

- Create\_Resource, page 27-3
- Update\_Resource, page 27-20

### Package JTF\_RS\_GROUPS\_PUB

All public procedures (APIs) relating to creating or updating a resource group are stored in package JTF\_RS\_GROUPS\_PUB. This package contains the following APIs:

- Create\_Resource\_Group, page 27-35
- Update\_Resource\_Group, page 27-41

### Package JTF\_RS\_SALESREPS\_PUB

All public procedures (APIs) relating to creating or updating a salesperson are stored in package JTF\_RS\_SALESREPS\_PUB. This package contains the following APIs:

- Create\_SalesRep, page 27-46
- Update\_SalesRep, page 27-52

## Resource Manager Public APIs

The following table describes the public APIs which are discussed in this chapter.

### **Resource Manager Public APIs**

---

<b>Procedure</b>	<b>Description</b>
Create_Resource, page 27-3	Creates all categories of resources like Employee, Party, Supplier Contact, Other, Partner, and To Be Hired.
Update_Resource, page 27-20	Updates resource attributes like Resource Name, Start Date, End Date, and User Identifier.
Create_Resource_Group, page 27-35	Creates a Resource Group.
Update_Resource_Group, page 27-41	Updates Resource Group attributes such as Group Name, Group Description, Start and End Date effectively.
Create_SalesRep, page 27-46	Creates a salesperson. The API verifies the resource ID parameter from the jtf_rs_resourc_.extns table.
Update_SalesRep, page 27-52	Updates SalesRep attributes like SalesRep Number, Sales Credit Type, Start and End Date effectively.

---

## **Package JTF\_RS\_RESOURCE\_PUB**

This package contains the following APIs:

- Create\_Resource, page 27-3
- Update\_Resource, page 27-20

## **Create\_Resource**

The Create\_Resource API creates all resource categories including Employee, Party, Supplier Contact, and Other. Before creating the resource, the Create\_Resource API first determines that it does not exist in the Resources table. After creating the resource, the API inserts Call Center, Compensation, and Account Receivables information that pertain to the resource into resource tables and generates the following unique identifiers:

- resource\_id
- resource\_number

## Resource Information versus Source Information

Within the Resource Manager, resource information and source information are two different types of information.

- **Resource** information (name, number, category, and similar items) is defined within the Resource Manager, or through its APIs. This information can be changed or modified at will.
- **Source** information (source\_id, source\_name, source\_number, and similar items) are pulled into the Resource Manager from another "source." Information extracted from a source external to the Resource Manager is not changed by the Resource Manager or its APIs.

Example sources include:

- Oracle Human Resources tables
- Oracle HZ Party tables

**Note:** The calling application must pass in the resource name to both p\_resource\_name and p\_source\_name. To ensure correct operation of the API, neither of these values can be NULL.

## Creating a Resource that is an Employee

When using the Create\_Resource API to create a resource that is an employee, the following input parameters must be set to null, otherwise the API will return an error message:

- p\_addresss\_id
- p\_contact\_id
- p\_managing\_emp\_id
- p\_managing\_emp\_num

## Procedure Specification

```

PROCEDURE create_resource
(
    p_api_version           in  number,
    p_init_msg_list          in  varchar2 default fnd_api.g_false,
    p_commit                 in  varchar2 default fnd_api.g_false,
    p_category                in  jtf_rs_resource_extns.category%type,
    p_source_id               in  jtf_rs_resource_extns.source_id%type
    default null,
    p_address_id              in  jtf_rs_resource_extns.address_id%type
    default null,
    p_contact_id               in  jtf_rs_resource_extns.contact_id%type
    default null,
    p_managing_emp_id         in  jtf_rs_resource_extns.
                                default null,
    managing_employee_id%type   in  per_employees_current_x.employee_num%
    type                      default null,
    p_start_date_active       in  jtf_rs_resource_extns.start_date_active%type,
    type                      default null,
    p_end_date_active         in  jtf_rs_resource_extns.end_date_active%type
    type                      default null,
    p_time_zone                in  jtf_rs_resource_extns.time_zone%type
    default null,
    p_cost_per_hr              in  jtf_rs_resource_extns.cost_per_hr%type
    default null,
    p_primary_language         in  jtf_rs_resource_extns.primary_language%type
    type                      default null,
    p_secondary_language        in  jtf_rs_resource_extns.
                                default null,
    secondary_language%type     in  jtf_rs_resource_extns.support_site_id%type
    type                      default null,
    p_ies_agent_login          in  jtf_rs_resource_extns.ies_agent_login%type
    type                      default null,
    p_server_group_id          in  jtf_rs_resource_extns.server_group_id%type
    type                      default null,
    p_interaction_center_name   in  varchar2 default null,
    p_assigned_to_group_id      in  jtf_rs_resource_extns.
                                default null,
    assigned_to_group_id%type    in  jtf_rs_resource_extns.cost_center%type
    default null,
    p_charge_to_cost_center     in  jtf_rs_resource_extns.
                                default null,
    charge_to_cost_center%type   in  jtf_rs_resource_extns.
                                default null,
    p_comp_currency_code        in  jtf_rs_resource_extns.
                                default null,
    compensation_currency_code%type  in  jtf_rs_resource_extns.
                                default null,
    p_commissionable_flag       in  jtf_rs_resource_extns.
                                default 'y',
    commissionable_flag%type     in  jtf_rs_resource_extns.hold_reason_code%type
    type                      default null,
    p_hold_payment              in  jtf_rs_resource_extns.hold_payment%type
    default 'n',
    p_comp_service_team_id      in  jtf_rs_resource_extns.
                                default null,
    comp_service_team_id%type    in  jtf_rs_resource_extns.user_id%type
    p_user_id                  in  jtf_rs_resource_extns.
                                default null,
    p_transaction_number         in  jtf_rs_resource_extns.
                                default null,
    transaction_number%type      out varchar2,
    x_return_status              out number,
    x_msg_count                 out varchar2,
    x_msg_data                  out jtf_rs_resource_extns.resource_id%type,
    x_resource_id                out jtf_rs_resource_extns.resource_number%type
    type,
    p_resource_name              in  jtf_rs_resource_extns_tl.resource_name%type
    type                      default null,

```

```

p_source_name          in jtf_rs_resource_extns.source_name%type,
p_source_number        in jtf_rs_resource_extns.source_number%type
default null,
p_source_job_title     in jtf_rs_resource_extns.source_job_title%
type      default null,
p_source_email         in jtf_rs_resource_extns.source_email%type
default null,
p_source_phone         in jtf_rs_resource_extns.source_phone%type
default null,
p_source_org_id        in jtf_rs_resource_extns.source_org_id%type
default null,
p_source_org_name      in jtf_rs_resource_extns.source_org_name%
type      default null,
p_source_address1      in jtf_rs_resource_extns.source_address1%
type      default null,
p_source_address2      in jtf_rs_resource_extns.source_address2%
type      default null,
p_source_address3      in jtf_rs_resource_extns.source_address3%
type      default null,
p_source_address4      in jtf_rs_resource_extns.source_address4%
type      default null,
p_source_city          in jtf_rs_resource_extns.source_city%type
default null,
p_source_postal_code   in jtf_rs_resource_extns.
source_postal_code%type
p_source_state          in jtf_rs_resource_extns.source_state%type
default null,
p_source_province       in jtf_rs_resource_extns.source_province%
type      default null,
p_source_county         in jtf_rs_resource_extns.source_county%type
default null,
p_source_country        in jtf_rs_resource_extns.source_country%
type      default null,
p_source_mgr_id         in jtf_rs_resource_extns.source_mgr_id%type
default null,
p_source_mgr_name       in jtf_rs_resource_extns.source_mgr_name%
type      default null,
p_source_business_grp_id in jtf_rs_resource_extns.
source_business_grp_id%type
p_source_business_grp_name in jtf_rs_resource_extns.
source_business_grp_name%type
p_source_first_name     in jtf_rs_resource_extns.source_first_name%
type      default null,
p_source_last_name      in jtf_rs_resource_extns.source_last_name%
type      default null,
p_source_middle_name    in jtf_rs_resource_extns.source_middle_name%
type      default null,
p_source_category        in jtf_rs_resource_extns.source_category%
type      default null,
p_source_status          in jtf_rs_resource_extns.source_status%type
default null,
p_source_office          in jtf_rs_resource_extns.source_office%type
default null,
p_source_location        in jtf_rs_resource_extns.source_location%
type      default null,
p_source_mailstop        in jtf_rs_resource_extns.source_mailstop%
type      default null,
p_user_name             varchar2,
p_source_mobile_phone   in jtf_rs_resource_extns.
source_mobile_phone%type
p_source_pager          in jtf_rs_resource_extns.source_pager%type
default null,
p_attribute1            in jtf_rs_resource_extns.attribute1%type
default null,
p_attribute2            in jtf_rs_resource_extns.attribute2%type
default null,

```

```

    p_attribute3           in  jtf_rs_resource_extns.attribute3%type
    default null,
    p_attribute4           in  jtf_rs_resource_extns.attribute4%type
    default null,
    p_attribute5           in  jtf_rs_resource_extns.attribute5%type
    default null,
    p_attribute6           in  jtf_rs_resource_extns.attribute6%type
    default null,
    p_attribute7           in  jtf_rs_resource_extns.attribute7%type
    default null,
    p_attribute8           in  jtf_rs_resource_extns.attribute8%type
    default null,
    p_attribute9           in  jtf_rs_resource_extns.attribute9%type
    default null,
    p_attribute10          in  jtf_rs_resource_extns.attribute10%type
    default null,
    p_attribute11          in  jtf_rs_resource_extns.attribute11%type
    default null,
    p_attribute12          in  jtf_rs_resource_extns.attribute12%type
    default null,
    p_attribute13          in  jtf_rs_resource_extns.attribute13%type
    default null,
    p_attribute14          in  jtf_rs_resource_extns.attribute14%type
    default null,
    p_attribute15          in  jtf_rs_resource_extns.attribute15%type
    default null,
    p_attribute_category   in  jtf_rs_resource_extns.
    attribute_category%type
    default null,
);

);

```

## Current Version

1.0

## Parameter Descriptions

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

### *Create\_Resource IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.

Parameter	Data Type	Required	Descriptions and Validations
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_category	jtf_rs_resource_extns. category%type	Yes	<p>Type of resource, valid types are:</p> <ul style="list-style-type: none"> <li>• EMPLOYEE</li> <li>• PARTNER</li> <li>• SUPPLIER_CON TACT</li> <li>• OTHER</li> <li>• PARTY</li> <li>• TBH</li> <li>• VENUE</li> </ul>
p_source_id	jtf_rs_resource_extns. resource_id%type	No	<p>Resource identifier, this parameter is only used with the following categories:</p> <ul style="list-style-type: none"> <li>• EMPLOYEE</li> <li>• VENUE</li> <li>• PARTY</li> <li>• SUPPLIER_CON TACT</li> <li>• PARTNER</li> </ul>

Parameter	Data Type	Required	Descriptions and Validations
p_address_id	jtf_rs_resource_extns. address_id%type	No	Resource address. When creating a resource that is an employee, this parameter must be set to NULL, otherwise, the API returns an error.
p_contact_id	jtf_rs_resource_extns. contact_id%type	No	Resource contact identifier. When creating a resource that is an employee, this parameter must be set to NULL, otherwise, the API returns an error.
p_managing_emp_id	jtf_rs_resource_extns. managing_employee_ id%type	No	Identifier for the manager of the resource. When creating a resource that is an employee, this parameter must be set to NULL, otherwise, the API returns an error.
p_managing_emp_nu m	per_employees_curre nt_x. employee_num%type	No	Employee number of the resource's manager. When creating a resource that is an employee, this parameter must be set to NULL, otherwise, the API returns an error.
p_start_date_active	jtf_rs_resource_extns. start_date_active% type	Yes	Date on which the resource becomes active. This value can not be NULL, and the start date must be less than the end date.

Parameter	Data Type	Required	Descriptions and Validations
p_end_date_active	jtf_rs_resource_extns. end_date_active% type	No	Date on which the resource is no longer active. If no end date is provided, the resource is active indefinitely.
p_time_zone	jtf_rs_resource_extns. time_zone %type	No	Time zone, this value must be a valid time zone as defined in table HZ_TIMEZONES.
p_cost_per_hr	jtf_rs_resource_extns. cost_per_hr %type	No	The salary cost per hour for this resource. This value is used in conjunction with the p_comp_currency_code parameter.
p_primary_language	jtf_rs_resource_extns. primary_language % type	No	The resource's primary language. This value must be a valid NLS language as defined in table FND_LANGUAGES.
p_secondary_language	jtf_rs_resource_extns. secondary_language %type	No	The resource's secondary language. This value must be a valid NLS language as defined in table FND_LANGUAGES.
p_support_site_id	jtf_rs_resource_extns. support_site_id%type	No	Value used by the Service applications.
p_ies_agent_login	jtf_rs_resource_extns. ies_agent_login%type	No	Value used by Interaction Center applications (if using Oracle Scripting).

Parameter	Data Type	Required	Descriptions and Validations
p_server_group_id	jtf_rs_resource_extns. server_group_id% type	No	Value used by Interaction Center applications.
p_interaction_center_ name	VARCHAR2	No	Value used by Interaction Center applications.
p_assigned_to_group _id	jtf_rs_resource_extns. assigned_to_group_i d%type	No	The group to which this resource is assigned.
p_cost_center	jtf_rs_resource_extns. cost_center%type	No	The cost center to which this resource is assigned.
p_charge_to_cost_cen ter	jtf_rs_resource_extns. charge_to_cost_center %type	No	Cost center to charge against, this may be different than the resource's current cost center.
p_comp_currency_co de	jtf_rs_resource_extns. compensation_curren cy_code	No	Compensation currency type, this value must be a valid currency code as listed in table FND_CURRENCIES.
			This value is used in conjunction with the p_cost_per_hour parameter.

Parameter	Data Type	Required	Descriptions and Validations
p_commissionable_flag	jtf_rs_resource_extns.%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - This resource is eligible for a commission.</li> <li>• No - This resource is not eligible for a commission.</li> </ul> <p>The default is Yes.</p>
p_hold_reason_code	jtf_rs_resource_extns.%type	No	<p>The reason that compensation is being withheld. This value must be one of the lookup codes for JTF_RS_HOLD_REASON_TYPE in table FND_LOOKUPS.</p>
p_hold_payment	jtf_rs_resource_extns.%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - Withhold compensation.</li> <li>• No - Do not withhold compensation.</li> </ul> <p>The default is No.</p>
p_comp_service_team_id	jtf_rs_resource_extns.%type	No	<p>The identifier for the team to which this resource belongs.</p>

Parameter	Data Type	Required	Descriptions and Validations
p_user_id	jtf_rs_resource_extns. user_id%type	No	<p>Specify this value only if the following categories are used:</p> <ul style="list-style-type: none"> <li>• EMPLOYEE</li> <li>• PARTY</li> <li>• 'SUPPLIER_CONTACT'</li> </ul> <p>If a category other than these three is specified, the API exits with an unexpected error condition.</p> <p>This value is validated against table FND_USERS.</p>
p_transaction_number	jtf_rs_resource_extns. transaction_number%type	No	Transaction identifier
p_resource_name	jtf_rs_resource_extns_. tl.resource_name%type	No	The calling application must pass the resource name.
p_source_name	jtf_rs_resource_extns. source_name%type	Yes	The calling application passes the source name.
p_source_number	jtf_rs_resource_extns. source_number%type	No	See Resource Information versus Source Information, page 27-4.
			See Resource Information versus Source Information, page 27-4.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_job_title	jtf_rs_resource_extns. source_job_title%type	No	See Resource Information versus Source Information, page 27-4.
p_source_email	jtf_rs_resource_extns. source_email%type	No	See Resource Information versus Source Information, page 27-4.
p_source_phone	jtf_rs_resource_extns. source_phone%type	No	See Resource Information versus Source Information, page 27-4.
p_source_org_id	jtf_rs_resource_extns. source_org_id%type	No	See Resource Information versus Source Information, page 27-4.
p_source_org_name	jtf_rs_resource_extns. source_org_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address1	jtf_rs_resource_extns. source_address1% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address2	jtf_rs_resource_extns. source_address2% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address3	jtf_rs_resource_extns. source_address3% type	No	See Resource Information versus Source Information, page 27-4.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_address4	jtf_rs_resource_extns. source_address4% type	No	See Resource Information versus Source Information, page 27-4.
p_source_city	jtf_rs_resource_extns. source_city%type	No	See Resource Information versus Source Information, page 27-4.
p_source_postal_code	jtf_rs_resource_extns. source_postal_code% type	No	See Resource Information versus Source Information, page 27-4.
p_source_state	jtf_rs_resource_extns. source_state%type	No	See Resource Information versus Source Information, page 27-4.
p_source_province	jtf_rs_resource_extns. source_province% type	No	See Resource Information versus Source Information, page 27-4.
p_source_county	jtf_rs_resource_extns. source_county%type	No	See Resource Information versus Source Information, page 27-4.
p_source_country	jtf_rs_resource_extns. source_country%type	No	See Resource Information versus Source Information, page 27-4.
p_source_mgr_id	jtf_rs_resource_extns. source_mgr_id%type	No	See Resource Information versus Source Information, page 27-4.

Parameter	Data Type	Required	Descriptions and Validations
p_source_mgr_name	jtf_rs_resource_extns. source_mgr_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_business_gr p_id	jtf_rs_resource_extns. source_business_grp_ id%type	No	See Resource Information versus Source Information, page 27-4.
p_source_business_gr p_name	jtf_rs_resource_extns. source_business_grp_ name%type	No	See Resource Information versus Source Information, page 27-4.
p_source_first_name	jtf_rs_resource_extns. source_first_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_last_name	jtf_rs_resource_extns. source_last_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_middle_na me	jtf_rs_resource_extns. source_middle_name %type	No	See Resource Information versus Source Information, page 27-4.
p_source_category	jtf_rs_resource_extns. source_category% type	No	See Resource Information versus Source Information, page 27-4.
p_source_status	jtf_rs_resource_extns. source_status%type	No	See Resource Information versus Source Information, page 27-4.

Parameter	Data Type	Required	Descriptions and Validations
p_source_office	jtf_rs_resource_extns. source_office%type	No	See Resource Information versus Source Information, page 27-4.
p_source_location	jtf_rs_resource_extns. source_location%type	No	See Resource Information versus Source Information, page 27-4.
p_source_mailstop	jtf_rs_resource_extns. source_mailstop% type	No	See Resource Information versus Source Information, page 27-4.
p_user_name	VARCHAR2	Yes	The identifier of the person entering values into the source.
p_source_mobile_phon e	jtf_rs_resource_extns. source_mobile_phone %type	No	See Resource Information versus Source Information, page 27-4.
p_source_pager	jtf_rs_resource_extns. source_pager%type	No	See Resource Information versus Source Information, page 27-4.
p_attribute1	in jtf_rs_resource_extns. attribute1%type	No	Descriptive flexfield segment, default = null.
p_attribute2	in jtf_rs_resource_extns. attribute2%type	No	Descriptive flexfield segment, default = null.
p_attribute3	in jtf_rs_resource_extns. attribute3%type	No	Descriptive flexfield segment, default = null.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_attribute4	in jtf_rs_resource_extns. attribute4%type	No	Descriptive flexfield segment, default = null.
p_attribute5	in jtf_rs_resource_extns. attribute5%type	No	Descriptive flexfield segment, default = null.
p_attribute6	in jtf_rs_resource_extns. attribute6%type	No	Descriptive flexfield segment, default = null.
p_attribute7	in jtf_rs_resource_extns. attribute7%type	No	Descriptive flexfield segment, default = null.
p_attribute8	in jtf_rs_resource_extns. attribute8%type	No	Descriptive flexfield segment, default = null.
p_attribute9	in jtf_rs_resource_extns. attribute9%type	No	Descriptive flexfield segment, default = null.
p_attribute10	in jtf_rs_resource_extns. attribute10%type	No	Descriptive flexfield segment, default = null.
p_attribute11	in jtf_rs_resource_extns. attribute11%type	No	Descriptive flexfield segment, default = null.
p_attribute12	in jtf_rs_resource_extns. attribute12%type	No	Descriptive flexfield segment, default = null.
p_attribute13	in jtf_rs_resource_extns. attribute13%type	No	Descriptive flexfield segment, default = null.

Parameter	Data Type	Required	Descriptions and Validations
p_attribute14	in jtf_rs_resource_extns. attribute14%type	No	Descriptive flexfield segment, default = null.
p_attribute15	in jtf_rs_resource_extns. attribute15%type	No	Descriptive flexfield segment, default = null.
p_attribute_category	in jtf_rs_resource_extns. attribute_category	No	Descriptive flexfield structure definition column, default = null.

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

#### ***Create\_Resource OUT Parameters***

Parameter	Data type	Description
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_resource_id	jtf_rs_resource_extns. resource_id %type	The automatically generated resource identifier that is created through an internal call to JTF_RS_RESOURCE_EXTNS_S.
x_resource_number	jtf_rs_resource_extns. resource_number %type	The automatically generated resource number that is created through an internal call to JTF_RS_RESOURCE_NUMBER_S.

## **Update\_Resource**

The Update\_Resource API updates resource attributes such as Resource Name, Start Date, End Date, and User ID. Before updating these attributes, the API first verifies that the resource exists in the Resources table. The Update\_Resource API updates resource information in the Call Center, Compensation, and the Account Receivables tables.

## Procedure Specification

```
PROCEDURE update_resource
(
    p_api_version           in  number,
    p_init_msg_list         in  varchar2
default fnd_api.g_false,
    p_commit                in  varchar2
default fnd_api.g_false,
    p_resource_id            in  jtf_rs_resource_extns.resource_id%type,
    p_resource_number          in  jtf_rs_resource_extns.resource_number%type,
    p_managing_emp_id          in  jtf_rs_resource_extns.managing_employee_id%
type      default fnd_api.g_miss_num,
    p_start_date_active        in  jtf_rs_resource_extns.start_date_active%type
type      default fnd_api.g_miss_date,
    p_end_date_active          in  jtf_rs_resource_extns.end_date_active%type
default fnd_api.g_miss_date,
    p_time_zone               in  jtf_rs_resource_extns.time_zone%type
default fnd_api.g_miss_num,
    p_cost_per_hr              in  jtf_rs_resource_extns.cost_per_hr%type
default fnd_api.g_miss_num,
    p_primary_language          in  jtf_rs_resource_extns.primary_language%type
default fnd_api.g_miss_char,
    p_secondary_language         in  jtf_rs_resource_extns.secondary_language%type
type      default fnd_api.g_miss_char,
    p_support_site_id            in  jtf_rs_resource_extns.support_site_id%type
default fnd_api.g_miss_num,
    p_ies_agent_login            in  jtf_rs_resource_extns.ies_agent_login%type
default fnd_api.g_miss_char,
    p_server_group_id            in  jtf_rs_resource_extns.server_group_id%type
default fnd_api.g_miss_num,
    p_assigned_to_group_id       in  jtf_rs_resource_extns.assigned_to_group_id%type
type      default fnd_api.g_miss_num,
    p_cost_center                in  jtf_rs_resource_extns.cost_center%type
default fnd_api.g_miss_char,
    p_charge_to_cost_center       in  jtf_rs_resource_extns.
charge_to_cost_center%type   default fnd_api.g_miss_char,
    p_comp_currency_code          in  jtf_rs_resource_extns.
compensation_currency_code%type
default fnd_api.g_miss_char,
    p_commissionable_flag          in  jtf_rs_resource_extns.
commissionable_flag%type     default fnd_api.g_miss_char,
    p_hold_reason_code            in  jtf_rs_resource_extns.hold_reason_code%type
type      default fnd_api.g_miss_char,
    p_hold_payment                 in  jtf_rs_resource_extns.hold_payment%type
default fnd_api.g_miss_char,
    p_comp_service_team_id         in  jtf_rs_resource_extns.
comp_service_team_id%type     default fnd_api.g_miss_num,
    p_user_id                     in  jtf_rs_resource_extns.user_id%type
default fnd_api.g_miss_num,
    p_resource_name                in  jtf_rs_resource_extns_tl.resource_name%type
type      default fnd_api.g_miss_char
    p_source_name                  in  jtf_rs_resource_extns.source_name%type,
    p_source_number                 in  jtf_rs_resource_extns.source_number%type
default fnd_api.g_miss_char,
    p_source_job_title              in  jtf_rs_resource_extns.source_job_title%type
type      default fnd_api.g_miss_char,
    p_source_email                  in  jtf_rs_resource_extns.source_email%type
default fnd_api.g_miss_char,
    p_source_phone                  in  jtf_rs_resource_extns.source_phone%type
default fnd_api.g_miss_char,
    p_source_org_id                 in  number
default fnd_api.g_miss_num,
    p_source_org_name                in  jtf_rs_resource_extns.source_org_name%type
type      default fnd_api.g_miss_char,
```

```

p_source_address1      in jtf_rs_resource_extns.source_address1%type
default fnd_api.g_miss_char,
  p_source_address2      in jtf_rs_resource_extns.source_address2%
type      default fnd_api.g_miss_char,
  p_source_address3      in jtf_rs_resource_extns.source_address3%
type      default fnd_api.g_miss_char,
  p_source_address4      in jtf_rs_resource_extns.source_address4%
type      default fnd_api.g_miss_char,
  p_source_city          in jtf_rs_resource_extns.source_city%type
default fnd_api.g_miss_char,
  p_source_postal_code   in jtf_rs_resource_extns.source_postal_code%
type      default fnd_api.g_miss_char,
  p_source_state          in jtf_rs_resource_extns.source_state%type
default fnd_api.g_miss_char,
  p_source_province       in jtf_rs_resource_extns.source_province%
type      default fnd_api.g_miss_char,
  p_source_county         in jtf_rs_resource_extns.source_county%type
default fnd_api.g_miss_char,
  p_source_country        in jtf_rs_resource_extns.source_country%type
default fnd_api.g_miss_char,
  p_source_mgr_id         in jtf_rs_resource_extns.source_mgr_id%type
default fnd_api.g_miss_num,
  p_source_mgr_name        in jtf_rs_resource_extns.source_mgr_name%
type      default fnd_api.g_miss_char,
  p_source_business_grp_id in jtf_rs_resource_extns.
source_business_grp_id%type
default fnd_api.g_miss_num,
  p_source_business_grp_name in jtf_rs_resource_extns.
source_business_grp_name%type
default fnd_api.g_miss_char,
  p_source_first_name      in jtf_rs_resource_extns.source_first_name%
type      default fnd_api.g_miss_char,
  p_source_last_name        in jtf_rs_resource_extns.source_last_name%type
default fnd_api.g_miss_char,
  p_source_middle_name      in jtf_rs_resource_extns.source_middle_name%
type      default fnd_api.g_miss_char,
  p_source_category         in jtf_rs_resource_extns.source_category%type
default fnd_api.g_miss_char,
  p_source_status           in jtf_rs_resource_extns.source_status%type
default fnd_api.g_miss_char,
  p_source_office            in jtf_rs_resource_extns.source_office%type
default fnd_api.g_miss_char,
  p_source_location          in jtf_rs_resource_extns.source_location%type
default fnd_api.g_miss_char,
  p_source_mailstop          in jtf_rs_resource_extns.source_mailstop%type
default fnd_api.g_miss_char,
  p_address_id              in jtf_rs_resource_extns.address_id%type
default fnd_api.g_miss_num,
  p_object_version_num      in out jtf_rs_resource_extns.
object_version_number%type,
  p_user_name                in varchar2
default fnd_api.g_miss_char,
  x_return_status            out varchar2,
  x_msg_count                out number,
  x_msg_data                  out varchar2,
  p_source_mobile_phone      in jtf_rs_resource_extns.source_mobile_phone%
type      default null,
  p_source_pager              in jtf_rs_resource_extns.source_pager%type
default null,
  p_attribute1                in jtf_rs_resource_extns.attribute1%type
default fnd_api.g_miss_char,
  p_attribute2                in jtf_rs_resource_extns.attribute2%type
default fnd_api.g_miss_char,
  p_attribute3                in jtf_rs_resource_extns.attribute3%type
default fnd_api.g_miss_char,
  p_attribute4                in jtf_rs_resource_extns.attribute4%type

```

```

        default  fnd_api.g_miss_char,
        p_attribute5           in  jtf_rs_resource_extns.attribute5%type
        default  fnd_api.g_miss_char,
        p_attribute6           in  jtf_rs_resource_extns.attribute6%type
        default  fnd_api.g_miss_char,
        p_attribute7           in  jtf_rs_resource_extns.attribute7%type
        default  fnd_api.g_miss_char,
        p_attribute8           in  jtf_rs_resource_extns.attribute8%type
        default  fnd_api.g_miss_char,
        p_attribute9           in  jtf_rs_resource_extns.attribute9%type
        default  fnd_api.g_miss_char,
        p_attribute10          in  jtf_rs_resource_extns.attribute10%type
        default  fnd_api.g_miss_char,
        p_attribute11          in  jtf_rs_resource_extns.attribute11%type
        default  fnd_api.g_miss_char,
        p_attribute12          in  jtf_rs_resource_extns.attribute12%type
        default  fnd_api.g_miss_char,
        p_attribute13          in  jtf_rs_resource_extns.attribute13%type
        default  fnd_api.g_miss_char,
        p_attribute14          in  jtf_rs_resource_extns.attribute14%type
        default  fnd_api.g_miss_char,
        p_attribute15          in  jtf_rs_resource_extns.attribute15%type
        default  fnd_api.g_miss_char,
        p_attribute_category    in  jtf_rs_resource_extns.attribute_category%
type    default  fnd_api.g_miss_char

);

```

## Current Version

1.0

## Parameter Descriptions

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

### *Update\_Resource IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.

Parameter	Data Type	Required	Descriptions and Validations
p_resource_id	jtf_rs_resource_extns.%type	Yes	This value is created through an internal call in the Create_Resource, page 27-3 API.
			The reference identifier is validated for existence in JTF_RS_RESOURCE_EXTNS.
p_resource_number	jtf_rs_resource_extns.%type	Yes	This value is created through an internal call in the Create_Resource, page 27-3 API.
			The reference number is validated for existence in JTF_RS_RESOURCE_EXTNS.
p_managing_emp_id	jtf_rs_resource_extns.%type	No	Identifier for the manager of the resource.
p_managing_emp_num	per_employees_current.%type	No	Employee number of the resource's manager.
p_start_date_active	jtf_rs_resource_extns.%type	No	Date on which the resource becomes active. This value can not be NULL, and the start date must be less than the end date.

Parameter	Data Type	Required	Descriptions and Validations
p_end_date_active	jtf_rs_resource_extns. end_date_active% type	No	Date on which the resource is no longer active. If no end date is provided, the resource is active indefinitely.
p_time_zone	jtf_rs_resource_extns. time_zone %type	No	Time zone, this value must be a valid time zone as defined in table HZ_TIMEZONES.
p_cost_per_hr	jtf_rs_resource_extns. cost_per_hr %type	No	The salary cost per hour for this resource.
p_primary_language	jtf_rs_resource_extns. primary_language % type	No	The resource's primary language. This value must be a valid NLS language as defined in table FND_LANGUAGES.
p_secondary_language	jtf_rs_resource_extns. secondary_language %type	No	The resource's secondary language. This value must be a valid NLS language as defined in table FND_LANGUAGES.
p_support_site_id	jtf_rs_resource_extns. support_site_id%type	No	Value used by Service applications.
p_ies_agent_login	jtf_rs_resource_extns. ies_agent_login%type	No	Value used by Interaction Center applications (if using Oracle Scripting).
p_server_group_id	jtf_rs_resource_extns. server_group_id% type	No	Value used by Interaction Center applications.

Parameter	Data Type	Required	Descriptions and Validations
p_assigned_to_group_id	jtf_rs_resource_extns.%type	No	The group to which this resource is assigned.
p_cost_center	jtf_rs_resource_extns.%type	No	The cost center to which this resource is assigned.
p_charge_to_cost_center	jtf_rs_resource_extns.%type	No	Cost center to charge against, this may be different than the resources current cost center.
p_comp_currency_code	jtf_rs_resource_extns.%type	No	Compensation currency type, this value must be a valid currency code as listed in view FND_CURRENCIES_ACTIVE_MONET_V (from table FND_CURRENCIES).
p_commissionable_flag	jtf_rs_resource_extns.%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - This resource is eligible for a commission.</li> <li>• No - This resource is not eligible for a commission.</li> </ul> <p>The default is Yes.</p>

Parameter	Data Type	Required	Descriptions and Validations
p_hold_reason_code	jtf_rs_resource_extns.%type	No	The reason that compensation is being withheld. This value must be one of the lookup codes for JTF_RS_HOLD_REASON_TYPE in table FND_LOOKUPS.
p_hold_payment	jtf_rs_resource_extns.%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - Withhold compensation.</li> <li>• No - Do not withhold compensation.</li> </ul> <p>The default is No.</p>
p_comp_service_team_id	jtf_rs_resource_extns.%type	No	The identifier for the team to which this resource belongs.

Parameter	Data Type	Required	Descriptions and Validations
p_user_id	jtf_rs_resource_extns. user_id%type	No	<p>Specify this value only if the following categories are used:</p> <ul style="list-style-type: none"> <li>• EMPLOYEE</li> <li>• PARTY</li> <li>• 'SUPPLIER_CONTACT'</li> </ul> <p>If a category other than these three is specified, the API exits with an unexpected error condition.</p> <p>This value is validated against table FND_USERS.</p>
p_resource_name	jtf_rs_resource_extns_ tl.resource_name% type	No	<p>The calling application must pass the resource name.</p> <p>This value can not be NULL.</p>
p_source_name	jtf_rs_resource_extns. source_name%type	Yes	<p>The calling application must pass the resource name.</p> <p>This value can not be NULL.</p>
p_source_number	jtf_rs_resource_extns. source_number%type	No	<p>See Resource Information versus Source Information, page 27-4.</p> <p>See Resource Information versus Source Information, page 27-4.</p>

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_job_title	jtf_rs_resource_extns. source_job_title%type	No	See Resource Information versus Source Information, page 27-4.
p_source_email	jtf_rs_resource_extns. source_email%type	No	See Resource Information versus Source Information, page 27-4.
p_source_phone	jtf_rs_resource_extns. source_phone%type	No	See Resource Information versus Source Information, page 27-4.
p_source_org_id	jtf_rs_resource_extns. source_org_id%type	No	See Resource Information versus Source Information, page 27-4.
p_source_org_name	jtf_rs_resource_extns. source_org_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address1	jtf_rs_resource_extns. source_address1% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address2	jtf_rs_resource_extns. source_address2% type	No	See Resource Information versus Source Information, page 27-4.
p_source_address3	jtf_rs_resource_extns. source_address3% type	No	See Resource Information versus Source Information, page 27-4.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_address4	jtf_rs_resource_extns. source_address4% type	No	See Resource Information versus Source Information, page 27-4.
p_source_city	jtf_rs_resource_extns. source_city%type	No	See Resource Information versus Source Information, page 27-4.
p_source_postal_code	jtf_rs_resource_extns. source_postal_code% type	No	See Resource Information versus Source Information, page 27-4.
p_source_state	jtf_rs_resource_extns. source_state%type	No	See Resource Information versus Source Information, page 27-4.
p_source_province	jtf_rs_resource_extns. source_province% type	No	See Resource Information versus Source Information, page 27-4.
p_source_county	jtf_rs_resource_extns. source_county%type	No	See Resource Information versus Source Information, page 27-4.
p_source_country	jtf_rs_resource_extns. source_country%type	No	See Resource Information versus Source Information, page 27-4.
p_source_mgr_id	jtf_rs_resource_extns. source_mgr_id%type	No	See Resource Information versus Source Information, page 27-4.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_mgr_name	jtf_rs_resource_extns. source_mgr_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_business_gr p_id	jtf_rs_resource_extns. source_business_grp_ id%type	No	See Resource Information versus Source Information, page 27-4.
p_source_business_gr p_name	jtf_rs_resource_extns. source_business_grp_ name%type	No	See Resource Information versus Source Information, page 27-4.
p_source_first_name	jtf_rs_resource_extns. source_first_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_last_name	jtf_rs_resource_extns. source_last_name% type	No	See Resource Information versus Source Information, page 27-4.
p_source_middle_na me	jtf_rs_resource_extns. source_middle_name %type	No	See Resource Information versus Source Information, page 27-4.
p_source_category	jtf_rs_resource_extns. source_category% type	No	See Resource Information versus Source Information, page 27-4.
p_source_status	jtf_rs_resource_extns. source_status%type	No	See Resource Information versus Source Information, page 27-4.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_source_office	jtf_rs_resource_extns. source_office%type	No	See Resource Information versus Source Information, page 27-4.
p_source_location	jtf_rs_resource_extns. source_location%type	No	See Resource Information versus Source Information, page 27-4.
p_source_mailstop	jtf_rs_resource_extns. source_mailstop% type	No	See Resource Information versus Source Information, page 27-4.
p_address_id	jtf_rs_resource_extns. address_id%type	No	See Resource Information versus Source Information, page 27-4.
p_object_version_nu m	jtf_rs_resource_extns. object_version numb er%type,	Yes	The object version number of the resource ID derives from the jtf_rs_resource_extns table.
p_user_name	VARCHAR2	No	The name of the person entering values into the source.
p_source_mobile_ph o ne	jtf_rs_resource_extns. source_mobile_phone %type	No	See Resource Information versus Source Information, page 27-4.
p_source_pager	jtf_rs_resource_extns. source_pager%type	No	See Resource Information versus Source Information, page 27-4.

Parameter	Data Type	Required	Descriptions and Validations
p_attribute1	in jtf_rs_resource_extns. attribute1%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute2	in jtf_rs_resource_extns. attribute2%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute3	in jtf_rs_resource_extns. attribute3%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute4	in jtf_rs_resource_extns. attribute4%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute5	in jtf_rs_resource_extns. attribute5%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute6	in jtf_rs_resource_extns. attribute6%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute7	in jtf_rs_resource_extns. attribute7%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute8	in jtf_rs_resource_extns. attribute8%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR

Parameter	Data Type	Required	Descriptions and Validations
p_attribute9	in jtf_rs_resource_extns. attribute9%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute10	in jtf_rs_resource_extns. attribute10%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute11	in jtf_rs_resource_extns. attribute11%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute12	in jtf_rs_resource_extns. attribute12%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute13	in jtf_rs_resource_extns. attribute13%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute14	in jtf_rs_resource_extns. attribute14%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute15	in jtf_rs_resource_extns. attribute15%type	No	Descriptive flexfield segment. Default = FND_API. G_MISS_CHAR
p_attribute_category	in jtf_rs_resource_extns. attribute_category	No	Descriptive flexfield structure definition column. Default = FND_API. G_MISS_CHAR

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

***Update\_Resource OUT Parameters***

<b>Parameter</b>	<b>Data Type</b>	<b>Descriptions</b>
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.

## **Package JTF\_RS\_GROUPS\_PUB**

This package contains the following APIs:

- Create\_Resource\_Group, page 27-35
- Update\_Resource\_Group, page 27-41

### **Create\_Resource\_Group**

The Create\_Resource Group API creates a Resource Group and generates the *group\_id* number.

## Procedure Specification

```
PROCEDURE create_resource_group
(
    p_api_version          in  number,
    p_init_msg_list         in  varchar2 default fnd_api.g_false,
    p_commit                in  varchar2 default fnd_api.g_false,
    p_group_name             in  jtf_rs_groups_vl.group_name%type,
    p_group_desc              in  jtf_rs_groups_vl.group_desc%type
    default null,
    p_exclusive_flag        in  jtf_rs_groups_vl.exclusive_flag%type
    default 'n',
    p_email_address           in  jtf_rs_groups_vl.email_address%type
    default null,
    p_start_date_active      in  jtf_rs_groups_vl.start_date_active%type,
    p_end_date_active        in  jtf_rs_groups_vl.end_date_active%type
    default null,
    p_accounting_code         in  jtf_rs_groups_vl.accounting_code%type
    default null,
    x_return_status            out varchar2,
    x_msg_count                  out number,
    x_msg_data                   out varchar2,
    x_group_id                     out jtf_rs_groups_vl.group_id%type,
    x_group_number                  out jtf_rs_groups_vl.group_number%type
);
```

## Current Version

1.0

## Parameter Descriptions

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

*Create\_Resource\_Group IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.

<b>Parameter</b>	<b>Data Type</b>	<b>Required</b>	<b>Descriptions and Validations</b>
p_group_name	jtf_rs_groups_vl. group_name%type	Yes	The name of the resource group.
p_group_desc	jtf_rs_groups_vl. group_desc%type	No	A description of the resource group.

Parameter	Data Type	Required	Descriptions and Validations
p_exclusive_flag	jtf_rs_groups_vl. exclusive_flag%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - Assign resources only to this group with a particular member role and group usage that is not assigned to any other exclusive group with the same role and usage in the same time frame.</li> <li>• No - Disable this feature.</li> </ul> <p>If p_exclusive_flag is set to yes, then you can assign resources only to this group with a particular member role and group usage that is not assigned to any other exclusive group with the same role and usage in the same time frame.</p> <p>Because you specify the resource in this group exclusively, the same resource with the same member roles cannot be assigned to</p>

Parameter	Data Type	Required	Descriptions and Validations
			another group during the overlapping time frame.
p_email_address	jtf_rs_groups_vl. email_address%type	No	The email address of the group owner.
p_start_date_active	jtf_rs_groups_vl. start_date_active% type	Yes	Date on which the resource group becomes active. This value can not be NULL, and the start date must be less than the end date.
p_end_date_active	jtf_rs_groups_vl. end_date_active% type	No	Date on which the resource group is no longer active. If no end date is provided, the group is active indefinitely.
p_accounting_code	jtf_rs_groups_vl. accounting_code% type	No	Account code, for internal use only.

Parameter	Data Type	Required	Descriptions and Validations
P_TIME_ZONE	NUMBER	No	<p>Indicates the time zone identifier.</p> <p>Validations:</p> <ul style="list-style-type: none"> <li>The time zone value is a foreign key reference to the 'upgrade_tz_id' column in 'fnl_timezones_v1'.</li> <li>The value that is inserted or updated into 'jtf_rs_groups_b.time_zone' table must be a valid time zone defined in the 'fnl_timezones_v1'</li> </ul>

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

#### ***Create\_Resource OUT Parameters***

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.

Parameter	Data Type	Descriptions
x_group_id	jtf_rs_groups_vl.group_id%type,	The automatically generated group identifier that is created through an internal call to JTF_RS_GROUPS_S.
x_group_number	jtf_rs_groups_vl.group_number%type	The automatically generated group number that is created through an internal call to JTF_RS_GROUP_NUMBER_S

## Update\_Resource\_Group

The Update\_Resource API updates resource group attributes such as Group Name, Group Description, Start and End Date effectively.

### Procedure Specification

```
PROCEDURE update_resource_group
(
    p_api_version          in number,
    p_init_msg_list        in varchar2
default fnd_api.g_false,
    p_commit                in varchar2
default fnd_api.g_false,
    p_group_id              in jtf_rs_groups_vl.group_id%type,
    p_group_number           in jtf_rs_groups_vl.group_number%type,
    p_group_name              in jtf_rs_groups_vl.group_name%type
default fnd_api.g_miss_char,
    p_group_desc              in jtf_rs_groups_vl.group_desc%type
default fnd_api.g_miss_char,
    p_exclusive_flag          in jtf_rs_groups_vl.exclusive_flag%type
default fnd_api.g_miss_char,
    p_email_address            in jtf_rs_groups_vl.email_address%type
default fnd_api.g_miss_char,
    p_start_date_active       in jtf_rs_groups_vl.start_date_active%type
default fnd_api.g_miss_date,
    p_end_date_active          in jtf_rs_groups_vl.end_date_active%type
default fnd_api.g_miss_date,
    p_accounting_code         in jtf_rs_groups_vl.accounting_code%type
default fnd_api.g_miss_char,
    p_object_version_num      in out jtf_rs_groups_vl.object_version_number%
type,
    x_return_status            out  varchar2,
    x_msg_count                out  number,
    x_msg_data                  out  varchar2
);
```

### Current Version

1.0

## Parameter Descriptions

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

### *Update\_Resource\_Group IN Parameters*

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_group_id	jtf_rs_groups_vl. group_id%type	Yes	The value is created through an internal call in the Create_Resource, page 27-3 API.
			This parameter is validated for existence in table JTF_RS_GROUPS_B.
p_group_number	jtf_rs_groups. group_id%type	No	The value is created through an internal call in the Create_Resource, page 27-3 API.
p_group_name	jtf_rs_groups_vl. group_name%type	Yes	The name of the resource group.
p_group_desc	jtf_rs_groups_vl. group_desc%type	No	A description of the resource group.

Parameter	Data Type	Required	Descriptions and Validations
p_exclusive_flag	jtf_rs_groups_vl. exclusive_flag%type	No	<p>Boolean value:</p> <ul style="list-style-type: none"> <li>• Yes - Assign resources only to this group with a particular member role and group usage that is not assigned to any other exclusive group with the same role and usage in the same time frame.</li> <li>• No - Disable this feature.</li> </ul> <p>If p_exclusive_flag is set to yes, then you can assign resources only to this group with a particular member role and group usage that is not assigned to any other exclusive group with the same role and usage in the same time frame.</p> <p>Because you specify the resource in this group exclusively, the same resource with the same member roles cannot be assigned to</p>

Parameter	Data Type	Required	Descriptions and Validations
			another group during the overlapping time frame.
			The default is No.
p_email_address	jtf_rs_groups_vl. email_address%type	No	The email address of the group owner.
p_start_date_active	jtf_rs_groups_vl. start_date_active% type	Yes	Date on which the resource group becomes active. This value can not be NULL, and the start date must be less than the end date.
p_end_date_active	jtf_rs_groups_vl. end_date_active% type	No	Date on which the resource group is no longer active. If no end date is provided, the resource is active indefinitely.
p_accounting_code	jtf_rs_groups_vl. accounting_code% type	No	Account Code, for internal use only.
p_object_version_nu m	jtf_rs_groups_vl. object_version_numb er%type	Yes	The object version number of the group ID derives from the jtf_rs_groups_extns table.

Parameter	Data Type	Required	Descriptions and Validations
P_TIME_ZONE	NUMBER	No	<p>Indicates the time zone identifier.</p> <p>Validations:</p> <ul style="list-style-type: none"> <li>§ The time zone value is a foreign key reference to the 'upgrade_tz_id' column in 'fnd_timezones_v l'.</li> <li>§ The value that is inserted or updated into 'jtf_rs_groups_b.time_zone' table must be a valid time zone defined in the 'fnd_timezones_v l'</li> </ul>

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

#### ***Update\_Resource\_Group OUT Parameters***

Parameter	Data Type	Descriptions
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.

## **Package JTF\_RS\_SALESREPS\_PUB**

This package contains the following APIs:

- Create\_SalesRep, page 27-46
- Update\_SalesRep, page 27-52

### **Create\_SalesRep**

The Create\_SalesRep API creates a resource of type Salesrep. The API verifies the resource ID parameter from the JTF\_RS\_RESOURC\_.EXTNS table which is created using the Create\_Resource API.

### **Setting the org\_context**

When using a script to call the Create\_Salesrep API, the following commands must be used to properly set the org\_context:

```
mo_global.set_org_context(999, NULL);
```

where 999 is the org\_id parameter value to which the context is set. This should be done every time you call the Create\_Salesrep API.

## Procedure Specification

```
PROCEDURE create_salesrep
(
    p_api_version          in  number,
    p_init_msg_list        in  varchar2  default fnd_api.g_false,
    p_commit                in  varchar2  default fnd_api.g_false,
    p_resource_id           in  jtf_rs_salesreps.resource_id%type,
    p_sales_credit_type_id in  jtf_rs_salesreps.sales_credit_type_id%
type,
    p_name                  in  jtf_rs_salesreps.name%type
default null,
    p_status                in  jtf_rs_salesreps.status%type
default null,
    p_start_date_active     in  jtf_rs_salesreps.start_date_active%type
default null,
    p_end_date_active       in  jtf_rs_salesreps.end_date_active%type
default sysdate,
    p_gl_id_rev             in  jtf_rs_salesreps.gl_id_rev%type
default null,
    p_gl_id_freight         in  jtf_rs_salesreps.gl_id_freight%type
default null,
    p_gl_id_rec              in  jtf_rs_salesreps.gl_id_rec%type
default null,
    p_set_of_books_id       in  jtf_rs_salesreps.set_of_books_id%type
default null,
    p_salesrep_number        in  jtf_rs_salesreps.salesrep_number%type
default null,
    p_email_address          in  jtf_rs_salesreps.email_address%type
default null,
    p_wh_update_date         in  jtf_rs_salesreps.wh_update_date%type
default null,
    p_sales_tax_geocode      in  jtf_rs_salesreps.sales_tax_geocode%type
default null,
    p_sales_tax_inside_city_limits in jtf_rs_salesreps.
sales_tax_inside_city_limits%type
default null,
    x_return_status           out  varchar2,
    x_msg_count               out  number,
    x_msg_data                out  varchar2,
    x_salesrep_id              out  jtf_rs_salesreps.salesrep_id%type
);

```

## Current Version

1.0

## Parameter Descriptions

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

**Create\_SalesRep IN Parameters**

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_resource_id	jtf_rs_salesreps. resource_id%type	Yes	This value is created through an internal call in the Create_Resource, page 27-3 API.
			The reference identifier is validated for existence in JTF_RS_RESOURCE_EXTNS.
p_sales_credit_type_id	jtf_rs_salesreps. sales_credit_type_id %type	Yes	This value is validated against table OE_SALES_CREDIT_TYPES.
p_name	jtf_rs_salesreps. name%type	No	The sales person's name.
p_status	jtf_rs_salesreps. status%type	No	The status of this salesperson. For example, this could be set to Active.

Parameter	Data Type	Required	Descriptions and Validations
p_start_date_active	jtf_rs_salesreps.start_date_active%type	No	Date on which the salesperson becomes active. This value can not be NULL, and the start date must be less than the end date.
p_end_date_active	jtf_rs_salesreps.end_date_active%type	No	The effective end date for the salespersons. If no end date is provided, the salesperson is active indefinitely.
p_gl_id_rev	jtf_rs_salesreps.gl_id_rev%type	No	Accounting flexfield used for Revenue accounts.
p_gl_id_freight	jtf_rs_salesreps.gl_id_frieght%type	No	Accounting flexfield used for Freight accounts.
			This value is used by Oracle Accounts Receivables only to determine the revenue accounts for invoices assigned to this salesperson.
			This value is used by Oracle Accounts Receivables only to determine the freight accounts for invoices assigned to this salesperson.

Parameter	Data Type	Required	Descriptions and Validations
p_gl_id_rec	jtf_rs_salesreps. gl_id_rec%type	No	Accounting flexfield used for Receivables accounts.
			This value is used by Oracle Accounts Receivables only to determine the receivable accounts for invoices assigned to this salesperson.
p_set_of_books_id	jtf_rs_salesreps. set_of_books_id% type	No	Set of books identifier, used by Oracle Accounts Receivables.
p_salesrep_number	jtf_rs_salesreps. salesrep_number% type	No	This typical alphanumeric designation is user generated.
p_email_address	jtf_rs_salesreps. email_address%type	No	Email address of the salesperson.
p_wh_update_date	jtf_rs_salesreps. wh_update_date% type	No	This date is sent to the data warehouse.
			This parameter is used for backwards compatibility with Oracle Accounts Receivables and is written to table RA_SALESREPS_ALL.

Parameter	Data Type	Required	Descriptions and Validations
p_sales_tax_geocode	jtf_rs_salesreps.sales_tax_geocode% type	No	Sales tax code, it associates the salesperson with a unique tax jurisdiction.  Use only if a sales tax vendor is also installed.
p_sales_tax_inside_cit_y_limits	jtf_rs_salesreps.sales_tax_inside_city_limits% type	No	Indicates that the tax jurisdiction for this address is within city limits.  Use only if a value for p_sales_tax_geocode is supplied.

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

#### *Create\_SalesRep OUT Parameters*

Parameter	Data Type	Description
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_salesrep_id	jtf_rs_salesrep.salesrep_id% type	The automatically generated salesperson identifier that is created through an internal call to JTF_RS_SALESREPS_S.

## Update\_SalesRep

The Update\_SalesRep API updates SalesRep attributes such as Salesperson Number, Sales Credit Type, Start and End Date.

### Setting the org\_context

When using a script to call the the Update\_Salesrep API, the following commands must be used to properly set the org\_context:

```
mo_global.set_org_context(999, NULL);
```

where 999 is the org\_id parameter value to which the context is set. This should be done every time you call the Update\_Salesrep API.

### Procedure Specification

```
PROCEDURE update_salesrep
(
    p_api_version           in  number,
    p_init_msg_list         in  varchar2
default  fnd_api.g_false,
    p_commit                in  varchar2
default  fnd_api.g_false,
    p_salesrep_id           in  jtf_rs_salesreps.salesrep_id%type,
    p_sales_credit_type_id  in  jtf_rs_salesreps.sales_credit_type_id%
type,
    p_name                  in  jtf_rs_salesreps.name%type
default  fnd_api.g_miss_char,
    p_status                in  jtf_rs_salesreps.status%type
default  fnd_api.g_miss_char,
    p_start_date_active     in  jtf_rs_salesreps.start_date_active%typ
default  fnd_api.g_miss_date,
    p_end_date_active       in  jtf_rs_salesreps.end_date_active%type
default  fnd_api.g_miss_date,
    p_gl_id_rev             in  jtf_rs_salesreps.gl_id_rev%type
default  fnd_api.g_miss_num,
    p_gl_id_freight         in  jtf_rs_salesreps.gl_id_freight%type
default  fnd_api.g_miss_num,
    p_gl_id_rec              in  jtf_rs_salesreps.gl_id_rec%type
default  fnd_api.g_miss_num,
    p_set_of_books_id       in  jtf_rs_salesreps.set_of_books_id%type
default  fnd_api.g_miss_num,
    p_salesrep_number        in  jtf_rs_salesreps.salesrep_number%type
default  fnd_api.g_miss_char,
    p_email_address          in  jtf_rs_salesreps.email_address%type
default  fnd_api.g_miss_char,
    p_wh_update_date        in  jtf_rs_salesreps.wh_update_date%type
default  fnd_api.g_miss_date,
    p_sales_tax_geocode      in  jtf_rs_salesreps.sales_tax_geocode%type
default  fnd_api.g_miss_char,
    p_sales_tax_inside_city_limits in  jtf_rs_salesreps.
sales_tax_inside_city_limits%type default  fnd_api.g_miss_char,
    p_org_id                 in  jtf_rs_salesreps.org_id%type,
    p_object_version_number   in  out  jtf_rs_salesreps.
object_version_number%type,
    x_return_status           out  varchar2,
    x_msg_count               out  number,
    x_msg_data                out  varchar2
);
```

**Current Version**

1.0

**Parameter Descriptions**

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

***Update\_SalesRep IN Parameters***

Parameter	Data Type	Required	Descriptions and Validations
p_api_version	NUMBER	Yes	See "Standard IN Parameters", page 26-3.
p_init_msg_list	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_commit	VARCHAR2	No	See "Standard IN Parameters", page 26-3.
p_salesrep_id	jtf_rs_salesreps. salesrep_id%type	Yes	This value is created through an internal call in the Create_SalesRep, page 27-46 API.
			If a valid value is not supplied, the API exits with an unexpected error condition.
p_sales_credit_type_id	jtf_rs_salesreps. sales_credit_type_id %type	Yes	This value is validated against table OE_SALES_CREDIT_TYPES.
p_name	jtf_rs_salesreps. name%type	No	The sales person's name.

Parameter	Data Type	Required	Descriptions and Validations
p_status	jtf_rs_salesreps. status%type	No	The status of this salesperson. For example, this could be set to Active.
p_start_date_active	jtf_rs_slaesreps. start_date_active% type	No	Date on which the salesperson becomes active. This value can not be NULL, and the start date must be less than the end date.
p_end_date_active	jtf_rs_salesreps. end_date_active% type	No	The effective end date for the salesperson. If no end date is provided, the salesperson is active indefinitely.
p_gl_id_rev	jtf_rs_salesreps. gl_id_freight%type	No	Accounting flexfield used for Revenue accounts.
p_gl_id_freight	jtf_rs_salesreps. gl_id_freight%type	No	Accounting flexfield used for Freight accounts.
			This value is used by Oracle Accounts Receivables only to determine the revenue accounts for invoices assigned to this salesperson.
			This value is used by Oracle Accounts Receivables only to determine the freight accounts for invoices assigned to this salesperson.

Parameter	Data Type	Required	Descriptions and Validations
p_gl_id_rec	jtf_rs_salesreps. gl_id_rec%type	No	Accounting flexfield used for Receivables accounts.
			This value is used by Oracle Accounts Receivables only to determine the receivable accounts for invoices assigned to this salesperson.
p_set_of_books_id	jtf_rs_salesreps. set_of_books_id% type	No	Parameter used by Oracle Accounts Receivables.
p_salesrep_number	jtf_rs_salesreps. salesrep_number% type	No	This typical alphanumeric designation is user generated.
p_email_address	jtf_rs_salesreps. email_address%type	No	Email address of the salesperson.
p_wh_update_date	jtf_rs_salesreps. wh_update_date% type	No	This date is sent to the data warehouse.
			This parameter is used for backwards compatibility with Oracle Accounts Receivables and is written to table RS_SALESREPS_ALL
p_sales_tax_geocode	jtf_rs_salesreps. sales_tax_geocode% type	No	Sales tax code, it associates the salesperson with a unique tax jurisdiction.
			Use only if a sales tax vendor is also installed.

Parameter	Data Type	Required	Descriptions and Validations
p_sales_tax_inside_cit_y_limits	jtf_rs_salesreps.sales_tax_inside_city_limits%type	No	Indicates that the tax jurisdiction for this address is within city limits.
p_org_id	jtf_rs_salesreps.org_id%type	Yes	Organization Identifier
p_object_version_number	jtf_rs_salesreps.object_version_number%type	Yes	The object version number of the salesrep ID derives from the JTF_RS_SALESREPS table.

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

#### ***Update\_SalesRep OUT Parameters***

Parameter	Data Type	Descriptions and Validations
x_return_status	VARCHAR2	See "Standard OUT Parameters", page 26-4.
x_msg_count	NUMBER	See "Standard OUT Parameters", page 26-4.
x_msg_data	VARCHAR2	See "Standard OUT Parameters", page 26-4.

## Messages and Notifications

The APIs contained in the Resource Manager public packages generate messages and notifications as needed. These messages are detailed in the following sections.

- JTF\_RS\_RESOURCE\_PUB, page 27-57

- JTF\_RS\_GROUPS\_PUB, page 27-66
- JTF\_RS\_SALESREPS\_PUB, page 27-70

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

## JTF\_RS\_RESOURCE\_PUB

This section describes the messages and notifications generated by the APIs contained in the JTF\_RS\_RESOURCE\_PUB package.

### Create\_Resource

The following table lists the messages and notifications generated by the Create\_Resource API.

#### *Create\_Resource Messages*

Number	Type	Name	Text
210209	E	JTF_RS_INVALID_EMP_RESOURCE_ID	Employee Resource ID &P_EMP_RESOURCE_ID is not Valid or the Resource is Inactive.
210211	E	JTF_RS_RESOURCE_CATEGORY_NULL	Resource Category should not be NULL.
210212	E	JTF_RS_INVALID_RESOURCE_CATEGORY	Resource Category &P_RESOURCE_CATEGORY is not Valid.
210213	E	JTF_RS_TIME_ZONE_NULL	Time Zone is Null.
210214	E	JTF_RS_INVALID_TIME_ZONE	Time Zone ID &P_TIME_ZONE_ID is not Valid.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210215	E	JTF_RS_LANGUAGE _NULL	Language is Null.
210216	E	JTF_RS_INVALID_L ANGUAGE	NLS Language &P_LANGUAGE is not Valid.
210224	U	JTF_RS_SOURCE_ID _NULL	Source Identifier should not be NULL.
210328	U	JTF_RS_USERID_ER ROR	For the specified Category, User ID should be NULL.
210229	U	JTF_RS_EMP_IDS_N OT_NULL	For Resource Category EMPLOYEE address_id and contact_id should be NULL.
210230	U	JTF_RS_ERR_ASSIG N_TO_GRP_ID	Assigned To Group ID, p_assigned_to_group _id is not Valid.
210231	U	JTF_RS_INVALID_FL AG_VALUE	Flag value should either be "Y" or "N."
210235	E	JTF_RS_SUPPORT_SI TE_NULL	Support Site should Not be Null.
210236	E	JTF_RS_INVALID_S UPPORT_SITE	Support Site &P_SUPPORT_SITE is Not Valid.
210237	E	JTF_RS_INVALID_SR V_GROUP_ID	Server Group ID &P_SERVER_GROUP _ID is Not Valid.
210238	E	JTF_RS_INVALID_SV R_GROUP_NAME	Interaction Center Name &P_SERVER_GROUP _NAME is Not Valid.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210239	E	JTF_RS_CURRENCY_NULL	Currency Code should Not be Null.
210240	E	JTF_RS_INVALID_CURRENCY	Currency Code &P_CURRENCY is Not Valid.
210241	E	JTF_RS_HOLD_REASON_CODE_NULL	Hold Reason Code should Not be Null.
210242	E	JTF_RS_INVALID_HOLD_REASON_CODE	Hold Reason Code &P_HOLD_REASON_CODE is Not Valid.
210243	E	JTF_RS_TEAM_NUL	Team ID and Team Number are Null.
210244	E	JTF_RS_INVALID_TEAM	Team ID &P_TEAM_ID is Not Valid.
210245	E	JTF_RS_INVALID_TEAM_NUMBER	Team Number &P_TEAM_NUMBER is Not Valid.
210259	E	JTF_RS_USER_ID_NULL	User ID cannot be Null.
210260	E	JTF_RS_INVALID_USER_ID	User ID &P_USER_ID is Not Valid.
210340	U	JTF_RS_INVALID_ROLE_RES_TYPE	Role Resource Type does not exist in JTF Objects.
210346	U	JTF_RS_INVALID_OBJECT_USER_CODE	Invalid Object User code.
210393	U	JTF_RS_UNEXP_ERROR	An unexpected error was encountered in &P_API_NAME.

Number	Type	Name	Text
210218	E	JTF_RS_TABLE_HANDLER_ERROR	Error in the Table Handler.
210265	E	JTF_RS_ERR_POST_CUST_USR_HOOK	Returned Error status from the Post Customer User Hook.
210266	E	JTF_RS_ERR_POST_VERT_USR_HOOK	Returned Error status from the Post Vertical User Hook.
210327	E	JTF_RS_ERR_POST_INI_USR_HOOK	Returned Error status from the Post Internal User Hook.
210267	E	JTF_RS_ERR_MSG_GENERATE_API	Returned Error status from the Message Generation API.
210393	E	JTF_RS_UNEXP_ERR_OR	Program Error: An unexpected error was encountered in &P_API_NAME.
210207	U	JTF_RS_RESOURCE_NULL	Resource identifier and Resource Number are NULL.
210219	U	JTF_RS_START_DATE_NULL	Start Date Active cannot be NULL.
210220	U	JTF_RS_OTHER_IDS_NOT_NULL	For Resource Category OTHER/TBH the Source identifier, Address identifier, Contact identifier, and Managing Emp identifier should be NULL.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210221	U	JTF_RS_PARTNER_IDS_NULL	For Resource Category PARTNER the Source_id should <b>not</b> be NULL.
210222	U	JTF_RS_INVALID_PARTNER_IDS	Partner does not exist for the passed Source_id.
210223	U	JTF_RS_ERR_PARTNER_CONTACT_ID	Partner contact id p_contact_id is not valid.
210225	U	JTF_RS_INVALID_SOURCE_ID	Source identifier p_source_id is not valid.
210226	U	JTF_RS_INVALID_PARTY_ADDRESS	Party Address identifier <Address_id> is Not Valid.
210227	U	JTF_RS_ERR_PARTY_CONTACT_ID	Party Contact identifier <Contact_id> is Not Valid.
210228	U	JTF_RS_SC_IDS_NOT_NULL	For Resource Category SUPPLIER_CONTACT Address_id and Contact_id should be NULL.
210374	U	JTF_RS_ERR_DUPLICATE_USER_ID	The user identifier <User_id> already exist.
380060	E	JTF_RS_NOT_PRIMARY_ADDR	The address ID passed in is not the primary address of the PARTY/PARTNER.

Number	Type	Name	Text
210383	U	JTF_RS_ERR_PARTNER_ADDRESS_ID	Partner address identifier p_address_id is not valid.

### Update\_Resource

The following table lists the messages and notifications generated by the Update\_Resource API.

#### *Update\_Resource Messages*

Number	Type	Name	Text
210204	U	JTF_RS_INVALID_RESOURCE	Resource identifier <Resource_id> is not valid or is inactive.
210208	U	JTF_RS_INVALID_RESOURCE_NUMBER	Resource Number <Resource_number> is not valid or the resource is inactive.
210209	E	JTF_RS_INVALID_EMP_RESOURCE_ID	Employee Resource ID &P_EMP_RESOURCE_ID is not Valid or the Resource is Inactive.
210213	E	JTF_RS_TIME_ZONE_NULL	Time Zone is Null.
210214	E	JTF_RS_INVALID_TIME_ZONE	Time Zone ID &P_TIME_ZONE_ID is not Valid.
210215	E	JTF_RS_LANGUAGE_NULL	Language is Null.

Number	Type	Name	Text
210216	E	JTF_RS_INVALID_LANGUAGE	NLS Language &P_LANGUAGE is not Valid.
210218	U	JTF_RS_TABLE_HANDLER_ERROR	Error in the Table Handler.
210219	U	JTF_RS_START_DATE_NULL	Start Date Active cannot be NULL.
210327	E	JTF_RS_ERR_POST_INTOUSRHOOK	Returned Error status from the Post Internal User Hook.
210230	U	JTF_RS_ERR_ASSIGN_TO_GRP_ID	Assigned To Group identifier <Assigned_to_group_id> is not valid.
210231	U	JTF_RS_INVALID_FLAG_VALUE	Flag value should either be "Y" or "N."
210232	E	JTF_RS_ERR_STDT_GREATER_EDDT	Start Date Active cannot be greater than the End Date Active.
210236	E	JTF_RS_INVALID_SUPPORT_SITE	Support Site &P_SUPPORT_SITE is Not Valid.
210237	E	JTF_RS_INVALID_SERVER_GROUP_ID	Server Group ID &P_SERVER_GROUP_ID is Not Valid.
210238	E	JTF_RS_INVALID_SERVER_GROUP_NAME	Interaction Center Name &P_SERVER_GROUP_NAME is Not Valid.
210239	E	JTF_RS_CURRENCY_NULL	Currency Code should Not be Null.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210240	E	JTF_RS_INVALID_CURRENCY	Currency Code &P_CURRENCY is Not Valid.
210241	E	JTF_RS_HOLD_REASON_CODE_NULL	Hold Reason Code should Not be Null.
210242	E	JTF_RS_INVALID_HOLD_REASON_CODE	Hold Reason Code &P_HOLD_REASON_CODE is Not Valid.
210243	E	JTF_RS_TEAM_NUL	Team ID and Team Number are Null.
210244	E	JTF_RS_INVALID_TEAM	Team ID &P_TEAM_ID is Not Valid.
210245	E	JTF_RS_INVALID_TEAM_NUMBER	Team Number &P_TEAM_NUMBER is Not Valid.
210254	E	JTF_RS_ROW_LOCK_ERROR	This record has been updated by another user. Please requery and try again.
210259	E	JTF_RS_USER_ID_N	User ID cannot be Null.
210260	E	JTF_RS_INVALID_USER_ID	User ID &P_USER_ID is Not Valid.
210265	E	JTF_RS_ERR_POST_CUST_USR_HOOK	Returned Error status from the Post Customer User Hook.
210266	E	JTF_RS_ERR_POST_VERTICAL_USR_HOOK	Returned Error status from the Post Vertical User Hook.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210267	E	JTF_RS_ERR_MSG_GENERATE_API	Returned Error status from the Message Generation API.
210275	E	JTF_RS_ERR_RES_R_OLE_START_DATE	Resource Start Date out of range for the role related start dates of the Resource.
210276	E	JTF_RS_ERR_RES_R_OLE_END_DATE	Resource End Date out of range for the role related End dates of the Resource.
210277	E	JTF_RS_ERR_RES_G_MBR_START_DATE	Resource Start Date out of range for the group member role related start dates for the Resource.
210278	E	JTF_RS_ERR_RES_G_MBR_END_DATE	Resource End Date out of range for the group member role related end dates for the Resource.
210279	E	JTF_RS_ERR_RES_T_MBR_START_DATE	Resource Start Date out of range for the Team Member role related start dates for the Resource, where the Team Member is a Resource.
210280	E	JTF_RS_ERR_RES_T_MBR_END_DATE	Resource End Date out of range for the Team Member role related end dates for the Resource, where the Team Member is a Resource.

Number	Type	Name	Text
210281	E	JTF_RS_ERR_RES_SR_P_START_DATE	Resource Start Date out of range for the salesperson related start dates of the resource.
210282	E	JTF_RS_ERR_RES_SR_P_END_DATE	Resource End Date out of range for the salesperson related End dates of the Resource.
210328	U	JTF_RS_USERID_ER_ROR	For the specified Category, User identifier should be NULL.
210374	U	JTF_RS_ERR_DUPLICATE_USER_ID	The user identifier <User_id> already exist.
210393	U	JTF_RS_UNEXP_ERR_OR	An unexpected error was encountered in &P_API_NAME.
380001	E	JTF_RS_XMLGEN_E_RR	Package JTF_USR_HKS is invalid.
380002	E	JTF_RS_JUHK_ERR	Package JTF_USR_HKS is invalid.
380060	E	JTF_RS_NOT_PRIMARY_ADDR	The address ID passed in is not the primary address of the PARTY/PARTNER.

## JTF\_RS\_GROUPS\_PUB

This section describes the messages and notifications generated by the APIs contained in the JTF\_RS\_GROUPS\_PUB package.

## Create\_Resource\_Group

The following table lists the messages and notifications generated by the Create\_Resource\_Group API.

### *Create\_Resource\_Group Messages*

Number	Type	Name	Text
210201	U	JTF_RS_GROUP_NAME_NULL	Group Name cannot be NULL.
210218	U	JTF_RS_TABLE_HANDLER_ERROR	Error in the Table Handler.
210219	U	JTF_RS_START_DATE_NULL	Start Date Active cannot be NULL.
210232	E	JTF_RS_ERR_STDT_GREATER_EDDT	Start Date Active cannot be greater than the End Date Active.
210261	E	JTF_RS_ERR_PRE_CUST_USR_HOOK	Returned Error status from the Pre Customer User Hook.
210262	E	JTF_RS_ERR_PRE_VERT_USR_HOOK	Returned Error status from the Pre Vertical User Hook.
210265	E	JTF_RS_ERR_POST_CUST_USR_HOOK	Returned Error status from the Post Customer User Hook.
210266	E	JTF_RS_ERR_POST_VERT_USR_HOOK	Returned Error status from the Post Vertical User Hook.
210267	E	JTF_RS_ERR_MSG_GENERATE_API	Returned Error status from the Message Generation API.

Number	Type	Name	Text
210327	E	JTF_RS_ERR_POST_IN_T_USR_HOOK	Returned Error status from the Post Internal User Hook.
210371	E	JTF_RS_ERR_PRE_IN_T_USR_HOOK	Returned Error status from the Pre Internal User Hook.

### **Update\_Resource\_Group**

The following table lists the messages and notifications generated by the Update\_Resource\_Group API.

#### ***Update\_Resource\_Group Messages***

Number	Type	Name	Text
210201	U	JTF_RS_GROUP_NAME_NULL	Group Name cannot be NULL.
210246	E	JTF_RS_ERR_ROLE_START_DATE	Group Start Date Out of Range for the Related Roles Start dates for the Group.
210247	E	JTF_RS_ERR_ROLE_END_DATE	Group End Date Out of Range for the Related Roles End dates for the Group.
210248	E	JTF_RS_ERR_GRP_MEMBER_START_DATE	Group Start Date Out of Range for the Group Member Role Related Start Dates.
210249	E	JTF_RS_ERR_GRP_MEMBER_END_DATE	Group End Date Out of Range for the Group Member Role Related End Dates.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210250	E	JTF_RS_ERR_GRP_R EL_START_DATE	Group Start Date Out of Range for the Group Relations Related Start Dates.
210251	E	JTF_RS_ERR_GRP_R EL_END_DATE	Group End Date Out of Range for the Group Relations Related End Dates.
210252	E	JTF_RS_ERR_TEAM_ MBR_START_DATE	Group Start Date Out of Range for the Team Member Role Related Start Dates for the Group, where the Team Member is a Group.
210253	E	JTF_RS_ERR_TEAM_ MBR_END_DATE	Group End Date Out of Range for the Team Member Role Related End Dates for the Group, where the Team Member is a Group.
210254	E	JTF_RS_ROW_LOCK_ _ERROR	This record has been updated by another user. Please requery and try again.
210261	E	JTF_RS_ERR_PRE_C UST_USR_HOOK	Returned Error status from the Pre Customer User Hook.
210262	E	JTF_RS_ERR_PRE_V ERT_USR_HOOK	Returned Error status from the Pre Vertical User Hook.

Number	Type	Name	Text
210264	E	JTF_RS_EXCLUSIVE_GROUP	Group usage cannot be created as one of the member dates overlap with another record for the same resource assigned to another exclusive group with the same usage in the same time period.
210266	E	JTF_RS_ERR_POST_VERT_USR_HOOK	Returned Error status from the Post Vertical User Hook.
210268	E	JTF_RS_INVALID_GROUP	Usage &P_USAGE is not setup for the Group ID &P_GROUP_ID.
210371	E	JTF_RS_ERR_PRE_INT_USR_HOOK	Returned Error status from the Pre Internal User Hook.

## **JTF\_RS\_SALESREPS\_PUB**

This section describes the messages and notifications generated by the APIs contained in the JTF\_RS\_SALESREPS\_PUB package.

### **Create\_SalesRep**

The following table lists the messages and notifications generated by the Create\_SalesRep API.

**Create\_SalesRep Messages**

Number	Type	Name	Text
210204	E	JTF_RS_INVALID_R_ESOURCE	Resource identifier <Resource_id> is not valid or is inactive.
210207	U	JTF_RS_RESOURCE_NULL	Resource identifier and Resource Number are NULL.
210208	E	JTF_RS_INVALID_R_ESOURCE_NUMBER	Resource Number <Resource_number> is not valid or the resource is inactive.
210218	U	JTF_RS_TABLE_HANDLER_ERROR	Error in the Table Handler.
210219	E	JTF_RS_START_DATE_NULL	Start Date Active cannot be null.
210232	E	JTF_RS_ERR_STDT_GREATER_EDDT	Start Date Active cannot be greater than the End Date Active.
210261	E	JTF_RS_ERR_PRE_CUST_USR_HOOK	Returned Error status from the Pre Customer User Hook.
210262	E	JTF_RS_ERR_PRE_VERT_USR_HOOK	Returned Error status from the Pre Vertical User Hook.
210265	E	JTF_RS_ERR_POST_CUST_USR_HOOK	Returned Error status from the Post Customer User Hook.
210266	E	JTF_RS_ERR_POST_VERT_USR_HOOK	Returned Error status from the Post Vertical User Hook.

Number	Type	Name	Text
210267	E	JTF_RS_ERR_MSG_GENERATE_API	Returned Error status from the Message Generation API.
210283	E	JTF_RS_ERR_SALESREP_NUMBER	Salesperson Number &P_SALESREP_NUMBER already exists.
210284	E	JTF_RS_SALESREP_NUMBER_NULL	Salesperson Number cannot be Null.
210285	E	JTF_RS_ERR_SALES_CREDIT_TYPE	Sales Credit Type identifier <Sales_credit_type_id> is invalid.
210286	E	JTF_RS_SALES_CREDIT_TYPE_NULL	Sales Credit Type identifier cannot be NULL.
210327	E	JTF_RS_ERR_POST_INT_USR_HOOK	Returned Error status from the Post Internal User Hook.
210371	E	JTF_RS_ERR_PRE_INT_USR_HOOK	Returned Error status from the Pre Internal User Hook.

### Update\_SalesRep

The following table lists the messages and notifications generated by the Update\_SalesRep API.

#### ***Update\_SalesRep Messages***

Number	Type	Name	Text
210218	E	JTF_RS_TABLE_HANDLER_ERROR	Error in the Table Handler.

<b>Number</b>	<b>Type</b>	<b>Name</b>	<b>Text</b>
210219	U	JTF_RS_START_DATE_NULL	Start Date Active cannot be NULL.
210232	E	JTF_RS_ERR_STDT_GREATER_EDDT	Start Date Active cannot be greater than the End Date Active.
210261	E	JTF_RS_ERR_PRE_CUST_USR_HOOK	Returned Error status from the Pre Customer User Hook.
210262	E	JTF_RS_ERR_PRE_VERT_USR_HOOK	Returned Error status from the Pre Vertical User Hook.
210265	E	JTF_RS_ERR_POST_CUST_USR_HOOK	Returned Error status from the Post Customer User Hook.
210266	E	JTF_RS_ERR_POST_VERT_USR_HOOK	Returned Error status from the Post Vertical User Hook.
210267	E	JTF_RS_ERR_MSG_GENERATE_API	Returned Error status from the Message Generation API.
210283	E	JTF_RS_ERR_SALESREP_NUMBER	Salesperson Number &P_SALESREP_NUMBER already exists.
210284	E	JTF_RS_SALESREP_NUMBER_NULL	Salesperson Number cannot be Null.
210285	E	JTF_RS_ERR_SALES_CREDIT_TYPE	Sales Credit Type identifier <Sales_credit_type_id> is invalid.
210286	E	JTF_RS_SALES_CREDIT_TYPE_NULL	Sales Credit Type identifier cannot be NULL.

Number	Type	Name	Text
210315	E	JTF_RS_SALESREP_ID_NULL	Salesperson identifier cannot be NULL.
210316	E	JTF_RS_INVALID_SALESREP_ID	Salesperson identifier <Salesrep_id> is invalid.
210327	E	JTF_RS_ERR_POST_IN_USR_HOOK	Returned Error status from the Post Internal User Hook.
210371	E	JTF_RS_ERR_PRE_IN_USR_HOOK	Returned Error status from the Pre Internal User Hook.

## Sample Code

This section contains SQL scripts that call the Resource Manager public APIs stored in the following packages and insert values as required:

- JTF\_RS\_RESOURCE\_PUB, page 27-74
- JTF\_RS\_GROUPS\_PUB, page 27-82
- JTF\_RS\_SALESREPS\_PUB, page 27-88

## JTF\_RS\_RESOURCE\_PUB

The SQL scripts in this section create and update a resource by calling the APIs contained in the JTF\_RS\_RESOURCE\_PUB package in succession and by providing them with the appropriate values.

### Create\_Resource

This script calls the Create\_Resource API and provides the values listed in the following table using the Create\_Resource IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

**Create Resource API Sample Code Variables**

---

<b>Variable</b>	<b>Description</b>
l_api_version	Version is 1.0.
l_category	Category is Employee.
l_source_id	Source ID is equal to 10001.
l_resource_name	Resource name is Rachel Abbott.
l_managing_emp_id	Her employee ID number is 1100.
l_user_id	User ID is 2294.
l_time_zone	Time zone is 92.
l_cost_per_hr	Her rate is \$25.00 per hour.
l_primary_language	The primary language is American English.
l_secondary_language	The secondary language is Candian French.
l_support_site_id	The support site ID number is 22917.
l_ies_agent_login	The agent login is "guest".
l_server_group_id	The server group ID number is 112.
l_interaction_center_name	The interaction center name for the group is "CCTAGENT01".

---

---

<b>Variable</b>	<b>Description</b>
l_assigned_to_group_id	The assigned group ID number is 1548.
l_cost_center	The cost center number is OA55.
l_charge_to_cost_center	The charge to the cost center is 250.
l_comp_currency_code	The currency code is AUD.
l_commissionable_flag	The flag is set to "Y".
l_hold_reason_code	The hold reason code is HRC-1.
l_hold_payment	The hold payment flag is set to "N".
l_comp_service_team_id	The comp service team ID is 10072.
l_source_name	The source name is Abbott, Ms. Rachel (Rachel).
l_source_number	The source number is 159.
l_source_job_title	The source title is SAL600.Sales Representative.
l_source_email	Her email address is rabbott@visionhr.com.
l_source_phone	Her phone number is 650-555-1212.

---

```

SET SERVEROUTPUT ON

declare

l_api_version          number
:= 1.0;
l_category              jtf_rs_resource_extns.category%type
:= 'EMPLOYEE';
l_source_id              jtf_rs_resource_extns.source_id%type
:= 10001;
l_resource_name          jtf_rs_resource_extns_tl.resource_name%type
:= 'Rachel Abbott';
l_managing_emp_id        jtf_rs_resource_extns.managing_employee_id%
type := 1100;
l_user_id                jtf_rs_resource_extns.user_id%type
:= 2294;
l_time_zone              jtf_rs_resource_extns.time_zone%type
:= 92;
l_cost_per_hr            jtf_rs_resource_extns.cost_per_hr%type
:= 25;
l_primary_language        jtf_rs_resource_extns.primary_language%type
:= 'AMERICAN';
l_secondary_language      jtf_rs_resource_extns.secondary_language%
type := 'CANADIAN FRENCH';
l_support_site_id         jtf_rs_resource_extns.support_site_id%type
:= 22917;
l_ies_agent_login         jtf_rs_resource_extns.ies_agent_login%type
:= 'guest';
l_server_group_id         jtf_rs_resource_extns.server_group_id%type
:= 112;
l_interaction_center_name ieo_svr_groups.group_name%type
:= 'CCTAGENT01';
l_assigned_to_group_id    jtf_rs_resource_extns.assigned_to_group_id%
type := 1548;
l_cost_center              jtf_rs_resource_extns.cost_center%type
:= '0A55';
l_charge_to_cost_center   jtf_rs_resource_extns.charge_to_cost_center%
type := '250';
l_comp_currency_code       jtf_rs_resource_extns.
compensation_currency_code%type := 'AUD';
l_commissionable_flag      jtf_rs_resource_extns.commissionable_flag%
type := 'Y';
l_hold_reason_code         jtf_rs_resource_extns.hold_reason_code%type
:= 'HRC-1';
l_hold_payment              jtf_rs_resource_extns.hold_payment%type
:= 'N';
l_comp_service_team_id     jtf_rs_resource_extns.comp_service_team_id%
type := 10072;
l_source_name              jtf_rs_resource_extns.source_name%type
:= 'Abbott, Ms. Rachel (Rachel)';
l_source_number             jtf_rs_resource_extns.source_number%type
:= 159;
l_source_job_title          jtf_rs_resource_extns.source_job_title%type
:= 'SAL600.Sales Representative';
l_source_email               jtf_rs_resource_extns.source_email%type
:= 'rabbott@visionhr.com';
l_source_phone               jtf_rs_resource_extns.source_phone%type
:= '650-555-1212';

l_return_status             varchar2(1);
l_msg_count                 number;
l_msg_data                  varchar2(2000);
l_resource_id                jtf_rs_resource_extns.resource_id%type;
l_resource_number             jtf_rs_resource_extns.resource_number%type;

l_msg_data1                 varchar2(2000);

```

```

l_msg_index_out      number;

begin

jtf_rs_resource_pub.create_resource
( P_API_VERSION          => l_api_version,
P_CATEGORY              => l_category,
P_START_DATE_ACTIVE    => sysdate,
P_END_DATE_ACTIVE      => null,
P_SOURCE_ID             => l_source_id,
P_RESOURCE_NAME         => l_resource_name,
P_MANAGING_EMP_ID       => l_managing_emp_id,
P_USER_ID               => l_user_id,
P_TIME_ZONE              => l_time_zone,
P_COST_PER_HR            => l_cost_per_hr,
P_PRIMARY_LANGUAGE       => l_primary_language,
P_SECONDARY_LANGUAGE     => l_secondary_language,
P_SUPPORT_SITE_ID        => l_support_site_id,
P_IES_AGENT_LOGIN        => l_ies_agent_login,
P_SERVER_GROUP_ID        => l_server_group_id,
P_INTERACTION_CENTER_NAME=> l_interaction_center_name,
P_ASSIGNED_TO_GROUP_ID   => l_assigned_to_group_id,
P_COST_CENTER             => l_cost_center,
P_CHARGE_TO_COST_CENTER  => l_charge_to_cost_center,
P_COMP_CURRENCY_CODE     => l_comp_currency_code,
P_COMMISSIONABLE_FLAG    => l_commissionable_flag,
P_HOLD_REASON_CODE       => l_hold_reason_code,
P_HOLD_PAYMENT            => l_hold_payment,
P_COMP_SERVICE_TEAM_ID   => l_comp_service_team_id,
P_SOURCE_NAME             => l_source_name,
P_SOURCE_NUMBER           => l_source_number,
P_SOURCE_JOB_TITLE        => l_source_job_title,
P_SOURCE_EMAIL             => l_source_email,
P_SOURCE_PHONE             => l_source_phone,
X_RETURN_STATUS           => l_return_status,
X_MSG_COUNT               => l_msg_count,
X_MSG_DATA                => l_msg_data,
X_RESOURCE_ID             => l_resource_id,
X_RESOURCE_NUMBER          => l_resource_number
);

if (l_return_status <> 'S') then
  if (fnd_msg_pub.count_msg > 0) then
    for i in 1..fnd_msg_pub.count_msg loop
      fnd_msg_pub.get
      (p_msg_index => i,
       p_data => l_msg_data,
       p_encoded => 'F',
       p_msg_index_out => l_msg_index_out
      );
      l_msg_data1 := l_msg_data1 || ' ' || l_msg_data;
    end loop;
    fnd_message.set_encoded(l_msg_data1);
    dbms_output.put_line(l_msg_data1);
  end if;
else
  dbms_output.put_line ('Return Status - ' || l_return_status);
  dbms_output.put_line ('Resource ID - ' || l_resource_id);
  dbms_output.put_line ('Resource Number - ' || l_resource_number);
end if;

end;
/

```

## **Update\_Resource**

This script calls the Update\_Resource API and provides the values listed in the following table using the Update\_Resource IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

### ***Update\_Resource API Sample Code Variables***

---

<b>Variable</b>	<b>Description</b>
l_api_version	Version is 1.0.
l_resource_id	The resource ID is 10000.
l_resource_number	The resource number is 10005.
l_resource_name	Resource name is Rachel Abbott.
l_managing_emp_id	Her employee ID number is 1100.
l_user_id	User ID is 2294.
l_time_zone	Time zone is 92.
l_cost_per_hr	Her rate is \$25.00 per hour.
l_primary_language	The primary language is American English.
l_secondary_language	The secondary language is Candian French.
l_support_site_id	The support site ID number is 22917.
l_ies_agent_login	The agent login is "guest".
l_server_group_id	The server group ID number is 112.

---

---

<b>Variable</b>	<b>Description</b>
l_assigned_to_group_id	The assigned group ID number is 1548.
l_cost_center	The cost center number is OA55.
l_charge_to_cost_center	The charge to the cost center is 250.
l_comp_currency_code	The currency code is AUD.
l_commissionable_flag	The flag is set to "Y".
l_hold_reason_code	The hold reason code is HRC-1.
l_hold_payment	The hold payment flag is set to "N".
l_comp_service_team_id	The comp service team ID is 10072.
l_source_name	The source name is Abbott, Ms. Rachel (Rachel).
l_source_number	The source number is 159.
l_source_job_title	The source title is SAL600.Sales Representative.
l_source_email	Her email address is rabbott@visionhr.com.
l_source_phone	Her phone number is 650-555-1212.

---

```

SET SERVEROUTPUT ON

declare

l_api_version          number
:= 1.0;
l_resource_id           jtf_rs_resource_extns.resource_id%type
:= 10000;
l_resource_number        jtf_rs_resource_extns.resource_number%type
:= '10005';
l_resource_name          jtf_rs_resource_extns_tl.resource_name%type
:= 'Rachel Abbott';
l_managing_emp_id        jtf_rs_resource_extns.managing_employee_id%
type := 1100;
l_user_id                jtf_rs_resource_extns.user_id%type
:= 2294;
l_time_zone               jtf_rs_resource_extns.time_zone%type
:= 92;
l_cost_per_hr             jtf_rs_resource_extns.cost_per_hr%type
:= 25;
l_primary_language        jtf_rs_resource_extns.primary_language%type
:= 'AMERICAN';
l_secondary_language       jtf_rs_resource_extns.secondary_language%
type := 'CANADIAN FRENCH';
l_support_site_id         jtf_rs_resource_extns.support_site_id%type
:= 22917;
l_ies_agent_login          jtf_rs_resource_extns.ies_agent_login%type
:= 'guest';
l_server_group_id         jtf_rs_resource_extns.server_group_id%type
:= 112;
l_assigned_to_group_id     jtf_rs_resource_extns.assigned_to_group_id%
type := 1548;
l_cost_center               jtf_rs_resource_extns.cost_center%type
:= '0A55';
l_charge_to_cost_center    jtf_rs_resource_extns.charge_to_cost_center%
type := '250';
l_comp_currency_code        jtf_rs_resource_extns.
compensation_currency_code%type := 'AUD';
l_commissionable_flag       jtf_rs_resource_extns.commissionable_flag%
type := 'Y';
l_hold_reason_code          jtf_rs_resource_extns.hold_reason_code%type
:= 'HRC-1';
l_hold_payment               jtf_rs_resource_extns.hold_payment%type
:= 'N';
l_comp_service_team_id      jtf_rs_resource_extns.comp_service_team_id%
type := 10072;
l_source_name               jtf_rs_resource_extns.source_name%type
:= 'Abbott, Ms. Rachel (Rachel)';
l_source_number              jtf_rs_resource_extns.source_number%type
:= 159;
l_source_job_title            jtf_rs_resource_extns.source_job_title%type
:= 'SAL600.Sales Representative';
l_source_email                 jtf_rs_resource_extns.source_email%type
:= 'rabbott@visionhr.com';
l_source_phone                  jtf_rs_resource_extns.source_phone%type
:= '650-555-1212';
l_object_version_number       jtf_rs_resource_extns.object_version_number%
type;

l_return_status                varchar2(1);
l_msg_count                     number;
l_msg_data                      varchar2(2000);
l_msg_data1                     varchar2(2000);
l_msg_index_out                  number;

begin

```

```

select object_version_number
into l_object_version_number
from jtf_rs_resource_extns
where resource_id = l_resource_id;

jtf_rs_resource_pub.update_resource
( P_API_VERSION          => l_api_version,
  P_RESOURCE_ID           => l_resource_id,
  P_RESOURCE_NUMBER        => l_resource_number,
  P_START_DATE_ACTIVE     => sysdate,
  P_END_DATE_ACTIVE       => null,
  P_RESOURCE_NAME          => l_resource_name,
  P_MANAGING_EMP_ID       => l_managing_emp_id,
  P_USER_ID                => l_user_id,
  P_TIME_ZONE               => l_time_zone,
  P_COST_PER_HR             => l_cost_per_hr,
  P_PRIMARY_LANGUAGE        => l_primary_language,
  P_SECONDARY_LANGUAGE      => l_secondary_language,
  P_SUPPORT_SITE_ID         => l_support_site_id,
  PIES_AGENT_LOGIN          => l_ies_agent_login,
  P_SERVER_GROUP_ID         => l_server_group_id,
  P_ASSIGNED_TO_GROUP_ID    => l_assigned_to_group_id,
  P_COST_CENTER              => l_cost_center,
  P_CHARGE_TO_COST_CENTER   => l_charge_to_cost_center,
  P_COMP_CURRENCY_CODE      => l_comp_currency_code,
  P_COMMISSIONABLE_FLAG     => l_commissionable_flag,
  P_HOLD_REASON_CODE        => l_hold_reason_code,
  P_HOLD_PAYMENT             => l_hold_payment,
  P_COMP_SERVICE_TEAM_ID    => l_comp_service_team_id,
  P_SOURCE_NAME              => l_source_name,
  P_SOURCE_NUMBER            => l_source_number,
  P_SOURCE_JOB_TITLE         => l_source_job_title,
  P_SOURCE_EMAIL              => l_source_email,
  P_SOURCE_PHONE              => l_source_phone,
  P_OBJECT_VERSION_NUM       => l_object_version_number,
  X_RETURN_STATUS             => l_return_status,
  X_MSG_COUNT                 => l_msg_count,
  X_MSG_DATA                  => l_msg_data
);

if (l_return_status <> 'S') then
  if (fnd_msg_pub.count_msg > 0) then
    for i in 1..fnd_msg_pub.count_msg loop
      fnd_msg_pub.get
      (p_msg_index => i,
       p_data => l_msg_data,
       p_encoded => 'F',
       p_msg_index_out => l_msg_index_out
      );
      l_msg_data1 := l_msg_data1 || ' ' || l_msg_data;
    end loop;
    fnd_message.set_encoded(l_msg_data1);
    dbms_output.put_line(l_msg_data1);
  end if;
else
  dbms_output.put_line ('Return Status - ' || l_return_status);
end if;

end;
/

```

## JTF\_RS\_GROUPS\_PUB

The SQL scripts in this section create and update a resource group by calling the APIs

contained in the JTF\_RS\_GROUPS\_PUB package in succession and by providing them with the appropriate values.

## Create\_Resource\_Group Sample Code

This script calls the Create\_Resource\_Group API and provides the values listed in the following table using the Create\_Resource\_Group IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

### ***Create\_Resource API Sample Code Variables***

Variable	Description
l_api_version	Version is 1.0.
l_category	Category is Employee.
l_source_id	Source ID is equal to 10001.
l_resource_name	Resource name is Rachel Abbott.
l_managing_emp_id	Her employee ID number is 1100.
l_user_id	User ID is 2294.
l_time_zone	Time zone is 92.
l_cost_per_hr	Her rate is \$25.00 per hour.
l_primary_language	The primary language is American English.
l_secondary_language	The secondary language is Candian French.
l_support_site_id	The support site ID number is 22917.
l_ies_agent_login	The agent login is "guest".
l_server_group_id	The server group ID number is 112.
l_interaction_center_name	The interaction center name for the group is "CCTAGENT01".
l_assigned_to_group_id	The assigned group ID number is 1548.

---

<b>Variable</b>	<b>Description</b>
l_cost_center	The cost center number is OA55.
l_charge_to_cost_center	The charge to the cost center is 250.
l_comp_currency_code	The currency code is AUD.
l_commissionable_flag	The flag is set to "Y".
l_hold_reason_code	The hold reason code is HRC-1.
l_hold_payment	The hold payment flag is set to "N".
l_comp_service_team_id	The comp service team ID is 10072.
l_source_name	The source name is Abbott, Ms. Rachel (Rachel).
l_source_number	The source number is 159.
l_source_job_title	The source title is SAL600.Sales Representative.
l_source_email	Her email address is rabbott@visionhr.com.
l_source_phone	Her phone number is 650-555-1212.

---

```

SET SERVEROUTPUT ON

declare

    l_api_version           number          := 1.0;
    l_group_name             jtf_rs_groups_vl.group_name%type := 'Sales Group';
    l_group_desc              jtf_rs_groups_vl.group_desc%type := 'Sales Group Representative';
    l_exclusive_flag         jtf_rs_groups_b.exclusive_flag%type := 'Y';
    l_email_address           jtf_rs_groups_b.email_address%type := 'sales_us@oracle.com';
    l_start_date_active      jtf_rs_groups_b.start_date_active%type := sysdate;
    l_end_date_active        jtf_rs_groups_b.end_date_active%type := null;
    l_accounting_code        jtf_rs_groups_b.accounting_code%type := 'Payment';

    l_return_status            varchar2(1);
    l_msg_count                number;
    l_msg_data                 varchar2(2000);
    l_group_id                  jtf_rs_groups_b.group_id%type;
    l_group_number               jtf_rs_groups_b.group_number%type;

    l_msg_data1                varchar2(2000);
    l_msg_index_out              number;

begin

    jtf_rs_groups_pub.create_resource_group
    (P_API_VERSION           => l_api_version,
     P_GROUP_NAME             => l_group_name,
     P_GROUP_DESC              => l_group_desc,
     P_EXCLUSIVE_FLAG          => l_exclusive_flag,
     P_ACCOUNTING_CODE        => l_accounting_code,
     P_EMAIL_ADDRESS            => l_email_address,
     P_START_DATE_ACTIVE       => l_start_date_active,
     P_END_DATE_ACTIVE          => l_end_date_active,
     X_RETURN_STATUS            => l_return_status,
     X_MSG_COUNT                => l_msg_count,
     X_MSG_DATA                  => l_msg_data,
     X_GROUP_ID                  => l_group_id,
     X_GROUP_NUMBER                => l_group_number
    );

    if (l_return_status <> 'S') then
        if (fnd_msg_pub.count_msg > 0) then
            for i in 1..fnd_msg_pub.count_msg loop
                fnd_msg_pub.get
                (p_msg_index => i,
                 p_data => l_msg_data,
                 p_encoded => 'F',
                 p_msg_index_out => l_msg_index_out
                );
                l_msg_data1 := l_msg_data1 || ' ' || l_msg_data;
            end loop;
            fnd_message.set_encoded(l_msg_data1);
            dbms_output.put_line(l_msg_data1);
        end if;
    else
        dbms_output.put_line ('Return Status - ' || l_return_status);
        dbms_output.put_line ('Group ID - ' || l_group_id);
        dbms_output.put_line ('Group Number - ' || l_group_number);
    end if;
end;

```

```

end if;
end;
/

```

### **Update\_Resource\_Group Sample Code**

This script calls the Update\_Resource\_Group API and provides the values listed in the following table using the Update\_Resource\_Group IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

#### ***Update\_Resource\_Group API Sample Code Variables***

<b>Variable</b>	<b>Description</b>
l_api_version	Version is 1.0.
l_group_id	The group ID is 10000.
l_group_desc	The description of the group is Sales Group Representative.
l_group_number	The group number is 10005.
l_group_name	The group name is "Sales Group".
l_group_desc	The group description is Sales Group Representative.
l_exclusive_flag	The exclusive flag is set to "Y".
l_email_address	The group's email address is sales_us@oracle.com.
l_start_date_active	The start date is the same as the sysdate.
l_end_date_active	The active end date is set to null.
l_accounting_code	The accounting code is "Payment".

```

SET SERVEROUTPUT ON

declare

    l_api_version           number          := 1.0;
    l_group_id               jtf_rs_groups_b.group_id%type      := 10000;
    l_group_number           jtf_rs_groups_b.group_number%type := '10005';
    l_group_name              jtf_rs_groups_vl.group_name%type := 'Sales Group';
    l_group_desc              jtf_rs_groups_vl.group_desc%type := 'Sales Group Representative';
    l_exclusive_flag          jtf_rs_groups_b.exclusive_flag%type := 'Y';
    l_email_address           jtf_rs_groups_b.email_address%type := 'sales_us@oracle.com';
    l_start_date_active       jtf_rs_groups_b.start_date_active%type := sysdate;
    l_end_date_active         jtf_rs_groups_b.end_date_active%type := null;
    l_accounting_code         jtf_rs_groups_b.accounting_code%type := 'Payment';

    l_object_version_number   jtf_rs_groups_b.object_version_number%type;
    l_return_status            varchar2(1);
    l_msg_count                number;
    l_msg_data                 varchar2(2000);
    l_msg_data1                varchar2(2000);
    l_msg_index_out             number;

begin

    select object_version_number
    into l_object_version_number
    from jtf_rs_groups_b
    where group_id = l_group_id;

    jtf_rs_groups_pub.update_resource_group
    (
        P_API_VERSION          => l_api_version,
        P_GROUP_ID              => l_group_id,
        P_GROUP_NUMBER           => l_group_number,
        P_GROUP_NAME              => l_group_name,
        P_GROUP_DESC              => l_group_desc,
        P_EXCLUSIVE_FLAG          => l_exclusive_flag,
        P_ACCOUNTING_CODE        => l_accounting_code,
        P_EMAIL_ADDRESS           => l_email_address,
        P_START_DATE_ACTIVE       => l_start_date_active,
        P_END_DATE_ACTIVE         => l_end_date_active,
        P_OBJECT_VERSION_NUM      => l_object_version_number,
        X_RETURN_STATUS            => l_return_status,
        X_MSG_COUNT                => l_msg_count,
        X_MSG_DATA                 => l_msg_data
    );

    if (l_return_status <> 'S') then
        if (fnd_msg_pub.count_msg > 0) then
            for i in 1..fnd_msg_pub.count_msg loop
                fnd_msg_pub.get
                (
                    p_msg_index => i,
                    p_data => l_msg_data,
                    p_encoded => 'F',
                    p_msg_index_out => l_msg_index_out
                );
                l_msg_data1 := l_msg_data1 || ' ' || l_msg_data;
            end loop;
        end if;
    end if;
end;

```

```

        end loop;
        fnd_message.set_encoded(l_msg_data1);
        dbms_output.put_line(l_msg_data1);
    end if;
else
    dbms_output.put_line ('Return Status - ' || l_return_status);
end if;

end;
/

```

## JTF\_RS\_SALESREPS\_PUB

The SQL scripts in this section create and update a sales representative by calling the APIs contained in the JTF\_RS\_SALESREPS\_PUB package in succession and by providing them with the appropriate values.

### Create\_SalesRep Sample Code

This script calls the Create\_SalesRep API and provides the values listed in the following table using the Create\_SalesRep IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

#### *Create\_SalesRep API Sample Code Variables*

---

Variable	Description
l_api_version	Version is 1.0.
l_resource_id	The resource ID number is 10001.
l_sales_credit_type_id	The sales credit type ID is 5.
l_name	The name is set to John Doe.
l_status	The status is set to "A".
l_start_date_active	The start date is equal to the sysdate.
l_end_date_active	The end date is set to null.
l_gl_id_rev	The global ID is 17736.
l_gl_id_freight	The global freight ID is 17734.
l_gl_id_rec	The global REC ID is 17703.

---

---

Variable	Description
l_set_of_books_id	The set of books ID is 23.
l_salesrep_number	The salesrep number is 42549.
l_email_address	The email address is john.doe@oracle.com
l_wh_update_date	Wh_update is equal to the sysdate.
l_sales_tax_geocode	The sales tax geocode is set to null.
l_sales_tax_inside_city_limits	The sales tax inside city limits is set to null.

---

```

SET SERVEROUTPUT ON

declare

l_api_version          number
:= 1.0;
l_salesrep_id           jtf_rs_salesreps.salesrep_id%type
:= 1000;
l_sales_credit_type_id jtf_rs_salesreps.sales_credit_type_id%
type                   := 5;
l_name                  jtf_rs_salesreps.name%type
:= 'John Doe';
l_status                jtf_rs_salesreps.status%type
:= 'A';
l_start_date_active     jtf_rs_salesreps.start_date_active%type
:= sysdate;
l_end_date_active       jtf_rs_salesreps.end_date_active%type
:= null;
l_gl_id_rev              jtf_rs_salesreps.gl_id_rev%type
:= 17736;
l_gl_id_freight         jtf_rs_salesreps.gl_id_freight%type
:= 17734;
l_gl_id_rec              jtf_rs_salesreps.gl_id_rec%type
:= 17703;
l_set_of_books_id        jtf_rs_salesreps.set_of_books_id%type
:= 23;
l_salesrep_number        jtf_rs_salesreps.salesrep_number%type
:= '42549';
l_email_address          jtf_rs_salesreps.email_address%type
:= 'john.doe@oracle.com';
l_wh_update_date         jtf_rs_salesreps.wh_update_date%type
:= sysdate;
l_sales_tax_geocode      jtf_rs_salesreps.sales_tax_geocode%type
:= null;
l_sales_tax_inside_city_limits    jtf_rs_salesreps.
sales_tax_inside_city_limits%type := null;
l_org_id                 jtf_rs_salesreps.org_id%type
:= 204;

l_object_version_number   jtf_rs_salesreps.object_version_number%
type;
l_return_status            varchar2(1);
l_msg_count                number;
l_msg_data                 varchar2(2000);
l_msg_data1                varchar2(2000);
l_msg_index_out             number;

begin

select object_version_number
into l_object_version_number
from jtf_rs_salesreps
where salesrep_id = l_salesrep_id;

jtf_rs_salesreps_pub.update_salesrep
(P_API_VERSION          => l_api_version,
P_SALESREP_ID           => l_salesrep_id,
P_SALES_CREDIT_TYPE_ID  => l_sales_credit_type_id,
P_NAME                  => l_name,
P_STATUS                => l_status,
P_START_DATE_ACTIVE     => l_start_date_active,
P_END_DATE_ACTIVE       => l_end_date_active,
P_GL_ID_REV              => l_gl_id_rev,
P_GL_ID_FREIGHT         => l_gl_id_freight,
P_GL_ID_REC              => l_gl_id_rec,
P_SET_OF_BOOKS_ID        => l_set_of_books_id,

```

```

P_SALESREP_NUMBER          => l_salesrep_number,
P_EMAIL_ADDRESS            => l_email_address,
P_WH_UPDATE_DATE           => l_wh_update_date,
P_SALES_TAX_GEOCODE         => l_sales_tax_geocode,
P_SALES_TAX_INSIDE_CITY_LIMITS =>
l_sales_tax_inside_city_limits,
P_ORG_ID                   => l_org_id,
P_OBJECT_VERSION_NUMBER     => l_object_version_number,
X_RETURN_STATUS              => l_return_status,
X_MSG_COUNT                  => l_msg_count,
X_MSG_DATA                   => l_msg_data
);

if (l_return_status <> 'S') then
  if (fnd_msg_pub.count_msg > 0) then
    for i in 1..fnd_msg_pub.count_msg loop
      fnd_msg_pub.get
        (p_msg_index => i,
         p_data => l_msg_data,
         p_encoded => 'F',
         p_msg_index_out => l_msg_index_out
        );
      l_msg_data1 := l_msg_data1 || ' ' || l_msg_data;
    end loop;
    fnd_message.set_encoded(l_msg_data1);
    dbms_output.put_line(l_msg_data1);
  end if;
else
  dbms_output.put_line ('Return Status - ' || l_return_status);
end if;

end;
/

```

### **Update\_SalesRep Sample Code**

This script calls the Update\_SalesRep API and provides the values listed in the following table using the Update\_SalesRep IN parameters. Calling applications can optionally insert more values than the ones listed in this section.

#### ***Update\_SalesRep API Sample Code Variables***

<b>Variable</b>	<b>Description</b>
l_api_version	Version is 1.0.
l_salesrep_id	The salesrep ID is 1000.
l_sales_credit_type_id	The sales credit type ID is 5.
l_name	The name is set to John Doe.
l_status	The status is set to "A".

Variable	Description
l_start_date_active	The start date is equal to the sysdate.
l_end_date_active	The end date is set to null.
l_gl_id_rev	The global ID is 17736.
l_gl_id_freight	The global freight ID is 17734.
l_gl_id_rec	The global REC ID is 17703.
l_set_of_books_id	The set of books ID is 23.
l_salesrep_number	The salesrep number is 42549.
l_email_address	The email address is john.doe@oracle.com
l_wh_update_date	Wh_update is equal to the sysdate.
l_sales_tax_geocode	The sales tax geocode is set to null.
l_sales_tax_inside_city_limits	The sales tax inside city limits is set to null.
l_org_id	The org ID is set to 204.

```

ET SERVEROUTPUT ON

declare

    l_api_version          number
    := 1.0;
    l_salesrep_id          jtf_rs_salesreps.salesrep_id%type
    := 1000;
    l_sales_credit_type_id jtf_rs_salesreps.sales_credit_type_id%
    type                  := 5;
    l_name                 jtf_rs_salesreps.name%type
    := 'John Doe';
    l_status                jtf_rs_salesreps.status%type
    := 'A';
    l_start_date_active     jtf_rs_salesreps.start_date_active%type
    := sysdate;
    l_end_date_active       jtf_rs_salesreps.end_date_active%type
    := null;
    l_gl_id_rev             jtf_rs_salesreps.gl_id_rev%type
    := 17736;
    l_gl_id_freight         jtf_rs_salesreps.gl_id_freight%type
    := 17734;
    l_gl_id_rec              jtf_rs_salesreps.gl_id_rec%type
    := 17703;
    l_set_of_books_id        jtf_rs_salesreps.set_of_books_id%type
    := 23;
    l_salesrep_number        jtf_rs_salesreps.salesrep_number%type
    := '42549';
    l_email_address           jtf_rs_salesreps.email_address%type
    := 'john.doe@oracle.com';
    l_wh_update_date          jtf_rs_salesreps.wh_update_date%type
    := sysdate;
    l_sales_tax_geocode        jtf_rs_salesreps.sales_tax_geocode%type
    := null;
    l_sales_tax_inside_city_limits jtf_rs_salesreps.
    sales_tax_inside_city_limits%type := null;
    l_org_id                  jtf_rs_salesreps.org_id%type
    := 204;

    l_object_version_number      jtf_rs_salesreps.object_version_number%
    type;
    l_return_status               varchar2(1);
    l_msg_count                   number;
    l_msg_data                     varchar2(2000);
    l_msg_data1                    varchar2(2000);
    l_msg_index_out                  number;

begin

    select object_version_number
    into l_object_version_number
    from jtf_rs_salesreps
    where salesrep_id = l_salesrep_id;

    jtf_rs_salesreps_pub.update_salesrep
    (
        P_API_VERSION          => l_api_version,
        P_SALESREP_ID           => l_salesrep_id,
        P_SALES_CREDIT_TYPE_ID   => l_sales_credit_type_id,
        P_NAME                  => l_name,
        P_STATUS                 => l_status,
        P_START_DATE_ACTIVE      => l_start_date_active,
        P_END_DATE_ACTIVE        => l_end_date_active,
        P_GL_ID_REV              => l_gl_id_rev,
        P_GL_ID_FREIGHT          => l_gl_id_freight,
        P_GL_ID_REC               => l_gl_id_rec,
        P_SET_OF_BOOKS_ID         => l_set_of_books_id,
    );

```

```

P_SALESREP_NUMBER                      => l_salesrep_number,
P_EMAIL_ADDRESS                         => l_email_address,
P_WH_UPDATE_DATE                        => l_wh_update_date,
P_SALES_TAX_GEOCODE                     => l_sales_tax_geocode,
P_SALES_TAX_INSIDE_CITY_LIMITS          =>
l_sales_tax_inside_city_limits,
P_ORG_ID                               => l_org_id,
P_OBJECT_VERSION_NUMBER                 => l_object_version_number,
X_RETURN_STATUS                          => l_return_status,
X_MSG_COUNT                            => l_msg_count,
X_MSG_DATA                             => l_msg_data
);

if (l_return_status <> 'S') then
  if (fnd_msg_pub.count_msg > 0) then
    for i in 1..fnd_msg_pub.count_msg loop
      fnd_msg_pub.get
        (p_msg_index => i,
         p_data => l_msg_data,
         p_encoded => 'F',
         p_msg_index_out => l_msg_index_out
        );
      l_msg_data1 := l_msg_data1 || ' '|| l_msg_data;
    end loop;
    fnd_message.set_encoded(l_msg_data1);
    dbms_output.put_line(l_msg_data1);
  end if;
else
  dbms_output.put_line ('Return Status - ' || l_return_status);
end if;

end;
/

```

---

# Index

## A

---

account contact role, 24-354

adapter

    API, 15-2

API

    adapter, 15-2

    bulk import, 14-1

    business event, 1-5

    business object

        attributes information overview, 24-2

        contact person information, 24-174

        customer account attributes, 24-3

        customer account contact attributes, 24-25

        customer account site attributes, 24-33

        customer account site use, 24-185

        customer profile, 24-200

        EDI attributes, 24-42

        EFT attributes, 24-51

        e-mail attributes, 24-59

        employment history, 24-220

        features, 22-5

        financial report, 24-228

        location, 24-237

        organization attributes, 24-67

        organization contact attributes, 24-96

        organization customer attributes, 24-108

        overview, 22-1

        party site attributes, 24-111

        person attributes, 24-121

        person customer attributes, 24-136

    phone attributes, 24-139

    seeded, 22-2

    SMS attributes, 24-148

    structure attributes, 24-174

    web attributes, 24-166

business object version 2

    customer account, 23-1

    customer account site, 23-3

    organization customer, 23-4

    person customer, 23-5

classification, 8-1

contact point, 6-1

contact preference, 9-1

customer account, 10-1

customer account role, 12-44

customer account site, 11-1

customer profile, 12-1

data quality management, 16-1

data quality management availability, 16-40

data quality management merge, 16-43

data quality management transformation, 16-41

debug messages

    Java, 2-10

    PL/SQL, 2-6

DQM index available, 16-40

DQM match rule available, 16-40

duplicate identification, 16-1

extensions, 17-1

find parties object, 16-42

general data formatting, 18-18

hierarchy retrieval, 7-37

human resources, 2-11

location, 5-31  
location service, 15-1  
merge, 16-43  
name and address formatting, 18-5  
organization information, 5-23  
overview, 1-1, 2-1  
parameters  
    Java, 2-7  
    PL/SQL, 2-4  
party, 3-1  
party contact, 4-1  
party information, 3-66  
party site, 5-46  
person information, 5-1  
phone parsing and formatting, 18-1  
private, 26-1  
public, 26-2  
    invalid parameters, 26-6  
    missing parameter attributes, 26-5  
    parameter size, 26-5  
    parameter specifications, 26-3  
    parameter validations, 26-6  
    standard IN parameters, 26-3  
    standard OUT parameters, 26-4  
    status messages, 26-7  
    version information, 26-6  
public, published, 26-2  
relationship, 7-11  
relationship type, 7-1  
resource manager, 26-1  
    public, 27-2  
sample code, 19-1  
source system management, 13-1  
tax assignment, 15-9  
third party data integration, 2-11  
transformation, 16-41

API procedures  
    business object  
        create, 22-6  
        get, 22-13  
        save, 22-11  
        update, 22-9

**B**

---

bank account use, 24-247  
bulk import

    API, 14-1  
business event, 1-5, 25-1  
    overview, 25-1  
    TCA, 25-3  
    TCA objects, 25-15  
business object  
    API  
        attributes information overview, 24-2  
        contact person information, 24-174  
        customer account attributes , 24-3  
        customer account contact attributes , 24-25  
        customer account site attributes , 24-33  
        customer account site use, 24-185  
        customer profile, 24-200  
        EDI attributes , 24-42  
        EFT attributes , 24-51  
        e-mail attributes , 24-59  
        employment history, 24-220  
        features, 22-5  
        financial report, 24-228  
        location, 24-237  
        organizataion contact attributes , 24-96  
        organization attributes , 24-67  
        organization customer attributes , 24-108  
        overview, 22-1  
        party site attributes , 24-111  
        person attributes , 24-121  
        person customer attributes , 24-136  
        phone attributes , 24-139  
        seeded, 22-2  
        SMS attributes , 24-148  
        structure attributes, 24-174  
        web attributes , 24-166  
    API procedures  
        create, 22-6  
        get, 22-13  
        save, 22-11  
        update, 22-9  
    customer account, 20-6  
    customer account contact, 20-11  
    customer account site, 20-14  
    EDI, 20-16  
    EFT, 20-19  
    e-mail, 20-21  
    granular entity attributes, 24-245  
        account contact role, 24-354

- bank account use, 24-247  
certification, 24-251  
citizenship, 24-254  
classification, 24-259  
contact preference, 24-264  
credit rating, 24-276  
customer account relationship, 24-293  
customer profile amount, 24-298  
education, 24-306  
extension, 24-311  
financial number, 24-315  
financial profile, 24-318  
organization contact role, 24-323  
party preference, 24-327  
party site use, 24-331  
payment method, 24-335  
person interest, 24-339  
person language, 24-343  
relationship, 24-348  
source system information, 24-358  
work class, 24-359
- organization, 20-23  
organization contact, 20-31  
organization customer, 20-34  
overview, 20-2  
party site, 20-35  
person, 20-38  
person customer, 20-43  
phone, 20-44  
SMS, 20-46  
telex, 20-49  
web, 20-51
- business object version 2
- API
- customer account, 23-1
  - customer account site, 23-3
  - organization customer, 23-4
  - person customer, 23-5
- customer account, 21-1  
customer account site, 21-2  
customer account site use, 21-2  
organization customer, 21-2  
overview, 21-1  
person customer, 21-3
- business structure
- contact person information, 20-53
  - customer account site use, 20-56
- customer profile, 20-61  
employment history, 20-65  
financial report, 20-68  
location, 20-70
- 
- C**
- certification, 24-251  
citizenship, 24-254  
classification, 24-259
- API, 8-1  
contact person information
- business structure, 20-53
- contact point
- API, 6-1
- contact preference, 24-264
- API, 9-1
- create\_resource, 27-3  
create\_resource\_group, 27-35  
create\_salesrep, 27-46  
credit rating, 24-276  
customer account
- API, 10-1
  - business object, 20-6
  - business object version 2, 21-1
- customer account contact
- business object, 20-11
- customer account relationship, 24-293
- customer account role
- API, 12-44
- customer account site
- API, 11-1
  - business object, 20-14
  - business object version 2, 21-2
- customer account site use
- business object version 2, 21-2
  - business structure, 20-56
- customer profile
- API, 12-1
  - business structure, 20-61
- customer profile amount, 24-298
- 
- D**
- data quality management
- API, 16-1
- data quality management availability
- API, 16-40

data quality management merge  
API, 16-43  
data quality management transformation  
API, 16-41  
DQM index available  
API, 16-40  
DQM match rule available  
API, 16-40  
duplicate identification  
API, 16-1

## E

---

EDI  
business object, 20-16  
education, 24-306  
EFT  
business object, 20-19  
e-mail  
business object, 20-21  
employment history  
business structure, 20-65  
extension, 24-311  
extensions  
API, 17-1

## F

---

financial number, 24-315  
financial profile, 24-318  
financial report  
business structure, 20-68  
find parties object  
API, 16-42

## G

---

general data formatting  
API, 18-18  
granular entity attributes, 24-245  
account contact role, 24-354  
bank account use, 24-247  
certification, 24-251  
citizenship, 24-254  
classification, 24-259  
contact preference, 24-264  
credit rating, 24-276  
customer account relationship, 24-293

customer profile amount, 24-298  
education, 24-306  
extension, 24-311  
financial number, 24-315  
financial profile, 24-318  
organization contact role, 24-323  
party preference, 24-327  
party site use, 24-331  
payment method, 24-335  
person interest, 24-339  
person language, 24-343  
relationship, 24-348  
source system information, 24-358  
work class, 24-359

## H

---

hierarchy retrieval  
API, 7-37  
human resources  
API, 2-11

## I

---

installation and setup, 25-1  
integration  
API, 2-11

## J

---

JTF\_RESOURCES\_PUB, 27-3  
JTF\_RS\_groups\_PUB  
create\_resource\_group, 27-35  
JTF\_RS\_Groups\_PUB  
update\_resource\_group, 27-41  
JTF\_RS\_GROUPS\_PUB, 27-2, 27-35  
JTF\_RS\_RESOURCES\_PUB, 27-2  
create\_resource, 27-3  
update resource, 27-20  
JTF\_RS\_SALESREPS\_PUB, 27-2, 27-46  
create\_salesrep, 27-46  
update\_salesrep, 27-52

## L

---

location  
API, 5-31  
business structure, 20-70  
location service

API, 15-1

## M

---

merge

API, 16-43

## N

---

name and address formatting

API, 18-5

## O

---

organization

business object, 20-23

organization contact

business object, 20-31

organization contact role, 24-323

organization customer

business object, 20-34

business object version 2, 21-2

organization information

API, 5-23

overview

business object version 2, 21-1

## P

---

parameter specifications, 26-3

invalid parameters, 26-6

missing parameter attributes, 26-5

parameter size, 26-5

parameter validations, 26-6

standard IN parameters, 26-3

standard OUT parameters, 26-4

status messages, 26-7

version information, 26-6

party

API, 3-1

party contact

API, 4-1

party information

API, 3-66

party preference, 24-327

party site

API, 5-46

business object, 20-35

party site use, 24-331

payment method, 24-335

person

business object, 20-38

person customer

business object, 20-43

business object version 2, 21-3

person information

API, 5-1

person interest, 24-339

person language, 24-343

phone

business object, 20-44

phone parsing and formatting

API, 18-1

## R

---

relationship, 24-348

API, 7-11

relationship type

API, 7-1

resource manager

API

overview, 26-1

public, 27-2

create\_resource, 27-3

create\_resource\_group, 27-35

create\_salesrep, 27-46

JTF\_RS\_GROUPS\_PUB, 27-35

JTF\_RS\_SALESREPS\_PUB, 27-46

messages and notifications, 27-56

JTF\_RS\_GROUPS\_PUB, 27-66

JTF\_RS\_RESOURCE\_PUB, 27-57

JTF\_RS\_SALESREPS\_PUB, 27-70

package

JTF\_RESOURCES\_PUB, 27-3

packages, 27-2

sample code, 27-74

JTF\_RS\_GROUPS\_PUB, 27-82

JTF\_RS\_RESOURCE\_PUB, 27-74

JTF\_RS\_SALESREPS\_PUB, 27-88

update\_resource, 27-20

update\_resource\_group, 27-41

update\_salesrep, 27-52

## S

---

sample code

API, 19-1  
SMS  
    business object, 20-46  
source system information, 24-358  
source system management  
    API, 13-1  
status messages, 26-7

## T

---

tax assignment  
    API, 15-9  
telex  
    business object, 20-49  
transformation  
    API, 16-41

## U

---

update\_resource, 27-20  
update\_resource\_group, 27-41  
update\_salesrep, 27-52

## W

---

web  
    business object, 20-51  
work class, 24-359