Oracle® Leads Management

Implementation and Administration Guide

Release 11*i*

Part No. B10593-01

March 2003

ORACLE®

Oracle Leads Management Implementation and Administration Guide, Release 11*i*

Part No. B10593-01

Contents

Part II Administering Oracle Leads Management

2 Administration Tasks for Oracle Leads Management

Send Us Your Comments

Oracle Leads Management Implementation and Administration Guide, Release 11*i*

Part No. B10593-01

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

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

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

Oracle Corporation
Oracle Your Product Name Here Documentation
Street Address, Mailstop
500 Oracle Parkway
Redwood Shores, CA 94065
USA

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

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

Preface

Intended Audience

Welcome to Release 11*i* of the Oracle Leads Management Implementation and Administration Guide.

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

- The principles and customary practices of your business area.
- Oracle Leads Management

  If you have never used Oracle Leads Management, Oracle suggests you attend one or more of the Oracle Leads Management training classes available through Oracle University.
- The Oracle Applications graphical user interface.

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

See Other Information Sources for more information about Oracle Applications product information.

How To Use This Guide

This document contains the information you need to understand and use Oracle Leads Management.

- Chapter 1 introduces you to the Oracle E-Business Suite.
- Chapter 2 explains the administration tasks in Oracle Leads Management.

Documentation Accessibility

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

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

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

Other Information Sources

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

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

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF). Online help patches are available on MetaLink.

Related Documentation

Oracle Leads Management shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other product documentation when you set up and use Oracle Leads Management.

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

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

Documents Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle Leads Management (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

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

Installation and System Administration

Oracle Applications Concepts

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

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your

installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Oracle Applications Supplemental CRM Installation Steps

This guide contains specific steps needed to complete installation of a few of the CRM products. The steps should be done immediately following the tasks given in the Installing Oracle Applications guide.

Upgrading Oracle Applications

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

Maintaining Oracle Applications

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

Oracle Applications System Administrator's Guide

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

Oracle Alert User's Guide

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

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer 6*i* forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

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

Other Implementation Documentation

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Oracle Leads Management. This manual details additional steps and setup considerations for implementing Oracle Leads Management with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use Oracle Leads Management with Oracle Applications' Multiple Organization support feature, so you can define and support different organization structures when running a single installation of Oracle Leads Management.

Oracle Workflow Administrator's Guide

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

Oracle Workflow Developer's Guide

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

Oracle Workflow User's Guide

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

Oracle Workflow API Reference

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

Oracle Applications Flexfields Guide

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

Oracle eTechnical Reference Manuals

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

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes APIs and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes APIs and open interfaces found in Oracle Order Management Suite.

Oracle Applications Message Reference Manual

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

Oracle CRM Application Foundation Implementation Guide

Many CRM products use components from CRM Application Foundation. Use this guide to correctly implement CRM Application Foundation.

Training and Support

Training

Oracle offers training courses to help you and your staff master Oracle Leads Management and reach full productivity quickly. You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your

facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Leads Management 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 Oracle8*i* server, and your hardware and software environment.

Oracle*MetaLink*

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

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

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

Do Not Use Database Tools to Modify Oracle Applications Data

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

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

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

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

About Oracle

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

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

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

Part I

Getting Started

This section of the book contains the following chapter:

1

Introduction

This chapter provides information on the following topics:

1.1 The Oracle E-Business Suite

The Oracle E-Business Suite is a comprehensive Web-based answer for business-to-business (B2B) and business-to-consumer (B2C) selling, marketing, and servicing through the Internet. The Oracle E-Business Suite consists of front-office Customer Relationship Management (CRM) applications and back-office Enterprise Resource Planning (ERP) applications. These applications automate marketing, sales, contracts, service, manufacturing, and supply chain processes as well as financial operations, project management, human resources operations, and business intelligence systems.

The Oracle E-Business Suite sits on a multi-layer platform which includes:

- Oracle 9*i* Database
- Oracle 9*i* Application Server
- Common Services and Components
- Oracle Internet Business Intelligence

Oracle 9*i* Database

All applications reside on the Oracle9*i* Database. The Oracle database drives enterprise E-Business applications, online transaction processing applications (OLTP), query-intensive data warehouses, and high capacity Web sites. Because the Oracle database is available on many different platforms, applications can scale from handheld to laptop to desktop to enterprise providing consistent information over multiple channels.

Oracle 9*i* Application Server

The Oracle 9*i* Application Server (Oracle 9*i*AS) is a middle-tier server which independently delivers the technology needed to build web sites and applications, create personalized portals, extract business intelligence, and manage a secure web site infrastructure.

Common Services and Components

All the applications can leverage the common infrastructure and services components. Functionality includes Oracle Forms, Oracle Reports, Oracle Application Object Library (AOL), the Oracle JDeveloper and Oracle Discoverer development tools, the coding and UI standards, and other functionality used by the applications.

For example, you can extend the applications according to your business needs using flexfields. You can create and assign responsibilities using the system administrator responsibility. Also, you can use Oracle Workflow to configure background processes and set up notifications so that all the appropriate managers and groups are notified.

Oracle Internet Business Intelligence

Above the E-Business Suite sits the Internet Business Intelligence application. This application integrate data from all of the E-Business Suite applications to provide key performance measurements, operating alerts, and management reports to every decision maker across the enterprise.

1.1.1 The Applications in the E-Business Suite

Customers can seamlessly share data from front-end applications (CRM) to backend applications (ERP). The CRM applications include:

- the Marketing Suite
- the Sales Suite

- the Contracts Suite
- the Service Suite
- the eCommerce Suite

The ERP applications include:

- Oracle Order Management
- Oracle Supply Chain Planning
- Oracle Manufacturing
- Oracle Financials
- Oracle Human Resources Management System

Customer Relation Management (CRM)

Companies use Oracle's CRM suite of applications to acquire, maintain, and enhance customer relationships, by assisting companies with marketing automation, sales force automation, contracts management, customer service and support, and business intelligence, in a multi-channel environment.

- The Marketing suite provides campaign planning and execution, budget management, list creation, reporting and analysis tools. Marketing professionals use the Oracle Marketing applications to drive quality leads to sales, to expand reach and to maximize marketing effectiveness by using a comprehensive set of marketing automation, analysis and multi-channel execution capabilities. The Marketing suite offers seamless integration with sales, service and operations.
- The Sales suite provides integrated tools for all those who are involved in the sales process, including field salespeople, telesales agents, distributors and resellers, customers purchasing over the Internet and sales executives.

  Armed with up-to-the minute information regarding customers, leads and opportunities, as well as forecasts and compensation plans and projections, managers can proactively and effectively manage a sales force while providing the sales people with the information needed to close sales. Using this information, the field sales force, telesales teams, resellers, and web storefronts can collaborate in closing more business together as one sales team.
- The Contracts suite enables authoring, executing and managing contracts, warranties and extended warranties which provides visibility to contract entitlements and proactively acting upon contractual commitments. Whether a

buyer or a seller, issuing contracts or receiving them, the Contracts suite automates the full contract life cycle.

- The Service suite manages service activities with the goals of profitability, employee productivity and complete customer satisfaction by addressing all service and support activities from initial contact with the customer through issue resolution. Automating service efforts can potentially transform an area that has historically proven to be a cost center into a revenue generator.

  This suite of applications provides customer support, field service and depot repair functionality. In addition, Oracle Services offers complete visibility into spare parts availability, logistics, service billing and customer contract entitlements. Oracle Customer Care provides full access to customer information from each touch point in the enterprise and to each customer care agent or other employees who interact with the customer. All of the Service products can be deployed across web, call center and mobile field channels.

- The eCommerce suite of products aids in establishing profitable long-term relationships with customers through one-to-one marketing and personalized shopping experiences as well as proactive support and self-service capabilities. Oracle eCommerce synchronizes all customer interactions and transactions by integrating web-based channels with traditional channels.

Enterprise Resource Planning (ERP)

Companies use the ERP applications to control their back-office operations. For example:

- Oracle Order Management applications feature advanced configurator functionality, global available to promise, flexible pricing support, efficient delivery, high volume transactions and flexibility to adapt to changing business conditions.
- Oracle Supply Chain Planning applications provide the tools required to optimize flow of material, cash, and information across the extended supply chain.
- Oracle Manufacturing applications support all styles of manufacturing - engineer-to-order, discrete, process, flow, lot based, and project based manufacturing.
- Oracle Financials provide solutions for strategic planning, accounting, treasury, project management, and travel management.

- Oracle Human Resources Management System is a comprehensive solution for managing a company's human resources, allowing organizations to attract, retain and develop critical skills and knowledge on a global basis.

Common Application Architecture

The Common Application Architecture includes functionality that supports both CRM and ERP applications. For example, TCA, Oracle's Trading Community Architecture, consists of a database schema and Application Programming Interfaces (APIs) where you can model the complex relationships that occur within a business community and enter that data consistently throughout the enterprise. Because the model is not hierarchical, Oracle applications can model complex B2B2C relationships and not to be limited to either a B2B or B2C implementation. TCA delivers a 360-degree view of the customer.

1.1.2 Oracle Marketing Family Overview

The Oracle Marketing family provides complete campaign management to help shorten time to market. To become more cost-effective and more responsive to changing market conditions and customer requirements, a marketing organization must be able to manage its planning, execution, and reporting in fully centralized and automated fashion.

The Oracle Marketing family of applications includes:

- Oracle Marketing
- Oracle Advanced Marketing Online
- Oracle Marketing Intelligence

1.2 Oracle Leads Management Overview

Oracle Leads Management comprises solutions to automate and optimize prospect-to-sales conversion across the enterprise. Leads Management provides a staging area for all prospect leads for data quality processing, prioritization and distribution, enterprise review, and conversion.

1.2.1 Oracle Leads Management Features

The Leads Management features include:

- Lead capture

- Customer and lead data quality
- Real time flexible rules engine for lead evaluation and distribution
- Integrated monitoring
- Multiple channel lead handling and maturation
- Lead utilization and effectiveness analysis

1.3 New in this Release

Note: This document describes functionality to be delivered in the Oracle E-Business Suite 11.5.9 release. If you are implementing this product prior to the release, using product minipacks or family packs, some new functionality may be dependent on integration with other Oracle products. Please consult MetaLink for relevant product patches and documentation.

This release includes the following new features:

- Section 1.3.1, "Lead Capture Management"
- Section 1.3.2, "Data Quality"
- Section 1.3.3, "Leads Management Processing and Distribution"
- Section 1.3.4, "Leads Management Monitoring Engine"
- Section 1.3.5, "Leads Management Navigation"
- Section 1.3.6, "Leads Management Intelligence"
- Section 1.3.7, "Leads Management Smart Time Frames"

1.3.1 Lead Capture Management

Oracle Leads Management leverages the Oracle Marketing List management tools, and Scripting to capture each response across online and offline channels.

Key features include:

- Response Management
- Lead Imports

1.3.2 Data Quality

Oracle Leads Management leverages E-Business Suite data quality engines to ensure quality lead data for follow up and tracking.

Key features include:

- Leads Management integrates with Oracle Data Quality Management (DQM) Trading Community Architecture, and leverages this model for fuzzy matching logic for existence checking for merge, match, and de-duplication functionality for customer (organization), contact (person), address, and contact point customer data.
- Lead de-duplication is also integrated with DQM.

1.3.3 Leads Management Processing and Distribution

The Oracle Leads Management Rules Engine is a real time, flexible rules engine which evaluates the quality and priority of leads, and then routes leads based on business workflow and best practices.

The Leads Rules Engine groups rules into domains in order to encapsulate business logic across organizational and regional boundaries; for example, by promotion, product, country, or industry. This way, rules can be managed both centrally and across organizations to support a range of best practices.

Key features include:

- Domain-Specific Business Logic
- Rules Flexibility
- Rule Engine Deployment Support

1.3.4 Leads Management Monitoring Engine

Oracle Leads Management introduces a Monitoring framework, which enables the user to set up active monitors to track the state of each lead. Monitors can be set to be triggered on a set of conditions, and can provide notification to one or more users on activation.

Monitors can be associated with lead attributes - Lead Country and Lead Rank.

Key features include:

- No follow ups within a specified period
- Set a threshold for the number of lead re-routes

- Reroute to a new resource

1.3.5 Leads Management Navigation

The Oracle Leads Management user interface provides a navigation tool to group and manage leads, to enable sales representatives to identify leads for action. The user interface supports flexible views across leads and contact attributes, and breaks each lead down into details snapshot, Contact information, and History.

Key features include:

- Lead State
- Leads Summary
- Usability

1.3.6 Leads Management Intelligence

Oracle Leads Management Daily Business Intelligence tracks and reports lead activity based on quantity of leads generated and also on lead quality. Leads and their actual value, revenue, and cost are correlated to marketing segments, campaigns, offers, and products to enable businesses to optimize marketing programs and segmentation to fully support the sales cycle.

In addition, Leads Management Intelligence provides full visibility into the channel and sales handling of leads by tracking lead activity, aging, and closure by each channel, group, and representative.

1.3.7 Leads Management Smart Time Frames

Smart time frames determine the expiration date of a lead. Expiration date assumes the maximum length of the time frame relative to the creation date. Previously, time frames were simply lookup codes. Now, the time frame are in terms of number of days.

Seeded examples:

- Within 1 week : 7 days
- 1-3 months : 90 days

1.4 Obsolete in this Release

The Scorecard and Lead Ranking Mapping function is obsolete from this release. If you are upgrading from 11.5.8, you can view the read-only versions of the scorecard and lead rank mapping data. For more details on viewing this data, see Section 2.3.9, "Migrating Lead Engines from 11.5.8"

1.5 Related Documentation

Following are additional documents relating to the modules discussed in this guide or referred to in implementation tasks:

- *Oracle Sales Online User Guide*
- *Oracle TeleSales User Guide*
- *Oracle CRM Application Foundation Implementation Guide*
- *Oracle CRM Application Foundation User Guide*
- *Oracle General Ledger User Guide*
- *Oracle Inventory User's Guide*
- *Oracle Marketing User Guide*
- *Oracle Receivables User Guide*
- *Oracle Partners Implementation Guide*
- *Oracle Applications System Administrator's Guide*
- *Oracle Applications Flexfields Guide*
- *Using Oracle HRMS - The Fundamentals*
- *Managing Your Workforce Using Oracle HRMS, Release 11i*
- *Multiple Organizations in Oracle Applications*
- *Installing Oracle Applications*
- *Oracle Universal Work Queue Implementation Guide*
- *Oracle Scripting Implementation Guide*
- *Oracle Scripting User Guide*
- *Oracle 9i Application Server Documentation, Release 1*
- *Oracle9i Database Utilities*

1.6 Oracle Leads Management Dependencies

Oracle Leads Management is dependent upon Oracle Sales Online and Oracle TeleSales for its proper functioning. Ensure that both these applications are installed and implemented correctly. For more information, please see the *Oracle Sales Online Implementation Guide* and the *Oracle TeleSales Implementation Guide*.

Part II

Administering Oracle Leads Management

This section contains the following chapter:

2

Administration Tasks for Oracle Leads Management

The administrative tasks for administering Oracle Leads Management are categorized under the following headings:

- Section 2.1, "Lead Capture"
- Section 2.2, "Data Quality"
- Section 2.3, "Processing and Distribution of Leads"
- Section 2.4, "Tracking"

2.1 Lead Capture

Capturing leads information includes the following tasks:

- Section 2.1.1, "Importing Leads"
- See topics covered in Appendix D, "Oracle Leads Management API Reference"

2.1.1 Importing Leads

The Import Sales Lead concurrent program allows you to import leads from other systems. While importing leads, the program also imports customers, addresses, and contacts information into the Trading Community Architecture (TCA) tables, if this data does not exist already.

Sections in this topic include:

- Section 2.1.1.1, "Before Running the Import Sales Lead Concurrent Program"
- Section 2.1.1.2, "Loading the AS_IMPORT_INTERFACE table"

- Section 2.1.1.3, "Setting Profiles Used by Import Sales Lead Concurrent Program"

2.1.1.1 Before Running the Import Sales Lead Concurrent Program

The following steps must be complete before running the Import Sales Lead concurrent program:

1. Load the following import interface tables-
   - AS_IMPORT_INTERFACE (mandatory)

     See Section 2.1.1.2, "Loading the AS_IMPORT_INTERFACE table".
   - AS_IMP_CNT_ROL_INTERFACE (optional)
   - AS_IMP_CNT_PNT_INTERFACE (optional)
   - AS_IMP_LINES_INTERFACE (optional)
   - AS_IMP_SL_FLEX (optional)

   For more details, see Section 2.1.2.1, "Understanding the Lead Import Interface Tables".
2. Create valid lookup codes for lookup type SOURCE_SYSTEM.
3. Set up the following profile options:
   - OS: Default Resource ID used for Sales Lead Assignment
   - OS: Use DQM Rule code to match Party
   - OS: Use DQM Rule code to match Person
   - OS: Use DQM Rule code to match Contact

   See Appendix A, "Oracle Leads Management Profile Options" for other profile options required to create leads.
4. Set up Territory - This involves two steps:
   a. Create territory inside Oracle Sales and Telesales node
      - Responsibility: Oracle Sales Administrator
      - Navigation: CRM Foundation > Territory Manager > Territory Administration
   b. Run concurrent program to create territory mappings
      - Responsibility: Oracle Sales Administrator

- Concurrent Program Name: Generate Territory Packages & Territory denormalization refresh

5. Set up DQM match rules for Party, Person and Contact.
   - Responsibility: Trading Community Manager
   - Navigation: Data Quality Management > Setup > Match Rules
6. Prepare Oracle Sales for any new data that you want to import.

2.1.1.2 Loading the AS_IMPORT_INTERFACE table

You can load the AS_IMPORT_INTERFACE table using two methods:

- Running a Concurrent Program
- Section 2.1.6, "Importing Leads from the User Interface"

Running a Concurrent Program

- Concurrent Program Name: Load Sales Lead Interface Table from Flat File
- Responsibility: Oracle Sales Administration
- Parameters: Absolute path and file name of the data file

The concurrent program supports only loading the AS_IMPORT_INTERFACE table. To take advantage of other interface tables, you must write your own program to populate them.

For more details, see Section 2.1.5, "Importing Leads from a File or by Populating the Lead Interface Table".

Note: The data file used to load the AS_IMPORT_INTERFACE table should be tilde(~) delimited and the file name must end with a .dat (for example, `mynewleads.dat`). Make sure the read and write permissions are granted to all directories including the file.

2.1.1.3 Setting Profiles Used by Import Sales Lead Concurrent Program

The Import Sales Lead concurrent program uses three new profiles to select the rules to run. The profiles must be assigned valid values before running the Import Sales Lead concurrent program. The profiles are:

- OS: Use DQM Rule code to match Party
- OS: Use DQM Rule code to match Person

- OS: Use DQM Rule code to match Contact

Prerequisites

You must set up leads, including the required lead system profile options as described in Appendix A, "Oracle Leads Management Profile Options".

Steps

1. Set up system profile options for lead import.
   - OS: Use DQM Rule code to match Party

     This profile option must be associated with rules to find matching records based on the Party ID in the imported record.
   - OS: Use DQM Rule code to match Contact

     This profile option must be associated with rules to find matching records based on the Contact in the imported record.
   - OS: Use DQM Rule code to match Person

     This profile option must be associated with rules to find matching records based on the Person in the imported record.
   - OS: Default Resource ID Used for Sales Lead Assignment

     Set this system profile option to the resource who is to handle any leads that are not assigned to any current territory.
   - OS: Auto Assigning from Lead Import

     Default setting: Y

     Note: You must set this profile to a resource that has a valid sales role assigned in Resource Manager. If you do not assign a valid resource, then the ownership of unassigned leads is assigned to the user importing or updating the leads. If that user is also not set up in Resource Manager, then the leads you import will not be accessible from either Oracle Sales Online or Oracle TeleSales.
   - OS: Auto Ranking from Lead Import

     Default setting: Y

2. Create your territories inside the Oracle Sales and TeleSales node on the territory setup form.
   - Responsibility: Oracle Sales Administrator
   - Navigation: CRM Foundation > Territory Manager > Territory Administration
3. See *Oracle CRM Application Foundation Implementation Guide,* Implementing Territory Manager.

   Successfully run the Generate Territory Packages concurrent program. This concurrent program builds the API that returns the winning territories which are defined in territory setup. It must be run at least once before you import leads and every time after the territory setup is modified. You need not run this program every time you import leads.

2.1.2 Running the Import Sales Lead Concurrent Program

To import leads, you must run the Import Sales Lead concurrent program.

Prerequisites

See the following sections:

- Section 2.1.1.1, "Before Running the Import Sales Lead Concurrent Program"
- Section 2.1.1.2, "Loading the AS_IMPORT_INTERFACE table"
- Section 2.1.1.3, "Setting Profiles Used by Import Sales Lead Concurrent Program"

Responsibility

Oracle Sales Administrator

Navigation

View > Requests > Submit New Request

Steps

Run the Import Sales Lead concurrent program.

Concurrent Program Parameters

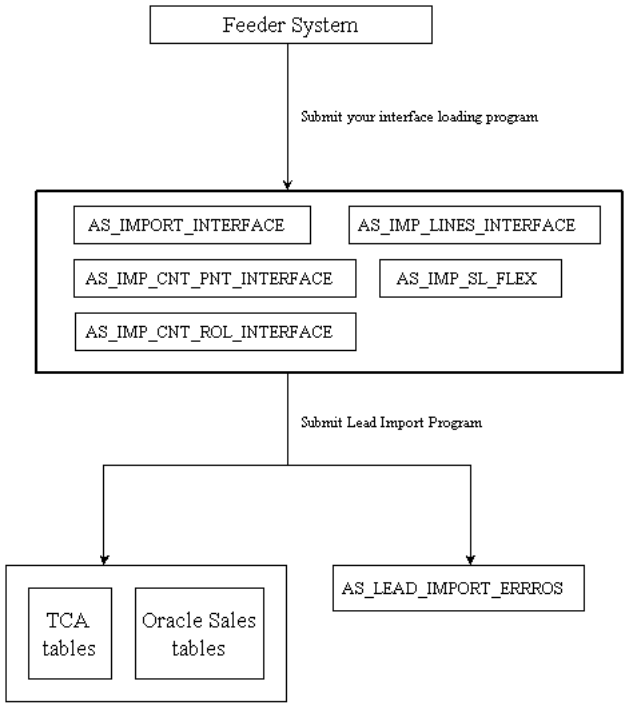
Table 2–1 lists the parameters for the Import Sales Lead concurrent program.

Table 2–1 Import Sales Lead concurrent program parameters

Parameter	Req?	Lookup	Remarks
Source System	Y	SOURCE_ SYSTEM	Used to identify leads generated from different business entities. Only the records that match the parameter value are selected for processing. This is case-sensitive.
Debug message?	N	Y or N	Default is N. If set to Y, the debug messages can be seen by clicking **View Log** in the Concurrent Request screen.
BatchID	N	-	Used to process a small set of data. This is particularly useful when leads are imported in batches. The Batch ID may be used to process only a particular set of data in a batch.
Purge error message?	N	Y or N	Default is N. If set to Y, all records in the AS_LEAD_IMPORT_ ERRORS table are deleted.

Figure 2–1 shows how lead information is imported into Oracle Sales tables.

Figure 2–1 Leads Data Imported into Oracle Sales Tables



2.1.2.1 Understanding the Lead Import Interface Tables

The following are the lead import interface tables:

- AS_IMPORT_INTERFACE - This interface table holds sales leads, customers, addresses, and contacts information to be imported. This table also holds space to import five lead lines in one record.
- AS_IMP_LINES_INTERFACE - This interface table can be used to hold imported lead lines information, in case you have more than five line items for a lead.
- AS_IMP_CNT_ROL_INTERFACE - This interface table is used to hold contact roles information to be imported.

- AS_IMP_CNT_PNT_INTERFACE - This interface table is to hold any extra contact points information to be imported apart from the AS_IMPORT_INTERFACE table.
- AS_IMP_SL_FLEX - This interface table is to store the flexfields values.

Note: If you populate purchase items columns (INTEREST_TYPE_ID_1 to OFFER_ID_5) in the AS_IMPORT_INTERFACE table as well as in the AS_IMP_LINES_INTERFACE table, the data in both the tables will be imported. For more information on the columns in these tables, see Section 2.1.7.1, "Table Structure for Lead Import".

2.1.2.2 About Flexfields

Use the AS_IMP_SL_FLEX table to store the flexfields values for all the following entities (tables). The entity names are seeded in AS_LOOKUPS, lookup_type = ENTITY_NAME.

- HZ_PARTIES
- HZ_LOCATIONS
- HZ_CONTACT_POINTS
- HZ_PARTY_SITES
- HZ_ORG_CONTACTS
- AS_SALES_LEADS
- AS_SALES_LEAD_LINES
- AS_SALES_LEAD_CONTACTS

The flexfields are imported along with the other data in the AS_IMPORT_INTERFACE table during the lead import process. To populate the data in the optional tables, use SQL*Loader or SQLPLUS.

The flexfields columns in HZ_ORG_CONTACT_ROLES, the global flexfields columns in HZ_PARTIES, HZ_LOCATIONS, HZ_CONTACT_POINTS, and HZ_ORG_CONTACTS are obsoleted. Hence, the Import Sales Lead concurrent program does not support these columns.

Reference

Refer the *Oracle Applications Flexfields Guide* for information about how to plan and set up flexfields.

2.1.2.3 Checking for and Correcting Errors in Imported Leads

Use this procedure to check for and correct any errors detected by the Import Sales Lead concurrent program during lead import. The errors are stored in the AS_LEAD_IMPORT_ERRORS table.

The status and descriptions for the lead import errors is given below.

Status	Description
Complete	All records complete/successful.
Incomplete - Errors Found	One or more records complete/successful with one or more errors.
Incomplete - Duplicates Found	One or more records complete/successful with one or more duplicates.
Incomplete - Duplicates and Errors Found	One or more records complete/successful with one or more duplicates AND one or more errors found.
Error	All records error.
Duplicate	All records duplicates.

Note: Do not attempt to change the status of a lead from `Success` to `New`. A status of `Success` means that a lead has been successfully imported and cannot be imported again. This illegal operation is not permitted from the user interface.

Prerequisites

- Populate the interface table.
- Run the Import Sales Lead concurrent program.

Login

Log in to Oracle HTML Applications

Responsibility

Oracle Sales Online Super User

Navigation

Administration > Leads > Import Records Manager

Steps

1. The Lead Import page appears displaying details of imported leads with different statuses.
2. Find the records:
   a. Specify the Source System
   b. To view records that resulted in an error, specify Load Status = `ERROR`.
   c. Optionally, you can specify the Import Interface ID, Batch ID, Source Code, Load Type, and Load Date to narrow down your search.
3. Click **Search** to view records that resulted in an error.
4. Click the Import Interface ID hyperlink to view details.
5. Click the Organization hyperlink to modify customer, address, contact and contact point attributes.
6. Click the Leads hyperlink to modify leads attributes.
7. Click the Errors hyperlink to view the exact error that occurred during lead import.
8. Change the load_status for the record to `NEW`.

   Note: You must change the load status of a corrected record to `New` for the Import Sales Lead concurrent program to load the data again.
9. Click **Update** after modifying the attributes.
10. Repeat steps 4 to 9 for each record with errors.
11. Once all of the records with errors are modified, run the Import Sales Leads concurrent program to import the leads again.

Note: To remove the records of past errors, enter `Yes` to the parameter **Purge error messages?** when you submit the Import Sales Lead concurrent program. This truncates all records in the AS_LEAD_IMPORT_ERRORS table.

2.1.3 Import Sales Lead Concurrent Program Flow

The Import Sales Lead concurrent program imports records in the following manner and sequence:

1. Checks for the existence of Original System Reference (OSR) using Leads Data Quality. See Section 2.2.1.1, "Checking for Original System Reference Duplication".
2. Checks for the existence of customer, address, contact, and contact points using DQM Data Quality. See Section 2.2.4, "DQM Duplication Check".
3. Checks for duplicate leads using Leads Data Quality. See Section 2.2.2, "Lead Import Deduplication".
4. Creates TCA data, if required.
5. Creates a lead, if not duplicate.
6. Qualifies and ranks the lead.
7. Assigns the lead and populates the lead sales team.
8. Creates customer sales team.

2.1.4 Limitation of the Import Sales Lead Concurrent Program

The Import Sales Lead concurrent program checks the database for duplicates using the DQM logic of customer, address, contact, contact point, and restriction information before creating new records. However, the DQM logic has one major limitation.

While importing leads, if the import program creates new records such as Party, Contact, Party Site and Contact Points, the new entries are not reflected in the DQM staging schema. For more information on the DQM staging schema, see the *Oracle Trading Community Architecture Data Quality Management User Guide*. Therefore, if the same set of leads is imported again without any changes, the DQM logic will fail causing the Lead deduplication program to fail as well. To overcome this, the Synchronization concurrent program must be run after the first import and before

the next import. To run this concurrent program, see Running the Synchronization Concurrent Program.

However, the lead deduplication program can still fail if there are duplicate leads in a single set of imported records.

2.1.4.1 Running the Synchronization Concurrent Program

The Synchronization concurrent program must be run after every lead import activity. This updates the DQM staging schema with new entries that were created during the lead import.

Prerequisites

None

Responsibility

Trading Community Manager

Login

Log in to Oracle Forms

Navigation

Control > Request > Run

Steps

Select **DQM Synchronization Program**.

Allow the concurrent program to run and complete the synchronization. This copies data from the HZ base table to the staging table.

2.1.5 Importing Leads from a File or by Populating the Lead Interface Table

Use this procedure to import leads into the database from a flat file or by entering data directly into a lead interface table. The lead import assigns the leads automatically to the appropriate agent or salesperson via the Territory Manager module of the Oracle E-Business Suite.

Prerequisites

- You must be familiar with running concurrent programs as described in the *Oracle Applications Systems Administrator's Guide*.

- You must be familiar with Oracle SQL Loader as described in the *Oracle9i Database Utilities* guide.
- You must be familiar with running SQL Plus database queries.

Steps

1. To import leads from a file-
   a. FTP the tilde-delimited file with lead data to a directory on the server. The file must have a file name with the extension `.dat`. For more details, see
      – Section 2.1.5.1, "Importing from a Flat File"
      – Section 2.1.5.2, "Sample Data File" for a sample file with one data line listed.
      – Section 2.1.7.1, "Table Structure for Lead Import" for the table structure.

      Note: The data in your import file must contain the required fields and the Load Status of each record must be `NEW`.

   b. In the same directory, create an Oracle SQL Loader parameter file. This file, which the Import Sales Lead concurrent program refers to as the P_DATAFILE, can have any name. Here is what a sample P_DATAFILE file will look like:

      ```
      userid=<username>/<password>
      control=ASTSLIMP.CTL
      data=<path><lead import data file name>.dat
      ```

      See the *Oracle 9iDatabase Utilities* guide for more information.
   c. Log in with the Oracle TeleSales Administration responsibility.
   d. Run the Oracle Sales Admin: Load Sales Lead Interface Table from Flat File concurrent program. This program populates the lead interface table (AS_IMPORT_INTERFACE). To run the program you must know the absolute path to the P_DATAFILE you created in the previous step.
2. Populate any flexfields data into the table AS_IMP_SL_FLEX. See Section 2.1.2.2, "About Flexfields".
3. After loading leads into the lead interface table, run the Import Sales Lead concurrent program. See Section 2.1.2, "Running the Import Sales Lead Concurrent Program" for more details.

4. After the Import Sales Lead concurrent program has run successfully, correct any errors. For more information, see Section 2.1.2.3, "Checking for and Correcting Errors in Imported Leads".

> **Note:** The Import Sales Lead concurrent program may run without any errors, but you may still have corrupt or missing data in the imported leads. Check for errors to ensure that leads have been successfully imported. The program writes errors into the AS_LEAD_IMPORT_ERRORS table.

2.1.5.1 Importing from a Flat File

The flat file you are importing leads from must be a tilde-delimited file. The file name must have the extension `.dat` (for example, `mynewleads.dat`). Please note:

- As the leads you import are not deleted from the AS_IMPORT_INTERFACE interface table even after they are successfully imported into the database, you should enter different batch IDs in each import file to make sure you can distinguish between runs.
- A `Yes` entry in a Required column means that a null or incorrect entry in the field results in a database error. Table 2–2 and Table 2–3 list the columns present in the AS_IMPORT_INTERFACE table.

  You must provide valid values to these columns. Run SQL*Plus queries to obtain some of the values required by this table. The columns you must query and sample queries for your use are listed in the Valid Values column.
- Lead import records can contain null fields except for the required fields.
- If you are importing leads and have flexfield set up in your application, you must also populate the AS_IMP_SL_FLEX table.

2.1.5.2 Sample Data File

Below is a sample data file for loading the AS_IMPORT_INTERFACE. This example contains only one line of data.

```
~16-Sep-01~-1~16-Sep-01~-1~-1~LEAD_LOAD~16-Sep-01~NEW~ABC Corp~US~123
Xyzst.~Suite1008~~~RedwoodCity~94065~CA~~~7374~2000~CUSTOMER~MAR~15000~5000000~6
000000~DECISIONMAKER~1023472~1900~N~M~MR~~Hislast~Hisfirst~A~AccountsPayableSupe
rvisor~ARC~101~926-2667~GEN~650~123~926-2600~650~www.xyz.com~abc@xyz.com~Y~Y~N~N
~NEW~DECISION_
MAKER~DIRECT~10000~APPROVED~US~1-3MONTHS~~AAA~Lead1~EMAIL~159~424~425~357~204~EA
~100~50000~10588~10699~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Leadcollectedon16
```

```
-SEP-01~~NEW~10001~OTN::990~10004~Y~Y~~~~~~N~Y~Y~~~~10060~Y~1-Jan-00~~~USERENTER
ED~XYZHQ~~~10~~~~~~94065-1282~~~HQ~~~~1008~~~~~~~~XYZ~~~~~~15-Aug-01~~~~~~~~~~N~
1987SIC~~101~550000~~~N~MARKET~~10~~~Importantcontact~~Y~InformationTechnology~I
T~DECISION_MAKER~10588~N~Y~USER_ENTERED~1~1~~~~MAILHTML~123~~~~~~~~~~~TSTENH
```

2.1.6 Importing Leads from the User Interface

Prerequisites

None

Login

Log in to Oracle HTML Applications

Responsibility

Sales Online Super User

Navigation

HTML - Audience > Import

Steps

1. In the Imports page, click **Create**.

   The Import: Introduction page appears.

2. From the Data Type column, select Leads.

3. Click **Next**.

   The Import Step 1a: Definition page appears.

4. In the General region, enter the name and description of the import.

5. In the Source File section:

   a. If the source file is at a client location, click **Go** next to the Client field to select the name and location of a source file from the local hard disk or network.

   b. If the source file is at a server location, in the Server field, enter the URL for the source file.

   c. If the source file is at a FTP location, click **Go** next to the FTP field, and enter the full path for the source file.

The file types supported are -

- a zip file containing an XML and a DTD file
- a zip file containing a CSV file
- a CSV file

d. Use the Character Set drop-down list to select the character set that the source file uses.

e. Use the Column Delimiter drop-down list to select the delimiter used in the file to distinguish between two columns.

Choose tilde (~) unless you are using SQL Loader.

f. Use the Field Enclosed By drop-down list to select the character that encloses each field in the file.

g. Select the File Header Exists check box, if the columns in the source file have a header.

6. Click **Next**.

The Import Mapping page appears. The Source Fields are the columns in your file. The Target Fields are the columns present in the table.

7. Select a Source field and a corresponding Target field.

8. Click **>.**

The mapped fields appear in the Mapped Source Target fields section.

9. Repeat steps 7 and 8 for all the Source fields.

Ensure that all mandatory fields are mapped. Mandatory fields have an asterisk in the Target Fields area.

10. Click **Next**.

The Import Step 3: Review page appears.

11. After reviewing the details, click **Import**.

The Confirmation page appears. Your import data is submitted for processing.

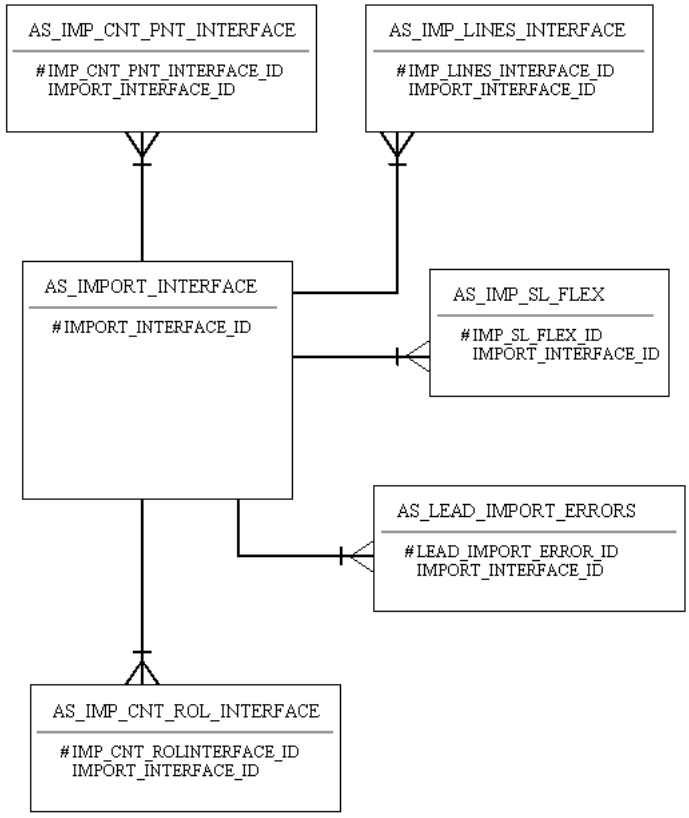
12. Click **Finish**.

13. Optionally, you can track the status of the import from the Imports page.

2.1.7 Database Design and Data Structures

The following figure depicts the database design used during lead import.

Figure 2–2 Database Design



Sections in this topic include:

- Section 2.1.7.1, "Table Structure for Lead Import"
- Section 2.1.7.2, "Table Structures for Multiple Contact Roles"
- Section 2.1.7.3, "Lead Import Errors Table Structure"
- Section 2.1.7.4, "Sequence Generators"

2.1.7.1 Table Structure for Lead Import

Table 2–2 lists the columns that are common to both the AS_IMPORT_INTERFACE table and the control file. The order of these fields is also the same in both. See Additional Columns in the AS_IMPORT_INTERFACE Table for the remaining columns in the AS_IMPORT_INTERFACE table.

A control file is used to give instructions to SQL*Loader. For more information, see the *Oracle9i Database Utilities* guide.

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
IMPORT_ INTERFACE_ID	Y	NUMBER	select as_import_ interface_s.nextval from sys.dual	Sequentially generated by the concurrent program.	-
LAST_UPDATE_ DATE	Y	DATE	-	If the Load Sales Lead interface table is used from the Flat File to load data, the loader populates these columns with sysdate(date) and -1 (user ID).	-
LAST_UPDATED_BY	Y	NUMBER	-	If the Load Sales Lead interface table is used from the Flat File to load data, the loader populates these columns with sysdate(date) and -1 (user ID).	-
CREATION_DATE	Y	DATE	-	If the Load Sales Lead interface table is used from the Flat File to load data, the loader populates these columns with sysdate(date) and -1 (user ID).	-
CREATED_BY	Y	NUMBER	-	If the Load Sales Lead interface table is used from the Flat File to load data, the loader populates these columns with sysdate(date) and -1 (user ID).	-

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
LAST_UPDATE_ LOGIN	Y	NUMBER	-	If the Load Sales Lead interface table is used from the Flat File to load data, the loader populates these columns with sysdate(date) and -1 (user ID).	-
MARKETING_SCORE	N	NUMBER	-	Ranking engine considers this score while ranking a lead.	-
PERSON_INITIALS	N	VARCHAR2 (6)	-	-	HZ_PERSON_ PROFILES.PERSON_ INITIALS
LOAD_TYPE	N	VARCHAR2 (20)	LEAD_LOAD	-	-
LOAD_DATE	N	DATE	-	-	-
LOAD_STATUS	Y	VARCHAR2 (20)	select lookup_code from as_lookups where lookup_type=LOAD_ STATUS	The value should be NEW before importing. The value is changed to SUCCESS if successfully imported, ERROR if not.	-
CUSTOMER_NAME	Y	VARCHAR2 (255)	-	-	HZ_PARTIES.party_ name
COUNTRY	N	VARCHAR2 (60)	select territory_code from fnd_territories_vl	If country is supplied then address1 needs to be supplied and vice-versa.	HZ_ LOCATIONS.country
ADDRESS1	N	VARCHAR2 (240)	-	Country and address1 are required for creating location, party site and party site.	HZ_ LOCATIONS.address1
ADDRESS2	N	VARCHAR2 (240)	-	-	HZ_ LOCATIONS.address2
ADDRESS3	N	VARCHAR2 (240)	-	-	HZ_ LOCATIONS.address3
ADDRESS4	N	VARCHAR2 (240)	-	-	HZ_ LOCATIONS.address4

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
CITY	N	VARCHAR2 (60)	select distinct location_ segment_description, location_segment_user_ value from ar_location_ values where location_ segment_qualifier = CITY	-	HZ_LOCATIONS.city
POSTAL_CODE	N	VARCHAR2 (60)	select min(p.from_ postal_code), max(p.to_ postal_code) from ar_ postal_code_ranges_v p, ar_location_values a where p.location_ segment_id = a.location_segment_id and a.location_ segment_qualifier||'' = CITY and a.location_ segment_value like X%	-	HZ_ LOCATIONS.postal_ code
STATE	N	VARCHAR2 (60)	select distinct location_ segment_description, location_segment_user_ value from ar_location_ values where location_ segment_qualifier = STATE	-	HZ_LOCATIONS.state
PROVINCE	N	VARCHAR2 (60)	select distinct location_ segment_description, location_segment_user_ value from ar_location_ values where location_ segment_qualifier = PROVINCE	-	HZ_ LOCATIONS.province
COUNTY	N	VARCHAR2 (60)	select distinct location_ segment_description, location_segment_user_ value from ar_location_ values where location_ segment_qualifier = COUNTY	-	HZ_ LOCATIONS.county
SIC_CODE	N	VARCHAR2 (30)	select lookup_code from ar_lookups where lookup_type = 1987 SIC and enabled_flag = Y	If given, then SIC_ CODE_TYPE must be provided too.	HZ_PARTIES.sic_code

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
ANALYSIS_FY	N	VARCHAR2 (5)	-	Analysis physical year	HZ_PARTIES.analysis_ fy
CUSTOMER_ CATEGORY_CODE	N	VARCHAR2 (30)	select lookup_code from ar_lookups where lookup_type = CUSTOMER_ CATEGORY and enabled_flag=Y	-	HZ_PARTIES.category_ code
FISCAL_YEAREND_ MONTH	N	VARCHAR2 (30)	-	free text format	HZ_PARTIES.fiscal_ yearend_month
NUM_OF_ EMPLOYEES	N	NUMBER(15)	-	-	HZ_ PARTIES.employees_ total
POTENTIAL_ REVENUE_CURR_FY	N	NUMBER	-	-	HZ_PARTIES.curr_fy_ potential_revenue
POTENTIAL_ REVENUE_NEXT_FY	N	NUMBER	-	-	HZ_PARTIES.next_fy_ potential_revenue
CUSTOMER_RANK	N	VARCHAR2 (30)	-	-	HZ_ORG_CONTACT_ ROLES.role_type & AS_ SALES_LEAD_ CONTACTS.rank
TAX_REFERENCE	N	VARCHAR2 (50)	-	-	HZ_PARTIES.tax_ reference
YEAR_ESTABLISHED	N	NUMBER(4)	-	-	HZ_PARTIES.year_ established
ADDR_DO_NOT_ MAIL_FLAG	N	VARCHAR2 (1)	-	If set to Y, no e-mails are sent to customer site	-
SEX_CODE	N	VARCHAR2 (30)	-	-	-
SALUTATION	N	VARCHAR2 (60)	ar_lookups.lookup_code where lookup_type = CONTACT_TITLE and enabled_flag = Y	This column is for terms used while greeting the contact. For example, Her Highness.	HZ_PARTIES.person_ academic_title & HZ_ PARTIES.person_title
TITLE	N	VARCHAR2 (30)	-	-	HZ_PARTIES.person_ pre_name_adjunct
LAST_NAME	N	VARCHAR2 (50)	-	Last name and first name are required for creating a contact.	HZ_PARTIES.person_ last_name

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
FIRST_NAME	N	VARCHAR2 (40)	-	Last name and first name are required for creating a contact.	HZ_PARTIES.person_ first_name
MIDDLE_INITIAL	N	VARCHAR2 (10)	-	-	HZ_PARTIES.person_ middle_name
JOB_TITLE	N	VARCHAR2 (50)	-	-	HZ_ORG_ CONTACTS.job_title
JOB_TITLE_CODE	N	VARCHAR2 (30)	ar_lookups.lookup_code where lookup_type =RESPONSIBILITY and enabled_flag = Y	Stores the job title code	HZ_ORG_ CONTACTS.job_title_ code
MAIL_STOP	N	VARCHAR2 (60)	-	-	HZ_ORG_ CONTACTS.mail_stop
PHONE_NUMBER	N	VARCHAR2 (25)	-	-	HZ_CONTACT_ POINTS.phone_ number
PHONE_TYPE	N	VARCHAR2 (30)	select lookup_code from ar_lookups where lookup_type = PHONE_ LINE_TYPE and enabled_flag = Y	-	HZ_CONTACT_ POINTS.phone_line_ type
AREA_CODE	N	VARCHAR2 (10)	-	-	HZ_CONTACT_ POINTS.phone_area_ code
EXTENSION	N	VARCHAR2 (20)	-	-	HZ_CONTACT_ POINTS.phone_ extension
FAX_NUMBER	N	VARCHAR2 (25)	-	-	HZ_CONTACT_ POINTS.phone_ number
FAX_AREA_CODE	N	VARCHAR2 (10)	-	-	HZ_CONTACT_ POINTS.phone_area_ code
URL	N	VARCHAR2 (2000)	-	-	HZ_CONTACT_ POINTS.url
EMAIL_ADDRESS	N	VARCHAR2 (240)	-	-	HZ_CONTACT_ POINTS.email_address
CONT_DO_NOT_ MAIL_FLAG	N	VARCHAR2 (1)	Y or N	If set to Y, then no e-mails are sent to the contact.	HZ_CONTACT_ RESTRICTIONS

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
DO_NOT_EMAIL_ FLAG	N	VARCHAR2 (1)	Y or N	If set to Y, then no e-mails sent to contact.	HZ_CONTACT_ RESTRICTIONS
DO_NOT_FAX_FLAG	N	VARCHAR2 (1)	Y or N	If set to Y, then no fax sent to contact.	HZ_CONTACT_ RESTRICTIONS
DO_NOT_PHONE_ FLAG	N	VARCHAR2 (1)	Y or N	If set to Y, then no phone calls made to contact.	HZ_CONTACT_ RESTRICTIONS
STATUS_CODE	N	VARCHAR2 (30)	select status_code from as_statuses_b where lead_flag = Y and enabled_flag = Y and usage_indicator in (ALL,OS)	-	AS_SALES_ LEADS.status_code
CONTACT_ROLE_ CODE	N	VARCHAR2 (30)	select lookup_code from as_lookups where lookup_type = LEAD_ CONTACT_ROLE	-	AS_SALES_ LEADS.contact_role_ code & AS_SALES_ LEAD_ CONTACTS.contact_ role_code
CHANNEL_CODE	N	VARCHAR2 (30)	select sales_channel_ code from aso_i_sales_ channels_v	-	AS_SALES_ LEADS.channel_code
BUDGET_AMOUNT	N	NUMBER	-	-	AS_SALES_ LEADS.budget_amount
BUDGET_STATUS_ CODE	N	VARCHAR2 (30)	select lookup_code from as_lookups where lookup_type = BUDGET_STATUS	-	AS_SALES_ LEADS.budget_status_ code
CURRENCY_CODE	N	VARCHAR2 (15)	select currency_code from fnd_currencies where enabled_flag = Y	-	AS_SALES_ LEADS.currency_code
DECISION_ TIMEFRAME_CODE	N	VARCHAR2 (30)	select lookup_code from as_lookups where lookup_type = DECISION_ TIMEFRAME	-	AS_SALES_ LEADS.decision_ timeframe_code
CLOSE_REASON	N	VARCHAR2 (30)	select lookup_code from as_lookups where lookup_type = CLOSE_ REASON	-	AS_SALES_ LEADS.close_reason

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
PARENT_PROJECT	N	VARCHAR2 (80)	-	-	AS_SALES_ LEADS.parent_project
DESCRIPTION	N	VARCHAR2 (2000)	-	-	AS_SALES_ LEADS.description
VEHICLE_ RESPONSE_CODE	N	VARCHAR2 (30)	select lookup_code from as_lookups where lookup_type = VEHICLE_RESPONSE_ CODE and enabled_flag = Y	-	AS_SALES_ LEADS.vehicle_ response_code
INTEREST_TYPE_ID_ 1	N	NUMBER	select interest_type_id from as_interest_types_ b where ENABLED_ FLAG = Y and EXPECTED_ PURCHASE_FLAG = Y	-	AS_SALES_LEAD_ LINES.interest_type_id
PRIMARY_ INTEREST_CODE_ ID_1	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = and Pic.Parent_Interest_ Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.primary_ interest_code_id
SECONDARY_ INTEREST_CODE_ ID_1	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = And Parent_Interest_Code_ Id = ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.secondary_ interest_code_id
INVENTORY_ITEM_ ID_1	N	NUMBER	select inventory_item_ id, organization_id from mtl_system_items	-	AS_SALES_LEAD_ LINES.inventory_item_ id
ORGANIZATION_ ID_1	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.organization_id
UOM_CODE_1	N	VARCHAR2 (3)	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_ LINES.uom_code
QUANTITY_1	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.quantity
BUDGET_AMOUNT_ 1	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.budget_amount

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
SOURCE_ PROMOTION_ID_1	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	-	AS_SALES_LEAD_ LINES.source_ promotion_id
OFFER_ID_1	N	NUMBER	select source_code_id from ams_source_codes where arc_source_code_ for = OFFR	-	AS_SALES_LEAD_ LINES.offer_id
INTEREST_TYPE_ID_ 2	N	NUMBER	select interest_type_id from as_interest_types_ b where ENABLED_ FLAG = Y and EXPECTED_ PURCHASE_FLAG = Y	-	AS_SALES_LEAD_ LINES.interest_type_id
PRIMARY_ INTEREST_CODE_ ID_2	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = and Pic.Parent_Interest_ Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.primary_ interest_code_id
SECONDARY_ INTEREST_CODE_ ID_2	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = And Parent_Interest_Code_ Id = ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.secondary_ interest_code_id
INVENTORY_ITEM_ ID_2	N	NUMBER	select inventory_item_ id, organization_id from mtl_system_items	-	AS_SALES_LEAD_ LINES.inventory_item_ id
ORGANIZATION_ ID_2	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.organization_id
UOM_CODE_2	N	VARCHAR2 (3)	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_ LINES.uom_code
QUANTITY_2	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.quantity
BUDGET_AMOUNT_ 2	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.budget_amount
SOURCE_ PROMOTION_ID_2	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	-	AS_SALES_LEAD_ LINES.source_ promotion_id

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
OFFER_ID_2	N	NUMBER	select source_code_id from ams_source_codes where arc_source_code_ for = OFFR	-	AS_SALES_LEAD_ LINES.offer_id
INTEREST_TYPE_ID_ 3	N	NUMBER	select interest_type_id from as_interest_types_ b where ENABLED_ FLAG = Y and EXPECTED_ PURCHASE_FLAG = Y	-	AS_SALES_LEAD_ LINES.interest_type_id
PRIMARY_ INTEREST_CODE_ ID_3	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = and Pic.Parent_Interest_ Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.primary_ interest_code_id
SECONDARY_ INTEREST_CODE_ ID_3	N	NUMBER	select Interest_Code_Id from As_Interest_ Codes_B where Interest_Type_Id = And Parent_Interest_Code_ Id = ENABLED_FLAG = Y	-	AS_SALES_LEAD_ LINES.secondary_ interest_code_id
INVENTORY_ITEM_ ID_3	N	NUMBER	select inventory_item_ id, organization_id from mtl_system_items	-	AS_SALES_LEAD_ LINES.inventory_item_ id
ORGANIZATION_ ID_3	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.organization_id
UOM_CODE_3	N	VARCHAR2 (3)	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_ LINES.uom_code
QUANTITY_3	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.quantity
BUDGET_AMOUNT_ 3	N	NUMBER	-	-	AS_SALES_LEAD_ LINES.budget_amount
SOURCE_ PROMOTION_ID_3	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	-	AS_SALES_LEAD_ LINES.source_ promotion_id
OFFER_ID_3	N	NUMBER	select source_code_id from ams_source_codes where arc_source_code_ for = OFFR	-	AS_SALES_LEAD_ LINES.offer_id

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
INTEREST_TYPE_ID_4	N	NUMBER	select interest_type_id from as_interest_types_b where ENABLED_FLAG = Y and EXPECTED_PURCHASE_FLAG = Y	-	AS_SALES_LEAD_LINES.interest_type_id
PRIMARY_INTEREST_CODE_ID_4	N	NUMBER	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = and Pic.Parent_Interest_Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.primary_interest_code_id
SECONDARY_INTEREST_CODE_ID_4	N	NUMBER	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = And Parent_Interest_Code_Id = and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.secondary_interest_code_id
INVENTORY_ITEM_ID_4	N	NUMBER	select inventory_item_id, organization_id from mtl_system_items	-	AS_SALES_LEAD_LINES.inventory_item_id
ORGANIZATION_ID_4	N	NUMBER	-	-	AS_SALES_LEAD_LINES.organization_id
UOM_CODE_4	N	VARCHAR2 (3)	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_LINES.uom_code
QUANTITY_4	N	NUMBER	-	-	AS_SALES_LEAD_LINES.quantity
BUDGET_AMOUNT_4	N	NUMBER	-	-	AS_SALES_LEAD_LINES.budget_amount
SOURCE_PROMOTION_ID_4	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	-	AS_SALES_LEAD_LINES.source_promotion_id
OFFER_ID_4	N	NUMBER	select source_code_id from ams_source_codes where arc_source_code_for = OFFR	-	AS_SALES_LEAD_LINES.offer_id

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
INTEREST_TYPE_ID_5	N	NUMBER	select interest_type_id from as_interest_types_b where ENABLED_FLAG = Y and EXPECTED_PURCHASE_FLAG = Y	-	AS_SALES_LEAD_LINES.interest_type_id
PRIMARY_INTEREST_CODE_ID_5	N	NUMBER	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = and Pic.Parent_Interest_Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.primary_interest_code_id
SECONDARY_INTEREST_CODE_ID_5	N	NUMBER	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = And Parent_Interest_Code_Id = and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.secondary_interest_code_id
INVENTORY_ITEM_ID_5	N	NUMBER	select inventory_item_id, organization_id from mtl_system_items	-	AS_SALES_LEAD_LINES.inventory_item_id
ORGANIZATION_ID_5	N	NUMBER	-	-	AS_SALES_LEAD_LINES.organization_id
UOM_CODE_5	N	VARCHAR2 (3)	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_LINES.uom_code
QUANTITY_5	N	NUMBER	-	-	AS_SALES_LEAD_LINES.quantity
BUDGET_AMOUNT_5	N	NUMBER	-	-	AS_SALES_LEAD_LINES.budget_amount
SOURCE_PROMOTION_ID_5	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	-	AS_SALES_LEAD_LINES.source_promotion_id
OFFER_ID_5	N	NUMBER	select source_code_id from ams_source_codes where arc_source_code_for = OFFR	-	AS_SALES_LEAD_LINES.offer_id
LEAD_NOTE	N	VARCHAR2 (2000)	-	For creating Lead note	-

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
LEAD_RANK_ID	N	NUMBER(15)	as_sales_lead_rank_vl.rank_id, meaning where enabled_flag = Y	Stores the sales lead rank ID. Need not supply if supplied is validated.	AS_SALES_LEADS.lead_rank_id
SOURCE_SYSTEM	Y	VARCHAR2 (250)	select lookup_code from as_lookups where lookup_type = SOURCE_SYSTEM and enabled_flag = Y	Stores the name/tag of the business entity which is populating the leads.	-
BATCH_ID	N	NUMBER(15)	select as_sl_imp_batch_s.nextval from sys.dual	Batch ID of the load; if provided, only the records with the same batch_id are processed.	-
ORIG_SYSTEM_REFERENCE	N	VARCHAR2 (240)	-	May be populated as <orig_system_code >: <identifier> Example: OTN::10100. Please note that this column is used to find customer_key.	-
ORIG_SYSTEM_CODE	N	VARCHAR2 (30)	-	Source application code where the lead originated. Example: OTN	-
URGENT_FLAG	N	VARCHAR2	-	-	AS_SALES_LEADS.URGENT_FLAG
ACCEPT_FLAG	N	VARCHAR2	-	-	AS_SALES_LEADS.accept_flag
ASSIGN_DATE	N	DATE	-	-	AS_SALES_LEADS.assign_date
ASSIGN_SALES_GROUP_ID	N	NUMBER	-	-	AS_SALES_LEADS.assign_sales_group_id
ASSIGN_TO_PERSON_ID	N	NUMBER	-	-	AS_SALES_LEADS.assign_to_person_id
ASSIGN_TO_SALESFORCE_ID	N	NUMBER	-	-	AS_SALES_LEADS.assign_to_salesforce_id

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
AUTO_ ASSIGNMENT_TYPE	N	VARCHAR2	-	-	AS_SALES_ LEADS.auto_ assignment_type
DELETED_FLAG	N	VARCHAR2	-	-	AS_SALES_ LEADS.DELETED_ FLAG
IMPORT_FLAG	N	VARCHAR2	-	-	AS_SALES_ LEADS.IMPORT_ FLAG
KEEP_FLAG	N	VARCHAR2	-	-	AS_SALES_ LEADS.KEEP_FLAG
PRM_ASSIGNMENT_ TYPE	N	VARCHAR2	-	-	AS_SALES_ LEADS.PRM_ ASSIGNMENT_TYPE
QUALIFIED_FLAG	N	VARCHAR2	-	-	AS_SALES_ LEADS.QUALIFIED_ FLAG
REJECT_REASON_ CODE	N	VARCHAR2	-	-	AS_SALES_ LEADS.REJECT_ REASON_CODE
SCORECARD_ID	N	NUMBER	-	-	AS_SALES_ LEADS.SCORECARD_ ID
PRIMARY_ CONTACT_FLAG	N	VARCHAR2	-	-	AS_SALES_LEAD_ CONTACTS.PRIMARY _CONTACT_FLAG
ADDRESS_ EFFECTIVE_DATE	N	DATE	-	-	HZ_ LOCATIONS.address_ effective_date
ADDRESS_LINES_ PHONETIC	N	VARCHAR2	-	-	HZ_ LOCATIONS.address_ lines_phonetic
ADDRESS_STYLE	N	VARCHAR2	-	-	HZ_ LOCATIONS.address_ style
CONTENT_SOURCE_ TYPE	N	VARCHAR2	select lookup_code from ar_lookups where lookup_type = CONTENT_SOURCE_ TYPE and enabled_flag = Y	This is a mandatory column in HZ_ LOCATIONS, but if not given, default it to USER_ENTERED	HZ_ LOCATIONS.content_ source_type

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
LOC_DESCRIPTION	N	VARCHAR2	-	-	HZ_ LOCATIONS.DESCRIPTION
LOC_HIERARCHY_ ID	N	NUMBER	-	-	HZ_ LOCATIONS.LOC_ HIERARCHY_ID
FA_LOCATION_ID	N	NUMBER	-	-	HZ_LOCATIONS.FA_ LOCATION_ID
FLOOR	N	VARCHAR2	-	-	HZ_ LOCATIONS.FLOOR
HOUSE_NUMBER	N	VARCHAR2	-	-	HZ_ LOCATIONS.HOUSE_ NUMBER
LANGUAGE	N	VARCHAR2	-	-	HZ_ LOCATIONS.LANGUAGE
LOCATION_ DIRECTIONS	N	VARCHAR2	-	-	HZ_ LOCATIONS.LOCATION_DIRECTIONS
PO_BOX_NUMBER	N	VARCHAR2	-	-	HZ_LOCATIONS.PO_ BOX_NUMBER
POSITION	N	VARCHAR2	-	-	HZ_ LOCATIONS.POSITION
POSTAL_PLUS4_ CODE	N	VARCHAR2	-	-	HZ_ LOCATIONS.POSTAL_ PLUS4_CODE
SALES_TAX_ GEOCODE	N	VARCHAR2	-	-	HZ_ LOCATIONS.SALES_ TAX_GEOCODE
SALES_TAX_INSIDE_ CITY_LIMITS	N	VARCHAR2	-	-	HZ_ LOCATIONS.SALES_ TAX_INSIDE_CITY_ LIMITS
SHORT_ DESCRIPTION	N	VARCHAR2	-	-	HZ_ LOCATIONS.SHORT_ DESCRIPTION
STREET	N	VARCHAR2	-	-	HZ_ LOCATIONS.STREET

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
STREET_NUMBER	N	VARCHAR2	-	-	HZ_ LOCATIONS.STREET_ NUMBER
STREET_SUFFIX	N	VARCHAR2	-	-	HZ_ LOCATIONS.STREET_ SUFFIX
SUITE	N	VARCHAR2	-	-	HZ_ LOCATIONS.SUITE
TIME_ZONE	N	VARCHAR2	-	-	HZ_ LOCATIONS.TIME_ ZONE
LOC_VALIDATED_ FLAG	N	VARCHAR2	-	-	HZ_ LOCATIONS.VALIDA TED_FLAG
DUNS_NUMBER	N	NUMBER	-	-	HZ_PARTIES.DUNS_ NUMBER
GROUP_TYPE	N	VARCHAR2	-	-	HZ_PARTIES.GROUP_ TYPE
GSA_INDICATOR_ FLAG	N	VARCHAR2	-	-	HZ_PARTIES.GSA_ INDICATOR_FLAG
HQ_BRANCH_IND	N	VARCHAR2	select lookup_code from ar_lookups where lookup_type = HQ_ BRANCH_IND and enabled_flag = Y	-	HZ_PARTIES.HQ_ BRANCH_IND
JGZZ_FISCAL_CODE	N	VARCHAR2	-	-	HZ_PARTIES.JGZZ_ FISCAL_CODE
KNOWN_AS	N	VARCHAR2	-	-	HZ_ PARTIES.KNOWN_AS
KNOWN_AS2	N	VARCHAR2	-	-	HZ_ PARTIES.KNOWN_ AS2
KNOWN_AS3	N	VARCHAR2	-	-	HZ_ PARTIES.KNOWN_ AS3
KNOWN_AS4	N	VARCHAR2	-	-	HZ_ PARTIES.KNOWN_ AS4

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
KNOWN_AS5	N	VARCHAR2	-	-	HZ_ PARTIES.KNOWN_ AS5
LANGUAGE_NAME	N	VARCHAR2	-	-	HZ_ PARTIES.LANGUAGE _NAME
LAST_ORDERED_ DATE	N	DATE	-	-	HZ_PARTIES.LAST_ ORDERED_DATE
MISSION_ STATEMENT	N	VARCHAR2	-	-	HZ_ PARTIES.MISSION_ STATEMENT
ORGANIZATION_ NAME_PHONETIC	N	VARCHAR2	-	-	HZ_ PARTIES.ORGANIZAT ION_NAME_ PHONETIC
PARTY_NUMBER	N	VARCHAR2	-	-	HZ_PARTIES.PARTY_ NUMBER
PERSON_FIRST_ NAME_PHONETIC	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ FIRST_NAME_ PHONETIC
PERSON_IDEN_TYPE	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ IDEN_TYPE
PERSON_ IDENTIFIER	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ IDENTIFIER
PERSON_LAST_ NAME_PHONETIC	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ LAST_NAME_ PHONETIC
PERSON_NAME_ SUFFIX	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ NAME_SUFFIX
PERSON_PREVIOUS_ LAST_NAME	N	VARCHAR2	-	-	HZ_ PARTIES.PERSON_ PREVIOUS_LAST_ NAME

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
PARTY_ REFERENCE_USE_ FLAG	N	VARCHAR2	-	-	HZ_ PARTIES.reference_ use_flag
SIC_CODE_TYPE	N	VARCHAR2	select lookup_code from ar_lookups where lookup_type = SIC_ CODE_TYPE and enabled_flag = Y	-	HZ_PARTIES.SIC_ CODE_TYPE
TAX_NAME	N	VARCHAR2	-	-	HZ_PARTIES.TAX_ NAME
TOTAL_NUM_OF_ ORDERS	N	NUMBER	-	-	HZ_PARTIES.TOTAL_ NUM_OF_ORDERS
TOTAL_ORDERED_ AMOUNT	N	NUMBER	-	-	HZ_PARTIES.TOTAL_ ORDERED_AMOUNT
PARTIES_ VALIDATED_FLAG	N	VARCHAR2	-	-	HZ_ PARTIES.VALIDATED _FLAG
PS_USES_ COMMENTS	N	VARCHAR2	-	-	HZ_PARTY_SITE_ USES.COMMENTS
PRIMARY_PER_TYPE	N	VARCHAR2	-	Y or N	HZ_PARTY_SITE_ USES.PRIMARY_PER_ TYPE
SITE_USE_TYPE	N	VARCHAR2	select lookup_code from ar_lookups where lookup_type=PARTY_ SITE_USE_CODE and enabled_flag = Y	This is a mandatory column for HZ_ PARTY_SITE_USES. But if not given, it is defaulted to BILL_TO.	HZ_PARTY_SITE_ USES.SITE_USE_TYPE
ADDRESSEE	N	VARCHAR2	-	-	HZ_PARTY_ SITES.addressee
MAILSTOP	N	VARCHAR2	-	-	HZ_PARTY_ SITES.MAILSTOP
PARTY_SITE_NAME	N	VARCHAR2	-	-	HZ_PARTY_ SITES.PARTY_SITE_ NAME
PARTY_SITE_ NUMBER	N	VARCHAR2	-	Can be populated ONLY when profile HZ: Generate Party Number is set to N	HZ_PARTY_ SITES.PARTY_SITE_ NUMBER
ORG_CNT_ COMMENTS	N	VARCHAR2	-	-	HZ_ORG_ CONTACTS.comments

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
CONTACT_NUMBER	N	VARCHAR2	-	-	HZ_ORG_ CONTACTS.CONTAC T_NUMBER
DECISION_MAKER_ FLAG	N	VARCHAR2	-	-	HZ_ORG_ CONTACTS.DECISIO N_MAKER_FLAG
DEPARTMENT	N	VARCHAR2	-	-	HZ_ORG_ CONTACTS.DEPART MENT
DEPARTMENT_ CODE	N	VARCHAR2	select lookup_code from ar_lookups where lookup_ type=DEPARTMENT_ TYPE and enabled_flag = Y	-	HZ_ORG_ CONTACTS.DEPART MENT_CODE
RANK	N	VARCHAR2	-	-	HZ_ORG_ CONTACTS.RANK
PROMOTION_ID	N	NUMBER	select source_code_id from ams_source_codes where active_flag = Y	If given, this is used for creating the sales lead header.	AS_SALES_ LEADS.source_ promotion_id
ROLE_LEVEL	N	VARCHAR2 (30)	-	-	HZ_ORG_CONTACT_ ROLES.role_level
PRIMARY_ CONTACT_PER_ ROLE_TYPE	N	VARCHAR2 (1)	-	Y or N	HZ_ORG_CONTACT_ ROLES.primary_ contact_per_role_type
CNT_PNT_ CONTENT_SOURCE_ TYPE	N	VARCHAR2 (30)	select lookup_code from ar_lookups where lookup_type =CONTENT_SOURCE_ TYPE and enabled_flag = Y	This is a mandatory column in HZ_ CONTACT_POINTS, but if not given, default it to USER_ENTERED	'HZ_CONTACT_ POINTS.CONTENT_ SOURCE_TYPE
PHONE_COUNTRY_ CODE	N	VARCHAR2 (30)	-	-	HZ_CONTACT_ POINTS.PHONE_ COUNTRY_CODE
FAX_COUNTRY_ CODE	N	VARCHAR2 (30)	-	-	HZ_CONTACT_ POINTS.PHONE_ COUNTRY_CODE
PHONE_CALLING_ CALENDAR	N	VARCHAR2 (30)	-	-	HZ_CONTACT_ POINTS.PHONE_ CALLING_ CALENDAR

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
CNT_PNT_TIME_ ZONE	N	NUMBER	-	-	HZ_CONTACT_ POINTS.TIME_ZONE
RAW_PHONE_ NUMBER	N	VARCHAR2 (60)	-	-	HZ_CONTACT_ POINTS.RAW_ PHONE_NUMBER
EMAIL_FORMAT	N	VARCHAR2 (30)	select lookup_code from ar_lookups where lookup_type=EMAIL_ FORMAT and enabled_ flag = Y	-	HZ_CONTACT_ POINTS.EMAIL_ FORMAT
FAX_EXTENSION	N	VARCHAR2 (20)	-	-	HZ_CONTACT_ POINTS.phone_ extension
ORG_CNT_TITLE	N	VARCHAR2	select lookup_code from ar_lookups where lookup_ type=CONTACT_TITLE and enabled_flag = Y	-	HZ_ORG_ CONTACTS.title
OFFER_ID	N	NUMBER	select source_code_id from ams_source_codes where ARC_SOURCE_ CODE_FOR = OFFR	-	AS_SALES_ LEAD.offer_id
PROMOTION_CODE	N	VARCHAR2 (50)	select source_code from ams_source_codes where active_flag = Y	If given and promotion_id is not provided, then this is transferred to promotion_id.	AS_SALES_ LEADS.source_ promotion_id
PARTY_ID	N	NUMBER(15)	-	Party Id to which the customer info is mapped	-
PARTY_SITE_ID	N	NUMBER(15)	-	Party Site Id to which the party and location info are mapped	-

Table 2–2 Columns Present in both the AS_IMPORT_INTERFACE Table and the Control File

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
LOCATION_ID	N	NUMBER(15)	-	Location Id to which address info is mapped	-
CONTACT_PARTY_ID	N	NUMBER	-	-	HZ_CONTACT_RELATIONSHIPS.subject_id
REL_PARTY_ID	N	NUMBER(15)	-	Party ID to which the relationship is mapped.	AS_SALES_LEAD_CONTACTS.contact_party_id & HZ_CONTACT_POINTS.owner_table_id & HZ_CONTACT_RESTRICTIONS.subject_id & HZ_CONTACT_POINTS.owner_table_id

Additional Columns in the AS_IMPORT_INTERFACE Table

Table 2–3 lists the columns present only in the interface table but not in the control file. Most of these columns are obsolete.

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
REQUEST_ID	N	NUMBER	-		-
PROGRAM_APPLICATION_ID	N	NUMBER	-	Do not populate	-
PROGRAM_ID	N	NUMBER	-	Concurrent process generated	-
PROGRAM_UPDATE_DATE	N	DATE	-		-
LOAD_ERROR_MESSAGE	N	VARCHAR2 (2000)	-	Currently, not used. To find the error messages, query from as_lead_import_errors	-
PHONE_ID	N	NUMBER	-	Do not populate	AS_SALES_LEAD_CONTACTS.phone_id
SECURITY_GROUP_ID	N	NUMBER	-	-	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
NET_WORTH	N	NUMBER	-	free text format	-
LEAD_NUMBER	N	VARCHAR2 (30)	-	-	AS_SALES_ LEADS.lead_number
PRM_SALES_LEAD_ TYPE	N	VARCHAR2 (30)	-	Partner specific	-
PRM_EXEC_ SPONSOR_FLAG	N	VARCHAR2 (1)	-	Partner specific	-
PRM_PRJ_LEAD_IN_ PLACE_FLAG	N	VARCHAR2 (1)	-	Partner specific	-
INCUMBENT_ PARTNER_PARTY_ ID	N	NUMBER	-	Partner specific	-
INCUMBENT_ PARTNER_ RESOURCE_ID	N	NUMBER	-	Partner specific	-
PRM_IND_ CLASSIFICATION_ CODE	N	VARCHAR2 (30)	-	Partner specific	-
PARTY_TYPE	N	VARCHAR2 (30)	-	If not given, it is defaulted to ORGANIZATION, if it is given but it is neither ORGANIZATION nor PERSON, results in an error.	HZ_PARTIES.party_ type
SALES_LEAD_ID	N	NUMBER(15)	-	<< Do not populate >> Sales Lead ID to which the sales lead info is mapped.	AS_SALES_ LEADS.sales_lead_id & AS_SALES_LEAD_ LINES.sales_lead_id & AS_SALES_LEAD_ CONTACTS.sales_ lead_id
NEW_PARTY_FLAG	N	NUMBER(1)	-	<< Do not populate >> If 1, party is newly created. If 0, existing party is used.	-
NEW_LOC_FLAG	N	NUMBER(1)	-	<< Do not populate >> If 1, this is a new location. If 0, this is already existing.	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
NEW_PS_FLAG	N	NUMBER(1)	-	<< Do not populate >> If 1, new PS, if 0, PS already existing	-
NEW_REL_FLAG	N	NUMBER(1)	-	<< Do not populate >> If 1, new relationship, if 0, existing Rel	-
NEW_CON_FLAG	N	NUMBER(1)	-	<< Do not populate >> If 1, new contact, if 0, existing Contact	-
HZ_PARTYID	N	NUMBER	-	-	AS_SALES_LEAD_ CONTACTS.CUSTOM ER_ID
HZ_PSID	N	NUMBER	-	-	AS_SALES_LEAD_ CONTACTS.ADDRESS _ID
The following columns are obsolete:	N		-	-	-
CUSTOMER_KEY	N	VARCHAR2 (50)	-	obsolete	-
ADDRESS_KEY	N	VARCHAR2 (500)	-	obsolete	-
CONTACT_KEY	N	VACHAR2(80)	-	obsolete	-
IMP_LIST_HEADER_ NUMBER	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_ID	N	NUMBER(15)	-	obsolete	-
CUSTOMER_ NUMBER	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_STATUS	N	VARCHAR2 (1)	-	obsolete	-
CUSTOMER_TYPE	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_ PROSPECT_CODE	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_CLASS_ CODE	N	VARCHAR2 (30)	-	obsolete	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
SALES_CHANNEL_ CODE	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_ GROUP_CODE	N	VARCHAR2 (30)	-	obsolete	-
CUSTOMER_ SUBGROUP_CODE	N	VARCHAR2 (30)	-	obsolete	-
REFERENCE_USE_ FLAG	N	VARCHAR2 (1)	-	obsolete	-
TAX_CODE	N	VARCHAR2 (50)	-	obsolete	-
THIRD_PARTY_ FLAG	N	VARCHAR2 (1)	-	obsolete	-
COMPETITOR_FLAG	N	VARCHAR2 (1)	-	obsolete	-
ADDRESS_ID	N	NUMBER(15)	-	obsolete	-
ADDRESS_STATUS	N	VARCHAR2 (1)	-	obsolete	-
SITE_CONTACT_ID	N	NUMBER(15)	-	obsolete	-
PHONE_STATUS	N	VARCHAR2 (1)	-	obsolete	-
SOURCE_ PROMOTION_ID	N	NUMBER	-	obsolete	-
INITIATING_ CONTACT_ID	N	NUMBER	-	obsolete	-
LEAD_RANK_CODE	N	VARCHAR2 (30)	-	obsolete	-
STATUS_CODE_1	N	VARCHAR2 (30)	-	obsolete	-
STATUS_CODE_2	N	VARCHAR2 (30)	-	obsolete	-
STATUS_CODE_3	N	VARCHAR2 (30)	-	obsolete	-
STATUS_CODE_4	N	VARCHAR2 (30)	-	obsolete	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
STATUS_CODE_5	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_1	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_1	N	NUMBER(3)	-	obsolete	-
TODO_DATE_1	N	DATE	-	obsolete	-
TODO_START_ TIME_1	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_1	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_2	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_2	N	NUMBER(3)	-	obsolete	-
TODO_DATE_2	N	DATE	-	obsolete	-
TODO_START_ TIME_2	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_2	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_3	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_3	N	NUMBER(3)	-	obsolete	-
TODO_DATE_3	N	DATE	-	obsolete	-
TODO_START_ TIME_3	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_3	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_4	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_4	N	NUMBER(3)	-	obsolete	-
TODO_DATE_4	N	DATE	-	obsolete	-
TODO_START_ TIME_4	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_4	N	VARCHAR2 (30)	-	obsolete	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
INTERACTION_ NOTES_5	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_5	N	NUMBER(3)	-	obsolete	-
TODO_DATE_5	N	DATE	-	obsolete	-
TODO_START_ TIME_5	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_5	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_6	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_6	N	NUMBER(3)	-	obsolete	-
TODO_DATE_6	N	DATE	-	obsolete	-
TODO_START_ TIME_6	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_6	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_7	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_7	N	NUMBER(3)	-	obsolete	-
TODO_DATE_7	N	DATE	-	obsolete	-
TODO_START_ TIME_7	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_7	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_8	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_8	N	NUMBER(3)	-	obsolete	-
TODO_DATE_8	N	DATE	-	obsolete	-
TODO_START_ TIME_8	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_8	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_9	N	VARCHAR2 (2000)	-	obsolete	-

Table 2–3 Additional Columns only in Table AS_IMPORT_INTERFACE

Column	Req. ?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
PRIORITY_9	N	NUMBER(3)	-	obsolete	-
TODO_DATE_9	N	DATE	-	obsolete	-
TODO_START_ TIME_9	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_9	N	VARCHAR2 (30)	-	obsolete	-
INTERACTION_ NOTES_10	N	VARCHAR2 (2000)	-	obsolete	-
PRIORITY_10	N	NUMBER(3)	-	obsolete	-
TODO_DATE_10	N	DATE	-	obsolete	-
TODO_START_ TIME_10	N	VARCHAR2 (5)	-	obsolete	-
ACTIVITY_CODE_10	N	VARCHAR2 (30)	-	obsolete	-
NUM_ENROLLED	N	NUMBER(5)	-	obsolete	-
CONFIRMED_FLAG	N	VARCHAR2 (1)	-	obsolete	-
PAYMENT_ AMOUNT	N	NUMBER	-	obsolete	-
NUM_ATTENDED	N	NUMBER(5)	-	obsolete	-
PROMOTION_NAME	N	VARCHAR2 (240)	-	obsolete	-
STATUS_CHANGE_ FLAG	N	VARCHAR2 (1)	-	obsolete	-

2.1.7.2 Table Structures for Multiple Contact Roles

Table structures for Multiple Contact Roles include:

- AS_IMP_CNT_ROL_INTERFACE
- AS_IMP_CNT_PNT_INTERFACE
- AS_IMP_LINES_INTERFACE

AS_IMP_CNT_ROL_INTERFACE

Table 2–4 lists the columns in the AS_IMP_CNT_ROL_INTERFACE table. Any values you populate in this interface table are imported along with the values in the main interface table.

Table 2–4 Columns in Table AS_IMP_CNT_ROL_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
IMP_CNT_ROL_ INTERFACE_ID	No	NUMBER	-	-	-
IMPORT_INTERFACE_ ID	No	NUMBER	-	-	-
LAST_UPDATE_DATE	No	DATE	-	Standard Who column	-
LAST_UPDATED_BY	No	NUMBER	-	Standard Who column	-
CREATION_DATE	No	DATE	-	Standard Who column	-
CREATED_BY	No	NUMBER	-	Standard Who column	-
LAST_UPDATE_LOGIN	No	NUMBER	-	Standard Who column	-
REQUEST_ID	Yes	NUMBER	-		-
PROGRAM_ APPLICATION_ID	Yes	NUMBER	-	Application identifier of the last concurrent program	-
PROGRAM_ID	Yes	NUMBER	-	Identifier of the last concurrent program to update record	-
PROGRAM_UPDATE_ DATE	Yes	DATE	-	Last date the concurrent program updated record	-
ORG_CONTACT_ID	Yes	NUMBER	-	Do not populate	-

Table 2–4 Columns in Table AS_IMP_CNT_ROL_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
ORG_CONTACT_ ROLE_ID	Yes	NUMBER	-	Do not populate	-
ORIG_SYSTEM_ REFERENCE	Yes	VARCHAR2(240)	-	-	HZ_ORG_ CONTACT_ ROLES.orig_ system_reference
PRIMARY_CONTACT_ PER_ROLE_TYPE	Yes	VARCHAR2(1)	Y or N	-	HZ_ORG_ CONTACT_ ROLES.primary_ contact_per_role_ type
PRIMARY_FLAG	Yes	VARCHAR2(1)	Y or N	-	HZ_ORG_ CONTACT_ ROLES.primary_ flag
ROLE_LEVEL	Yes	VARCHAR2(30)	-	-	HZ_ORG_ CONTACT_ ROLES.role_level
ROLE_TYPE	Yes	VARCHAR2(30)	select lookup_code from ar_ lookups where lookup_type = CONTACT_ ROLE_TYPE	-	HZ_ORG_ CONTACT_ ROLES.role_type

AS_IMP_CNT_PNT_INTERFACE

Table 2–5 lists the columns in the AS_IMP_CNT_PNT_INTERFACE table. Any values you populate in this interface table are imported along with the values in the main interface table.

Table 2–5 Columns in Table AS_IMP_CNT_PNT_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
IMP_CNT_PNT_ INTERFACE_ID	No	NUMBER	-	-	-
IMPORT_ INTERFACE_ID	No	NUMBER	-	-	-
OWNER_TYPE	No	VARCHAR2(30)	-	CONTACT or PERSON	-
PARTY_ID	Yes	NUMBER	-	Person's party ID	-
LAST_UPDATE_ DATE	No	DATE	-	Standard Who column	-
LAST_UPDATED_BY	No	NUMBER	-	Standard Who column	-
CREATION_DATE	No	DATE	-	Standard Who column	-
CREATED_BY	No	NUMBER	-	Standard Who column	-
LAST_UPDATE_ LOGIN	No	NUMBER	-	Standard Who column	-
REQUEST_ID	Yes	NUMBER	-	-	-
PROGRAM_ APPLICATION_ID	Yes	NUMBER	-	Application identifier of the last concurrent program	-
PROGRAM_ID	Yes	NUMBER	-	Identifier of the last concurrent program to update record	-
PROGRAM_ UPDATE_DATE	Yes	DATE	-	Last date the concurrent program updated record	-
ORIG_SYSTEM_ REFERENCE	Yes	VARCHAR2(240)	-	-	HZ_CONTACT_ POINTS.orig_ system_reference

Table 2–5 Columns in Table AS_IMP_CNT_PNT_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
CONTACT_POINT_TYPE	Yes	VARCHAR2(30)	select lookup_code from ar_lookups where lookup_type=COMMUNICATION_TYPE	-	HZ_CONTACT_POINTS.CONTACT_POINT_TYPE
EMAIL_FORMAT	Yes	VARCHAR2(30)	select lookup_code from ar_lookups where lookup_type=EMAIL_FORMAT	-	HZ_CONTACT_POINTS.EMAIL_FORMAT
EMAIL_ADDRESS	Yes	VARCHAR2(2000)	-	-	HZ_CONTACT_POINTS.EMAIL_ADDRESS
WEB_TYPE	Yes	VARCHAR2(60)	-	-	HZ_CONTACT_POINTS.WEB_TYPE
URL	Yes	VARCHAR2(2000)	-	-	HZ_CONTACT_POINTS.URL
PHONE_LINE_TYPE	Yes	VARCHAR2(30)	select lookup_code from ar_lookups where lookup_type = PHONE_LINE_TYPE	-	HZ_CONTACT_POINTS.PHONE_LINE_TYPE
PHONE_COUNTRY_CODE	Yes	VARCHAR2(30)	-	-	HZ_CONTACT_POINTS.PHONE_COUNTRY_CODE

Table 2–5 Columns in Table AS_IMP_CNT_PNT_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
PHONE_AREA_CODE	Yes	VARCHAR2(40)	-	-	HZ_CONTACT_POINTS.PHONE_AREA_CODE
PHONE_NUMBER	Yes	VARCHAR2(240)	-	-	HZ_CONTACT_POINTS.PHONE_NUMBER
PHONE_EXTENSION	Yes	VARCHAR2(20)	-	-	HZ_CONTACT_POINTS.PHONE_EXTENSION
RAW_PHONE_NUMBER	Yes	VARCHAR2(60)	-	-	HZ_CONTACT_POINTS.RAW_PHONE_NUMBER
OWNER_TABLE_NAME	Yes	VARCHAR2(30)	select lookup_code from ar_lookups where lookup_type = OWNER_TABLE_NAME	If not given, use HZ_PARTIES	HZ_CONTACT_POINTS.OWNER_TABLE_NAME
OWNER_TABLE_ID	Yes	NUMBER	-	If not given, use relationship's party ID for party_type=ORGANIZATION; or party ID for others	HZ_CONTACT_POINTS.OWNER_TABLE_ID
PRIMARY_FLAG	Yes	VARCHAR2(1)	-	-	HZ_CONTACT_POINTS.PRIMARY_FLAG

Table 2–5 Columns in Table AS_IMP_CNT_PNT_INTERFACE

Column	Can be NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
CONTENT_ SOURCE_TYPE	Yes	VARCHAR2(30)	select lookup_code from ar_ lookups where lookup_ type=CONTENT_ SOURCE_ TYPE	-	HZ_CONTACT_ POINTS.CONTENT_SOURCE_TYPE
PHONE_CALLING_ CALENDAR	Yes	VARCHAR2(30)	-	-	HZ_CONTACT_ POINTS.PHONE_ CALLING_ CALENDAR
TELEX_NUMBER	Yes	VARCHAR2(50)	-	-	HZ_CONTACT_ POINTS.TELEX_ NUMBER
TIME_ZONE	Yes	NUMBER	-	-	HZ_CONTACT_ POINTS.TIME_ ZONE
CONTACT_POINT_ PURPOSE	Yes	VARCHAR2(30)	if contact_ point_type <> WEB: select lookup_code from ar_ lookups where lookup_type = CONTACT_ POINT_ PURPOSE	-	HZ_CONTACT_ POINTS.WH_ UPDATE_DATE
PRIMARY_BY_ PURPOSE	Yes	VARCHAR2(30)	-	-	-
TRANSPOSED_ PHONE_NUMBER	Yes	VARCHAR2(60)	-	-	-

AS_IMP_LINES_INTERFACE

Table 2–6 lists the columns in the AS_IMP_LINES_INTERFACE table. Any values you populate in this interface table is imported along with the values in the main interface table.

Table 2–6 Columns in Table AS_IMP_LINES_INTERFACE

Column	Can Leave NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
IMP_LINES_ INTERFACE_ID	No	NUMBER	-	-	-
IMPORT_INTERFACE_ ID	No	NUMBER	-	-	-
LAST_UPDATE_DATE	No	DATE	-	Standard Who column	-
LAST_UPDATED_BY	No	NUMBER	-	Standard Who column	-
CREATION_DATE	No	DATE	-	Standard Who column	-
CREATED_BY	No	NUMBER	-	Standard Who column	-
LAST_UPDATE_LOGIN	No	NUMBER	-	Standard Who column	-
REQUEST_ID	Yes	NUMBER	-	-	-
PROGRAM_ APPLICATION_ID	Yes	NUMBER	-	-	-
PROGRAM_ID	Yes	NUMBER	-	-	-
PROGRAM_UPDATE_ DATE	Yes	DATE	-	-	-
INTEREST_TYPE_ID	No	-	select interest_ type_id from as_interest_ types_b where ENABLED_ FLAG = Y and EXPECTED_ PURCHASE_ FLAG = Y	-	AS_SALES_ LEAD_ LINES.INTER EST_TYPE_ID

Table 2–6 Columns in Table AS_IMP_LINES_INTERFACE

Column	Can Leave NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
PRIMARY_INTEREST_CODE_ID	Yes	-	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = and Pic.Parent_Interest_Code_Id Is Null and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.PRIMARY_INTEREST_CODE_ID
SECONDARY_INTEREST_CODE_ID	Yes	-	select Interest_Code_Id from As_Interest_Codes_B where Interest_Type_Id = And Parent_Interest_Code_Id = and ENABLED_FLAG = Y	-	AS_SALES_LEAD_LINES.SECONDARY_INTEREST_CODE_ID
INVENTORY_ITEM_ID	Yes	-	select inventory_item_id, organization_id from mtl_system_items	-	AS_SALES_LEAD_LINES.INVENTORY_ITEM_ID
ORGANIZATION_ID	Yes	-	-	-	AS_SALES_LEAD_LINES.ORGANIZATION_ID
UOM_CODE	Yes	-	select uom_code from mtl_units_of_measure	-	AS_SALES_LEAD_LINES.UOM_CODE
QUANTITY	Yes	-	-	-	AS_SALES_LEAD_LINES.QUANTITY

Table 2–6 Columns in Table AS_IMP_LINES_INTERFACE

Column	Can Leave NULL?	Data Type	Select for LOV or List of Valid Values	Comments	Target Table.column
BUDGET_AMOUNT	Yes	-	-	-	AS_SALES_ LEAD_ LINES.BUDGET_AMOUNT
SOURCE_ PROMOTION_ID	Yes	-	select source_ code_id from ams_source_ codes where active_flag = Y	-	AS_SALES_ LEAD_ LINES.SOURCE_ PROMOTION_ID
OFFER_ID	Yes	-	select source_ code_id from ams_source_ codes where active_flag = Y and ARC_ SOURCE_ CODE_FOR = OFFR	-	AS_SALES_ LEAD_ LINES.OFFER_ID

2.1.7.3 Lead Import Errors Table Structure

Table 2–7 lists the columns in the AS_LEAD_IMPORT_ERRORS table which stores error messages recorded during lead import.

Table 2–7 Columns in Table AS_LEAD_IMPORT_ERRORS

Column	Can leave NULL?	Data Type	Comments
LEAD_IMPORT_ERROR_ ID	No	NUMBER	-
LAST_UPDATED_BY	No	NUMBER	Standard Who column
LAST_UPDATE_DATE	No	DATE	Standard Who column
CREATION_DATE	No	DATE	Standard Who column
CREATED_BY	No	NUMBER	Standard Who column
LAST_UPDATE_LOGIN	No	NUMBER	Standard Who column

Table 2–7 Columns in Table AS_LEAD_IMPORT_ERRORS

Column	Can leave NULL?	Data Type	Comments
IMPORT_INTERFACE_ID	No	NUMBER	-
BATCH_ID	No	NUMBER	-
ERROR_TEXT	Yes	VARCHAR2(2000)	Stores error messages from lead import
SECURITY_GROUP_ID	Yes	NUMBER	-
REQUEST_ID	Yes	NUMBER	-
PROGRAM_ APPLICATION_ID	Yes	NUMBER	-
PROGRAM_ID	Yes	NUMBER	-
PROGRAM_UPDATE_ DATE	Yes	DATE	-

2.1.7.4 Sequence Generators

The following functions generate sequences for lead import tables:

- AS_SL_IMP_BATCH_S: Populates the batch ID.

  Note: Any program used to populate the interface tables must use AS_SL_IMP_BATCH_S to generate the sequence for any given batch. All the records in a batch must have the same Batch ID.

- AS_LEAD_IMPORT_ERRORS_S: Populates the PK for AS_LEAD_IMPORT_ ERRORS table.
- AS_IMP_SL_FLEX_S: Populates the PK for AS_IMP_SL_FLEX table.
- AS_IMPORT_INTERFACE_S: Can be used to populate the PK for AS_ IMPORT_INTERFACE_S.
- AS_IMP_CNT_ROL_INTERFACE_S: Can be used to populate the PK for AS_ IMP_CNT_ROL_INTERFACE.
- AS_IMP_CNT_PNT_INTERFACE_S: Can be used to populate the PK for AS_ IMP_CNT_PNT_INTERFACE.
- AS_IMP_LINES_INTERFACE_S: Can be used to populate the PK for AS_IMP_ LINES_INTERFACE.

2.2 Data Quality

An information system is only as good as the data which resides within it. In Oracle Leads Management, any lead records that are imported go through rigorous screening and filtering. The records are checked for:

- Leads Data Quality - performed by the Import Sales Lead concurrent program.
- Customer Data Quality - performed by Data Quality Management (DQM).

Table 2–8 gives the sequence in which the Customer and Leads data quality checks are performed.

Table 2–8 Checking for Data Quality in imported Lead Records

Task Performed	Program
Checking for Original System Reference Duplication	Import Sales Lead concurrent program
Checking for Duplicate Customers	DQM
Checking for Duplicate Addresses	DQM
Checking for Contacts and Contact Points	DQM
Lead Import Deduplication	Import Sales Lead concurrent program
Custom Deduplication Using User Hooks	Import Sales Lead concurrent program

2.2.1 Leads Data Quality

While importing a lead, the Import Sales Lead concurrent program also imports customer, address, contact and contact point information. Because the customer being imported may already be present in the system, the Import Sales Lead concurrent program checks for their existence before creating them. The customer entities that are checked for existence are: Customer (Organization or Person), Address, Contacts, and Contact Points.

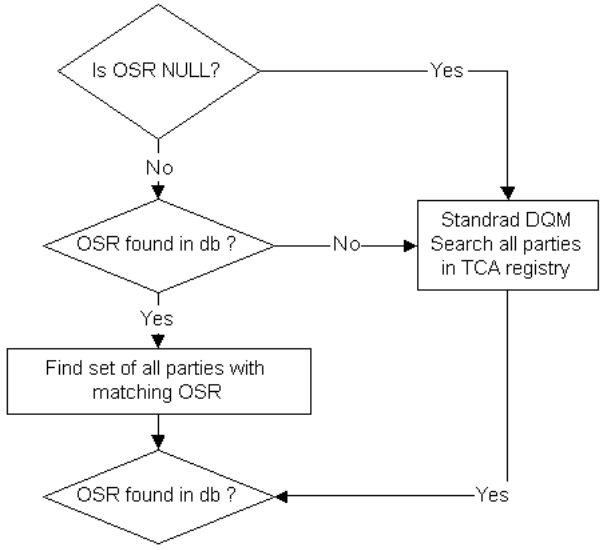
In the earlier releases, the Import Sales Lead concurrent program used fuzzy keys to identify the existence of these entities. Now, rule-based Data Quality Management (DQM) APIs are used for existence checking. For more information, see Section 2.2.4, "DQM Duplication Check".

2.2.1.1 Checking for Original System Reference Duplication

As part of DQM rule attributes, Original System Reference (OSR) is not included. So, the OSR check is done outside DQM.

Figure 2–3 gives the logic followed by the Import Sales Lead concurrent program before it starts matching party and address.

Figure 2–3 OSR Logic Used by Import Sales Lead Concurrent Program



If the OSR that is passed in matches with one or more parties in the HZ_PARTIES table, the Import Sales Lead concurrent program passes in the set of parties that match the OSR. The set of parties is passed in using the `p_restrict_sql` parameter in FIND_PARTIES API.

If the OSR is passed in and no matching set is found, or if the OSR is not passed at all, then the Import Sales Lead concurrent program scans through the TCA registry to find a match. If a match is found, the Import Sales Lead concurrent program reuses the party_id. If a match is not found, it creates a new party.

Note: It is recommended that you pass OSR in an import record, if known. This substantially improves the performance of the Import Sales Lead concurrent program.

2.2.2 Lead Import Deduplication

The Import Sales Lead concurrent program considers the following attributes while checking for duplicate leads:

- Customer Name
- Country
- Address 1
- Contact Last Name
- Contact First Name
- Contact E-mail
- Contact Phone (if E-mail null)
- Source Code
- Vehicle Response
- Creation Date (in the last 7 days)
- First Lead Note (on the same creation date as the lead)

2.2.2.1 The Dedupe Algorithm

The dedupe algorithm used by the Import Sales Lead concurrent program is given below:

1. Look for duplicate contact (last name, first name) AND e-mail address. If duplicate, continue checking. Else unique lead.
   - If e-mail is null, check phone. If duplicate, continue checking. Else unique lead.
   - If phone is null, unique lead.
2. Look at customer (name, address, and country). If duplicate, continue checking. Else unique lead.
3. Look at creation date. If the matching lead(s) is created within 7 days of the creation date of the compared lead, continue checking. Else unique lead.

4. Look at all other fields (source code, vehicle response, first lead note). If all are duplicate, lead is duplicate. Else unique lead.

Examples

The Leads dedupe algorithm is explained in Table 2–9 with examples. The Comments column in the table gives the result for each instance. The example does not take the Creation Date into consideration.

Table 2–9 Deduplication Examples

Customer	Country	Address	Last Name	First Name	E-mail	Area Code	Phone	Source Code	Vehicle Response	Note	Comments
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	Note 1	Master Lead
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	Note 1	Exact Duplicate Lead
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	-	805	408 0498	1010015	E-mail	Note 1	Duplicate Lead (phone checked)
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	222	546 2298	1010015	E-mail	Note 1	Duplicate Lead (phone not checked)
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	-	-	-	1010015	E-mail	Note 1	Unique Lead (e-mail and phone null)
Digital Harvester	US	500 Oracle Pkwy	Irvin	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	Note 1	Unique Lead (fails first dup check)

Table 2–9 Deduplication Examples

Customer	Country	Address	Last Name	First Name	E-mail	Area Code	Phone	Source Code	Vehicle Response	Note	Comments
Digital Harvester	GB	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	Note 1	Unique Lead (fails customer dup check)
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	New Note 1	Unique Lead (fails first note check)
Digital Harvester	US	500 Oracle Pkwy	Lorna	Bennie	lb@yahoo.com	805	408 0498	1010015	E-mail	Note 1New Note 2	Duplicate Lead (second note not checked)

Variants in any other fields do not affect this check (such as Role, Source System, Postal Code, SIC code, and so on).

Contact Restrictions

The Import Sales Lead concurrent program sets the do_not_phone_flag, do_not_fax_flag, do_not_email_flag, and do_not_mail_flag for the contact (relationship party_id). It also sets the do_not_mail_flag for the address (party_site).

No existence checking is done.

Contact Points

Lead Import creates the Phone, E-mail, Web, and Fax contact points.

2.2.3 Custom Deduplication Using User Hooks

User hooks permit you to bypass Oracle code and implement custom code instead. Use the following user hook to implement custom code and check for duplicate leads.

Hook name: IS_DUPLICATE_LEAD

Package Name: AS_IMPORT_SL_CUHK

Purpose

While importing leads, the Import Sales Lead concurrent program does not check for duplicate leads that may be stored in other third-party applications.

To implement custom lead duplicate checking, write a package according to the following specifications.

The Import Sales Lead concurrent program creates either a new lead or skips a lead import record based on the value returned by your program. If the record is skipped, then no lead is created and load_status of that lead import record is set to DUPLICATE.

Do not commit in the package body. After the transaction is completed, Oracle Application code issues a commit.

This user hook is called by the Import Sales Lead concurrent program.

Calling package

AS_IMPORT_SL_PVT.Is_Duplicate_Lead

API name

Is_Duplicate_Lead_Pre

In parameters

The four parameters below are standard inputs:

Parameter	Description
p_api_version_number	For 11*i* Oracle Sales application, this is 2.0.
p_init_msg_list	Initialize message stack or not. This is set to FND_API.G_FALSE by default.
p_validation_level	Validation level for pass-in values. Set to FND_API.G_VALID_ LEVEL_FULL by default.
p_commit	To commit the whole API at the end of API, set to FND_API. G_ FALSE by default.

The following parameter does not have a standard input:

Parameter	Description
p_import_interface_id	The import interface identifier. Pass into import_interface_id of the lead import record for which you want to perform the lead existence checking.

Out Parameters

The following three parameters are standard output parameters.

Parameter	Description
x_return_status	The return status. If your code completes successfully, then FND_API.G_RET_STS_SUCCESS must be returned. If you get an expected error, then return FND_API.G_RET_STS_ERROR, otherwise return FND_API.G_RET_STS_UNEXP_ERROR.
x_msg_count	The message count. Call FND_MSG_PUB.Count_And_Get to get the message count and messages.
x_msg_data	The messages. Call FND_MSG_PUB.Count_And_Get to get the message count and messages.

The following parameter does not have a standard output:

Parameter	Description
x_duplicate_flag	`Y` indicates the lead import record is a duplicate and was not imported. `N` indicates the lead import record is not a duplicate and was imported.

2.2.4 DQM Duplication Check

Prior to release 11.5.9, the Import Sales Lead concurrent program used fuzzy keys that were generated on the lead attributes. The key generation is now obsolete. Instead, rule-based matching is performed using Data Quality Management (DQM). The rules that are used are dependent upon the profiles that are set.

For more detailed information on DQM, see the *Oracle Trading Community Architecture Data Quality Management User Guide.*

Sections in this topic include:

- Section 2.2.4.1, "Checking for Duplicate Customers"

- Section 2.2.4.2, "Checking for Duplicate Addresses"
- Section 2.2.4.3, "Checking for Contacts and Contact Points"

2.2.4.1 Checking for Duplicate Customers

The Import Sales Lead concurrent program calls the HZ_PARTY_SEARCH.FIND_PARTIES API to run the rules that find duplicate customers. The Organization name or first name and last name is passed in along with the address-related information in the party site record. If the lead is created for Organization, the contact information is also passed in to find a better match on the party. The API call returns the context ID and the number of matches found. The parties are returned and populated in the HZ_MATCHED_PARTIES_GT table sorted on score. If the number of matches found is greater than zero, the context ID is used to get the match details. The highest score will have the best match. If multiple parties with the same high scores are found, the party that was created last is picked up. If the lead is created for Person, the contact information is not passed in.

2.2.4.2 Checking for Duplicate Addresses

Once the party is found, the get_matching_party_sites() API is called to check if the matching address exists. The party sites are returned and populated in the HZ_MATCHED_PARTY_SITES_GT table. If a match is found, the existing location_id and party site ID are used. If no match is found, a new location and party site is created.

When a lead is created for Organization, the best-matched party is found by calling HZ_PARTY_SEARCH.FIND_PARTIES. The matching contacts are found in the HZ_MATCHED_CONTACTS_GT table by calling the get_matching_contacts() API. If a matching contact is found, the party ID of the contact is retrieved from HZ_RELATIONSHIPS. The subject ID (the person ID of the contact) is passed in while calling the get_matching_contact_points API. The API is called along with the contact points available and the results are returned in the HZ_MATCHED_CPTS_GT table.

2.2.4.3 Checking for Contacts and Contact Points

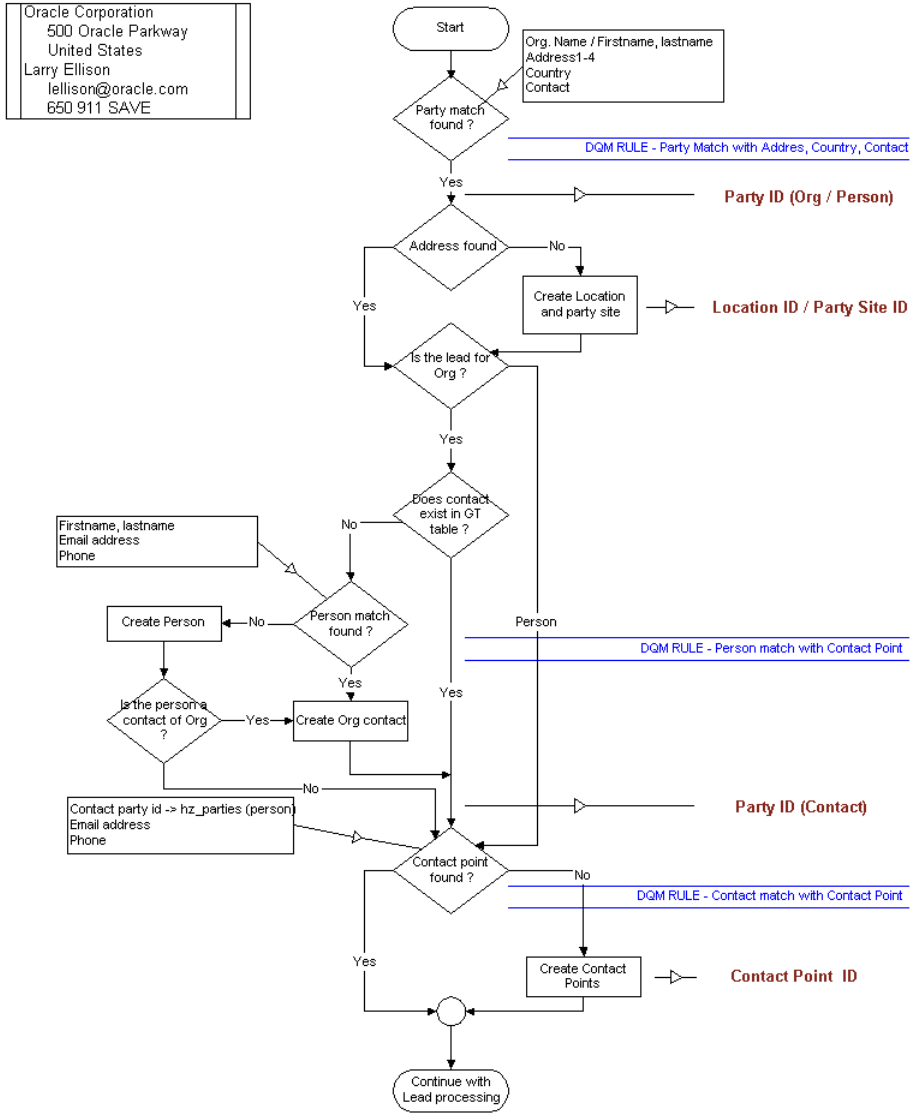
If the lead is created for Person and an existing party is found, the party ID of the person along with the contact points are passed while calling the get_matching_contact_points(). The matched contact points are found in the HZ_MATCHED_CPTS_GT table.

If the lead is created for an Organization and there is no contact Person found for the Organization, a search is performed to find a matching contact Person existing

in the database. In order to find a match, the contact person's first name, last name and contact points are passed in while calling find_parties. To find out a duplicate contact person, you must create a new rule which takes the details of the contact person name and the contact points.

Figure 2–4 depicts how the DQM program checks for duplicate entities in the database.

Figure 2–4 How DQM Checks for Duplicate Entities



2.2.5 Designing Matching Rules to Detect Duplicate Leads

Use the following as templates while designing DQM matching rules for the leads import program. Matching rules can also be tailored according to specific requirements using other seeded and custom attributes. For more detailed information, see the *Oracle Trading Community Architecture Data Quality Management User Guide*.

2.2.5.1 Matching Rule to Identify Duplicate Parties

Use the following sample rule to identify existence of party (Organization) and addresses for the same. Since this rule is also used to identify the existence of addresses, it must contain PARTY_SITE entity attributes as well as PARTY entity attributes.

Match Rule Name: LEAD_DUP_PARTY

Description: Finds duplicate organization parties based on Party Name and Address information

Purpose: To identify duplicate organizations and addresses

Table 2–10, Table 2–11, and Table 2–12 give the attributes for this rule.

Table 2–10 Existence Checking for Party: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match All	105	<null>	<null>

Table 2–11 Existence Checking for Party: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Type	PARTY	EXACT	Catches format errors	Lookup
Party Name	PARTY	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case, removes vowels, and double letters	-

Table 2–11 Existence Checking for Party: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
-	-	WR NAMES + CLUSTER	Captures the exact string, removes non-alphanumeric characters, forces upper case, and keeps the first four characters of the first two words.	-
Country	PARTY_SITE	EXACT	Catches format errors.	Lookup

Table 2–12 Existence Checking for Party: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Party Name	PARTY	100	EXACT_STRING	Captures the exact string, removes non-alphanumeric characters, and forces upper case.	-	100
-	-	-	WR NAMES	Person and Org names with word replacement.	-	90
-	-	-	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case, and removes vowels and double letters.	-	70

Table 2–12 Existence Checking for Party: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
-	-	-	WR NAMES + CLUSTER	Captures the exact string, removes non-alphanumeric characters, forces upper case, and keeps the first four characters of the first two words.	-	50
Address	PARTY_SITE	30	WR ADDRESS	Address with word replacement.	Custom Attribute	100
-	-	-	WR ADDRESS + CLEANSE	Address with word replacement, and removes vowels and double letters.	-	70
Postal Code	PARTY_SITE	20	EXACT	Catches format errors.	-	100
Country	PARTY_SITE	30	EXACT	Catches format errors.	Lookup	100

Apart from the specified attributes in the above sample matching rule, more party (organization) related attributes like DUNS Number, Tax Reference, and SIC Code can be specified as per custom requirements.

2.2.5.2 Matching Rule to Identify Duplicate Persons

Since Party Type is an Acquisition attribute, the above sample matching rule can also be used for person existence checking. You may add more Person-related attributes to the matching rule as per custom requirements such as Person Name, Person First Name, Person Last Name, and so on. The PARTY_SITE entity attributes must be specified since the same rule is used to identify existence of addresses for a specified person.

Match Rule Name: LEAD_DUP_PERSON

Description: Finds duplicate persons based on Person Name & Address Information

Purpose: To identify duplicate persons and addresses

Table 2–13, Table 2–14, and Table 2–15 give the attributes for this rule.

Table 2–13 Existence Checking for Person: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match All	105	<null>	<null>

Table 2–14 Existence Checking for Person: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Type	PARTY	EXACT	Catches format errors	Lookup
Party Name	PARTY	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case and removes vowels and double letters.	-
-	-	WR NAMES + CLUSTER	Captures the exact string, removes non-alphanumeric characters, forces upper case, and keeps the first four characters of the first two words.	-
Country	PARTY_SITE	EXACT	Catches format errors.	Lookup

Table 2–15 Existence Checking for Person: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Party Name	PARTY	100	EXACT_STRING	Captures the exact string, removes non-alphanumeric characters, and forces upper case.	-	100
-	-	-	WR NAMES	Person and Org names with word replacement.	-	90
-	-	-	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case, and removes vowels and double letters.	-	70
-	-	-	WR NAMES + CLUSTER	Captures the exact string, removes non-alphanumeric characters, forces upper case and keeps the first four characters of the first two words.	-	50
Address	PARTY_SITE	30	WR ADDRESS	Address with word replacement.	Custom Attribute	100
-	-	-	WR ADDRESS + CLEANSE	Address with word replacement and removes vowels and double letters.	-	70

Table 2–15 Existence Checking for Person: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Postal Code	PARTY _SITE	20	EXACT	Catches format errors.	-	100
Country	PARTY _SITE	30	EXACT	Catches format errors.	Lookup	100

2.2.5.3 Matching Rule to Identify Duplicate Contacts

Use the following sample matching rule to identify duplicate Contacts and Contact Points like e-mail, phone number, fax number and URL. Because the same rule is used to identify Contact Points, specify the CONTACT_POINT entity attributes while designing the matching rule for the identification of contacts.

Match Rule Name: LEAD_DUP_CONTACT

Description: Finds duplicate contacts based on Contact Name and Contact Points.

Purpose: To identify duplicate contact and contact points

Table 2–16, Table 2–17, and Table 2–18 give the attributes for this rule.

Table 2–16 Existence Checking for Contact: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match All	50	<null>	<null>

Table 2–17 Existence Checking for Contact: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Name	PARTY	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case, and removes vowels and double letters.	-

Table 2–17 Existence Checking for Contact: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
-	-	WR NAMES + CLUSTER	Captures the exact string, removes non-alphanumeric characters, forces upper case, and keeps the first four characters of the first two words.	-
Contact Name	CONTACT	WR PERSON + CLEANSE	Person word replacement and removes vowels and double letters.	Custom Attribute
-	-	WR PERSON + CLUSTER	Person word replacement and keeps the first four characters of the first two words.	-
E-mail Address	CONTACT_ POINT	CLEANSE (E-mail)	Removes non-alphanumeric characters, white spaces, and removes vowels and double letters.	-
Phone Number	CONTACT_ POINT	EXACT	Removes non-alphanumeric characters and white spaces.	-
URL	CONTACT_ POINT	CLEANSE (URL)	Removes non-alphanumeric characters, white spaces, and removes vowels and double letters.	-

Table 2–18 Existence Checking for Contact: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Contact Name	CONT ACT	50	WR PERSON	Person with word replacement.	Custom Attribut e	100

Table 2–18 Existence Checking for Contact: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
-	-	-	WR PERSON + CLEANSE	Person word replacement and removes vowels and double letters.	-	90
-	-	-	WR PERSON + CLUSTER	Person word replacement and keeps the first four characters of the first two words.	-	70
E-mail Address	CONTACT_POINT	10	EXACT (E-mail)	Forces uppercase.	-	100
Phone Number	CONTACT_POINT	10	EXACT	Removes non-alphanumeric characters and white spaces.	-	100
URL	CONTACT_POINT	10	CLEANSE (URL)	Removes non-alphanumeric characters, white spaces, vowels and double letters.	-	100

Note: For existence checking of Party (Org & Person), Addresses, Contact and Contact Points, the Import Sales Lead concurrent program solely depends on the results returned by the DQM matching rules. The more effective the matching rule, the more precise the result.

2.3 Processing and Distribution of Leads

Sections in this topic include:

- Section 2.3.1, "Setting Up the Lead Rules Engine"
- Section 2.3.2, "Qualification Rule Sets"

2.3.1 Setting Up the Lead Rules Engine

Setting up the lead rules engine comprises setting up the qualification engine, the rating engine, and the channel selection engine.

- The Qualification Engine confirms that there is sufficient interest for a selling interface to engage the prospect through a touch point. See Section 2.3.2.1, "Qualification Engine Overview".
- The Rating Engine grades leads using business-specific logic appropriate to different campaign strategies, regions, or products. Rating enables the business to prioritize follow-up and response handling for effective cost and resource management. See Section 2.3.3.1, "Rating Engine Overview".
- The Channel Selection Engine channels the lead to a set of resources for assignment: direct or indirect, depending on its quality or domain. A sales channel is direct when routed to the sales force of a company and indirect when routed to partners or value added distributors. See Section 2.3.4.1, "Sales Channel Selection Engine Overview".

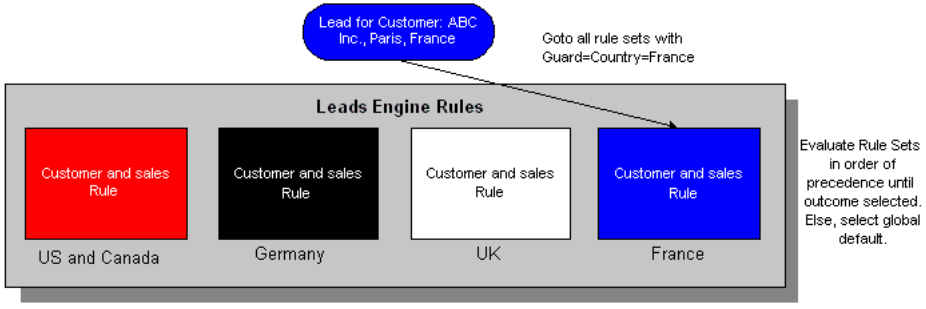
Note: The leads rules engine runs only if the profile OS: Run new Lead Qualification/Rating/Channel Engine is set to `Y`.

The engines are based on a generic rules model, which consists of Guards, Precedence, and Rules.

- Rule Guards are used to group rule sets into domain-specific buckets. These parse rule sets into groups based on business-specific practices. Each rule set defines the set of leads to which it applies, including product-specific, campaign-specific, and country-specific lead processing logic for each stage of lead evaluation.

  For example, for country-specific guards:

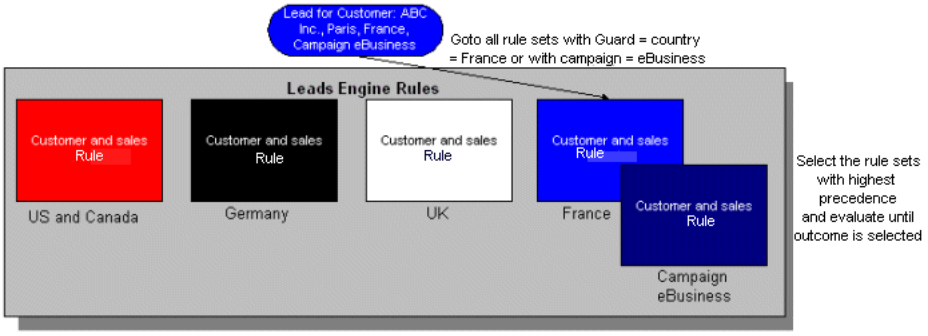
Figure 2–5 Lead Engine Example 1



- Once the rule sets are bucketed into different groups, the Precedence of each rule set is used to determine the order of evaluation. Overlapping filters enables the business to ensure that the highest priority regional, customer, or marketing strategy is used to evaluate the lead.

  For example, by assigning a Campaign-specific rule set a higher precedence, this rule set is evaluated before a region-specific rule set.

Figure 2–6 Lead Engine Example 2



On evaluation, the winning rule set with the highest precedence is used to select the rule set result. If more than one winning rule set has equal precedence, the best, or highest ranked result, wins.

- Once the correct rule sets are selected, the rules of each rule set determine the conditions and action to be performed on the lead. For example, if certain conditions are true at the time of evaluation, the lead is set to qualified, or graded A, or channelized to Direct Sales.

2.3.2 Qualification Rule Sets

The qualification engine is run only if the Qualified_flag = `N` and the profile OS: Auto Qualify Lead is set to `Y`.

This section includes:

- Section 2.3.2.1, "Qualification Engine Overview"
- Section 2.3.2.2, "Creating a Qualification Rule Set"

2.3.2.1 Qualification Engine Overview

When a lead is run through the Rules Engine, the Qualification Engine first identifies the correct qualification rule sets to evaluate by processing the guards of the rule sets. This process finds all matching rule sets by applying the lead attribute values against the guard values of each qualification rule set . For example, find all rule sets for Campaign A or B, or all rule sets for Product Y.

Note: Guards can have multiple qualifying conditions. There is an implicit AND across conditions and an implicit OR within condition. For example, if Guard is Country = *France*, *Germany*, *UK*; Product Category = *Printers*, *Desktops*, this is interpreted as evaluate all leads that originate from countries *France* or *Germany* or *UK*, AND for product lines *Printer* or *Desktops*.

Once the matching qualification rule sets are identified, the engine starts evaluating the Rules of each rule set, starting with the rule set of the highest precedence.

Note: There is an implicit AND across conditions and an implicit OR within a condition.

When a rule set wins, i.e., all the qualification rules of the rule set are met for the lead, the engine stops evaluation. The lead is then qualified and the winning rule set is logged into a history table for analysis.

If no rule sets win, by default the lead is set to *Not Qualified*.

Note: Only qualified leads go on to the Rating and Channel Selection engines, and assignment. For unqualified leads, the immature lead handling process kicks in and handles assignment, and builds a repository of unqualified leads for the selected resource.

2.3.2.2 Creating a Qualification Rule Set

Use this procedure to enable the automatic qualification of leads.

Prerequisites

Set the system profile OS: Auto Qualify Lead to `Y`. This is the default setting.

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Qualification Rules

Steps

Set up one or more qualification rule sets for lead qualification.

1. Click **Create**.
2. In the Qualification Rule Set Name field, enter a name.
3. In the Description field, enter a description for the rule set.
4. In the Start Date and End Date fields, enter dates.

   These are the dates between which the rule set is valid.
5. Open the Status drop-down list, and select a status for the rule set.
6. In the Precedence field, enter a number.

   Each rule set can have different precedences, to define order of importance for evaluation (where 100 is higher than 1).
7. Open the Currency drop-down list, and select a currency to be associated with this rule set.
8. In the Owner field, enter a name.
   a. Click **Go**.
   b. Select the owner to own this rule set.
9. Click **Create**.

   The qualification rule set is created. You must now add Guards and Rules to the rule set.

See Also

- Section 2.3.5, "Guards and Rules for a Rule Set"
- Section 2.3.7, "Searching for a Rule Set"

2.3.3 Rating Rule Sets

This section includes:

- Section 2.3.3.1, "Rating Engine Overview"
- Section 2.3.3.2, "Setting up Ratings"
- Section 2.3.3.3, "Creating a Rating Rule Set"

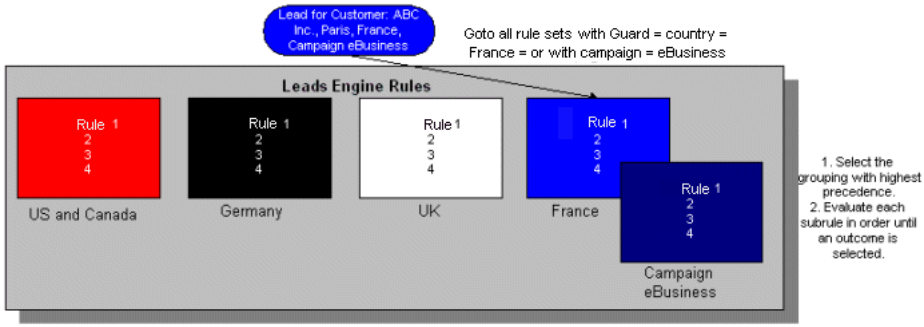
2.3.3.1 Rating Engine Overview

The rating engine differs from the qualification engine. For each rating rule set, there are many rules, each of which has its own precedence, or order of evaluation. These rules effectively comprise the rule set for each grouping (such as France, or Campaign e-business).

Each Rule Set is a grouping of rules. The rule set is defined by its Guard. The rules define the criteria and grade outcome.

For example:

Figure 2–7 Rating Engine Example 1



When a lead is run through the Rating Engine, the engine first identifies the correct Rating rule set to evaluate. This process finds all matching rule sets by applying the lead attribute values against each rule set's guard values. For example, find all rule sets for Campaign A, or all rule sets for Product Y.

Note: Guards can have multiple conditions. There is an implicit AND across conditions and an implicit OR within conditions. For example, if the Guard is defined as Country = *France*, *Germany*, *UK*; Product Category = *Printers*, *Desktops*, then this is interpreted as evaluate all leads that originate from countries *France* or *Germany* or *UK* for product lines *Printer* or *Desktops*.

Once the matching Rating rule sets are identified, the engine starts evaluating the rules for each rule set, starting with the rule set of highest precedence. The Rating

engine evaluates the rules in the order of evaluation assigned and stops when it finds a rule that wins (matches the lead).

Note: There is an implicit AND across conditions and an implicit OR within conditions. For example, if Rating rule is defined as Condition(s) for Customer/Account Type = *Non Named Account*; Budget Amount less than *50K*, then this is interpreted as evaluate all leads for *non named accounts* with Budget Amount less than *50K*.

When a rule wins, that is, all the criteria are met for the lead, the Rating engine stops evaluation. The lead is then graded and the winning rule set is logged into a history table for analysis.

If more than one rule set wins with equal precedence, the highest ranked grade is selected.

If no rule sets win, the default rating set in OS: Default Rating for Lead Rating Engine is used.

2.3.3.2 Setting up Ratings

Use this procedure to set up ratings.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Setup > Rating

Steps

1. In the Grade field, enter a grade for the lead.

   The lead is rated with this grade.

2. In the Description field, enter a brief description of the grade.
3. In the Precedence field, enter a number to set the precedence for the rating.

The precedence is used when the attributes of the lead match with more than one grade. In this case, the lead is assigned the grade with a higher precedence. For precedence, 100 is higher than 1.

4. Select the Enabled check box to include this grade to rate leads.
5. Click **Update**.
6. Click **Restore** to clear a grade that you are setting up.

   The grade is cleared.
7. Select the check box in the Remove column to remove a grade.
8. Click **Update**.

   The grade is removed.

2.3.3.3 Creating a Rating Rule Set

Create one or more rule sets for lead rating. Each rule set can have a different precedence, to define order of importance for evaluation (where 100 is higher than 1).

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Rating Rules

Steps

1. Click **Create**.
2. In the Rating Rule Set Name field, enter the name by which the rule set will be identified.
3. In the Description field, enter a brief description of the rule set.
4. In the Start Date and End Date fields, enter dates.

   These are the dates between which the rule set is valid.
5. Open the Status drop-down list, and select a status for the rule set.

6. In the Precedence field, enter a number.

   Each rule set can have a different precedence, to define the order of importance for evaluation (where 100 is higher than 1).

7. Open the Currency drop-down list to select the currency to be associated with this rule set.

8. In the Owner field, enter a name.

   a. Click **Go**.

   b. Select the owner to own this rule set.

9. Click **Create**.

   The rating rule set is created. You must now add Guards and Rules to the rule set.

> **Note:** Rules are evaluated in rankings from 1-n, where 1 is evaluated first. For a complete list of supported conditions, see Section 2.3.8, "Rule Engine Qualifiers".

The rating engine is run only when rank is null and the profile OS: Rank Lead is set to `System`.

See Also

- Section 2.3.5, "Guards and Rules for a Rule Set"
- Section 2.3.7, "Searching for a Rule Set"

2.3.4 Channel Selection Rule Sets

This section includes:

- Section 2.3.4.1, "Sales Channel Selection Engine Overview"
- Section 2.3.4.2, "Setting Up Channels"
- Section 2.3.4.3, "Creating a Channel Selection Rule Set"

2.3.4.1 Sales Channel Selection Engine Overview

Sales Channel Selection is similar to the Rating Engine in that for each channel selection rule set, the criteria are a set of rules. Each rule contains attributes used for

determining the best channel for a lead and is associated with a channel selection outcome.

When a lead is run through the Channel Selection Engine, the engine first identifies the correct rule sets to evaluate. This process finds all matching rule sets by applying the lead attribute values against each rule set's guard values.

Once the matching channel rule sets are identified, the engine starts evaluating the rules, starting with the rule of highest precedence. Each rule has an order of evaluation associated with it. The Channel Selection Engine evaluates the rules in that order and stops when it finds a rule that matches the lead.

When a rule wins, i.e., all the criteria are met for the lead, the engine stops evaluating channel rule sets. The lead is then channeled and the winning rule set is logged into a history table for analysis.

If more than one rule set wins with equal precedence, the highest ranked channel is selected.

If no rule sets win, the default grade set in the profile OS: Default Channel for Lead Channel Selection Engine is used.

2.3.4.2 Setting Up Channels

Use this procedure to set up channels.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Setup > Channel

Steps

1. Open the Channel drop-down list, and select a channel.
2. Select the Indirect check box if this is an indirect channel.
3. In the Precedence field, enter a number to set the precedence for this channel.

The precedence is used when the attributes of a lead match with more than one channel. In this case, the lead is assigned the channel with a higher precedence. For precedence, 100 is higher than 1.

4. Click **Update**.
5. Click **Restore** to clear a channel that you are setting up.

   The channel is cleared.
6. Select the check box in the Remove column to remove a channel.
7. Click **Update**.

   The channel is removed.

2.3.4.3 Creating a Channel Selection Rule Set

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Channel Selection Rules

Steps

1. Click **Create**.
2. In the Name field, enter the name by which the rule set will be identified.
3. In the Description field, enter a brief description of the rule set.
4. In the Start Date and End Date fields, enter dates.

   These are the dates between which the rule set is valid.
5. Open the Status drop-down list, and select a status for the rule set.
6. In the Precedence field, enter a number.

   Each rule set can have a different precedence, to define the order of importance for evaluation (where 100 is higher than 1).

7. Open the Currency drop-down list to select the currency to be associated with this rule set.
8. In the Owner field, enter a name.
   a. Click **Go**.
   b. Select the owner to own this rule set.
9. Click **Create**.

   The channel selection set is created. You must now add Guards and Rules to the rule set.

> **Note:** Rules are evaluated in precedences from 1-n, where 1 is evaluated first. For a complete list of supported conditions, see Section 2.3.8, "Rule Engine Qualifiers".

See Also

- Section 2.3.5, "Guards and Rules for a Rule Set"
- Section 2.3.7, "Searching for a Rule Set"

2.3.5 Guards and Rules for a Rule Set

Guards in a rule set are used to group rule sets into domain-specific buckets. These parse rule sets into groups based on business-specific practices. Each rule set defines the set of leads to which it applies, including product-, campaign-, and country-specific lead processing logic for each stage of lead evaluation.

Further, after the rule sets have been identified, the Rules in the rule sets are used to qualify the lead, rate the lead, or select a channel.

2.3.5.1 Creating a Guard or Rule for a Rule Set

Use this procedures to create, modify, or remove Guards and Rules in a rule set.

Prerequisites

You must have created the rule set. See

- Section 2.3.2.2, "Creating a Qualification Rule Set"
- Section 2.3.3.3, "Creating a Rating Rule Set"
- Section 2.3.4.3, "Creating a Channel Selection Rule Set"

Responsibility

None

Navigation

HTML - Administration > Leads > Processing Rules > Qualification/Rating/Channel Selection

Steps

1. Create the rule set.
2. Click the Guard or the Rule link.

   The Rule Set: <Rule Set name> Guard/Rule page appears.
3. Click **Add Attributes**.

   The Add Guard/Rule Profile Attribute Name page appears.
4. Open the Profile Attribute Name drop-down list, and select a profile.

   The page reloads to display the values for the profile that you have chosen.
5. Open the Condition drop-down list, and select a condition for the profile.
6. From the list of values displayed, select a value(s) for the profile.
7. Click **Apply**.

   The attribute is added.
8. Optionally, click **Apply And Add Another** to add another attribute.

   Follow steps 4 to 6.
9. To modify an attribute, click the hyperlinked Profile Attribute Name in the Add Guard/Rule Profile Attribute Name page.

   Follow steps 4 to 6.
10. To remove an attribute, select it from the Remove column in the Add Guard/Rule Profile Attribute Name page.

    Click **Update**.

    The attribute is removed.

Reference

For a complete list of supported attributes, see Section 2.3.8, "Rule Engine Qualifiers".

2.3.6 Copying Rule Sets

You can copy a rule set to make a duplicate one. The guards and rules are also copied. Typically, you would copy a rule set when you want another similar rule set with minor differences. Use the following procedure to copy a rule set successfully.

Prerequisites

None

Responsibility

None

Navigation

HTML - Administration > Leads > Processing Rules > Qualification/Rating/Channel Selection > Hyperlinked Rule Set Name

Steps

1. From the Rule Set Details page, click **Copy**.
2. Make changes to the rule set details, as required.
3. Click **Copy**.

   The rule set is created.
4. Modify the Guard and Rule details, as required.

   See Section 2.3.5, "Guards and Rules for a Rule Set" for more details.

2.3.7 Searching for a Rule Set

You can search for any rule sets - rating, qualification, and channel selection- using the Search feature. You can search on the following attributes of the rule set - name, owner, status, rule set valid date, customer category, and country.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Qualification/Rating/Channel Selection

Steps

1. In the Name field, enter the name of the rule set.
2. In the Owner field, enter the name of the owner of the rule set.
3. Open the Status drop-down list, and select the status of the rule set.
4. In the Start Date and End Date fields, enter dates.

   These are the dates between which the rule set is valid.
5. Open the Customer Category drop-down list, and select the customer category for the rule set that you are searching for.
6. Open the Country drop-down list, and select the country.
7. Click **Search**.

   The results are displayed based on your search criteria.
8. Optionally, click **Clear** to clear the values in all the Search fields.

2.3.8 Rule Engine Qualifiers

Table 2–19 lists the supported Guard attributes for the leads rules engine.

Table 2–19 Guard Attributes for the Lead Rules Engine

Attribute	Table.Column	Links
Creation Date	as_sales_ leads.creation_ date	-
Country	hz_ locations.country	From as_sales_leads link to hz_party_sites mapping Address_id and party_site_id From hz_party_sites link to hz_locations mapping Location_id with location_id
Product Interest (Purchase Line)	as_sales_lead_ lines.interest_ type_id	-

Table 2–19 Guard Attributes for the Lead Rules Engine

Attribute	Table.Column	Links
Campaign	as_sales_lead.source_promotion_id	-
Lead Status	as_sales_leads.status_code	-
City	hz_location.city	From as_sales_leads link to hz_party_sites mapping address_id and party_site_id From hz_party_sites link to hz_locations mapping location_id with location_id
County	hz_location.county	From as_sales_leads link to hz_party_sites mapping address_id and party_site_id From hz_party_sites link to hz_locations mapping location_id with location_id
State	hz_location.state	From as_sales_leads link to hz_party_sites mapping address_id and party_site_id From hz_party_sites link to hz_locations mapping location_id with location_id
Province	hz_location.province	From as_sales_leads link to hz_party_sites mapping address_id and party_site_id From hz_party_sites link to hz_locations mapping location_id with location_id
Postal Code	hz_location.postal_code	From as_sales_leads link to hz_party_sites mapping address_id and party_site_id From hz_party_sites link to hz_locations mapping location_id with location_id
Customer Category	hz_parties.category_code	Seeded lookup type Customer Category
Area Code	hz_contact_points.phone_area_code	From as_sales_leads link to hz_contact_points mapping owner_table_id and customer_id
Primary Contact E-mail Address	hz_contact_points.email_address	From as_sales_leads link to hz_contact_points mapping owner_table_id and customer_id

Table 2–20 lists the rule attributes for the Leads Rules Engine.

Table 2–20 Rule Attributes for the lead rules engine

Attribute	Table.Column	Links
Customer Category	hz_parties.category_code	Seeded lookup type Customer Category
Customer Annual Revenue	hz_organization_profiles.curr_fy_potential_revenue	From as_sales_leads link to hz_organization_profiles mapping Customer_id and party_id
Primary Contact Specified	as_sales_leads.primary_contact_party_id	As_sales_leads to hz_parties.party_id
Purchase Timeframe	as_sales_lead_contacts. contact_role_code	From as_sales_leads link to as_sales_lead_contacts mapping Primary_contact_party_id and contact_party_id Code is from lookup type `lead_contact_role`
Budget Status	as_sales_leads.budget_status_code	-
Lead Score (Applicable for imported leads only)	as_import_interface.Marketing_score	-
Lead Status	as_sales_leads.status_code	-
Total Budget	as_sales_leads.budget_amount	-
Purchase Quantity	as_sales_lead_lines.quantity	-
Purchase Amount	as_sales_lead_lines.budget_amount	From as_sales_leads link to as_sales_lead_lines mapping sales_lead_id and sales_lead_id
Response Channel	as_sales_leads.vehicle_response_code	Seeded data from lookup type `vehicle response code`
Project	As_sales_leads.parent_project	-
Qualify Flag	as_sales_leads.qualified_flag	-

Table 2–20 Rule Attributes for the lead rules engine

Attribute	Table.Column	Links
Lead Grading/Rating	as_sales_leads. lead_ rank_id to as_sales_ ranks_b.rank_id	-
Sales Channel	as_sales_leads. channel_code	Seeded data from lookup type `sales channel`
Creation Date	as_sales_leads. creation_date	-
State	hz_locations. state	From hz_party_sites link to as_ sales_leads mapping Party_site_id and address_id From hz_locations link to hz_ party_sites mapping Location_id and location_id
Primary Contact Phone	as_sales_lead_ contacts.phone_id	links to as_sales_leads.phone_id
Customer/Account Type	Hz_code_ assugnments.class_ code	Links as_sales_leads.customer_id to hz_code_assignments for owner_table_id and owner_table_ name = HZ_PARTIES
Country	hz_locations.country	From as_sales_leads link to hz_ party_sites mapping Address_id and party_site_id From hz_party_sites link to hz_ locations mapping Location_id with location_id
Campaign	as_sales_lead. source_ promotion_id	Links to ams_source_ codes.source_code_id
Customer Name	Hz_parties.party_ name	Links as_sales_leads.customer_id to hz_parties.party_id
Customer Address	Hz_party_sites.party_ site_id	Links as_sales_leads.address_id to hz_party_sites.party_site_id for the customer party_id
Product Interest	As_sales_lead_ lines.interest_types_ id + primary_ interest_code_id + secondary_interest_ code_id	Links to as_interest_types and as_ interest_codes

Table 2–20 Rule Attributes for the lead rules engine

Attribute	Table.Column	Links
Total Amount	sum(as_sales_lead_ lines. budget_ amount)	From as_sales_lead_lines link to as_sales_leads mapping Sales_ lead_id and sales_lead_id
City	hz_location.city	From as_sales_leads link to hz_ party_sites mapping address_id and party_site_id From hz_party_sites link to hz_ locations mapping location_id with location_id
County	hz_location.county	From as_sales_leads link to hz_ party_sites mapping address_id and party_site_id From hz_party_sites link to hz_ locations mapping location_id with location_id
Province	hz_location.province	From as_sales_leads link to hz_ party_sites mapping address_id and party_site_id From hz_party_sites link to hz_ locations mapping location_id with location_id
Postal Code	hz_location.postal_ code	From as_sales_leads link to hz_ party_sites mapping address_id and party_site_id From hz_party_sites link to hz_ locations mapping location_id with location_id
Area Code	hz_contact_ points.phone_area_ code	From as_sales_leads link to hz_ contact_points mapping owner_ table_id and customer_id
Primary Contact E-mail Address	hz_contact_ points.email_address	From as_sales_leads link to hz_ contact_points mapping owner_ table_id and customer_id
Primary Contact E-mail or Phone	hz_contact_ points.contact_point_ id	From as_sales_leads link to hz_ contact_points mapping owner_ table_id and customer_id

2.3.8.1 Using Derived Attributes

If you wish to add more attributes apart from the seeded attributes listed in Table 2–23, use derived attributes.

Prerequisites

None

Responsibility

Vendor Administrator

Navigation

HTML - Administration > Partner > Attributes > Create

Steps

1. Define derived attributes.

   For information on defining attributes, see the *Oracle Partners Implementation Guide*, Set up Custom Attributes.

2. Use these attributes while creating a rule set.

   You can also use customer classification for more flexible customer-related attributes.

2.3.9 Migrating Lead Engines from 11.5.8

In version 11.5.8, using the profile OS: Run New Lead Qualification, Rating, Channel Selection Engines, you were able to choose between using Scorecards and Rating Rules to rank leads.

In version 11.5.9, you cannot work with Scorecards and Lead Rank mapping since it has been obsoleted. However, you can view read-only versions of the Scorecard and Lead Rank mapping data for reference purposes.

To view your old Scorecard and Lead Rank mapping data, set the OS: Lead View Scorecard Data profile to `Y`. This displays buttons on the Lead Rating Setup page.

2.3.10 Migrating Lead Engines from Versions prior to 11.5.8

Sections in this topic include:

- Section 2.3.10.1, "Migrating to the New Qualification Rules Engine"

- Section 2.3.10.2, "Migrating to the New Rating Rules Engine"
- Section 2.3.10.3, "Migrating to the New Channel Selection Engine"

2.3.10.1 Migrating to the New Qualification Rules Engine

In 11.5.7 and prior versions, the lead qualification process used the following profile options:

- OS: Project name required (AS_SALES_LEAD_PROJECT_NAME_REQ)
- OS: Sales channel required (AS_SALES_LEAD_CHANNEL_REQ)
- OS: Decision time frame required (AS_SALES_LEAD_TIME_FRAME_REQ)
- OS: Total budget amount required (AS_SALES_LEAD_TOTAL_BUDGET_REQ)
- OS: Contact phone required (AS_SALES_LEAD_CONTACT_PHONE_REQ)
- OS: Contact role required (AS_SALES_LEAD_CONTACT_ROLE_REQ)
- OS: Budget status required (AS_SALES_LEAD_BUDGET_STATUS_REQ)
- OS: Campaign code required (AS_SALES_LEAD_CAMPAIGN_CODE_REQ)

This section discusses how to map these options to rule sets in the Rules Engine.

About Guards

One new qualification rule set can comprise the entire existing profile option set; however, the new Rules Engine can define multiple qualification rule sets to accommodate different business scenarios and segmentation, via Guards.

Each rule set requires at least one Guard. On evaluation, Guards are used to select rule sets appropriate for each lead. This groups rule sets into business-specific sets for lead evaluation. For example, country-specific, product-specific, or campaign-specific. This enables you to create multiple rule sets, each of which applies to different leads.

Among the profile options currently supported, only Campaign Code is a Guard in the new Qualification Engine. Therefore, if you have set the profile OS: Campaign code required to `Yes`, you can add the attribute Campaign as a rule Guard, and set the operator to `Is Not Null`. This way, all leads with a Campaign Code use the rule set.

If you do not use the profile OS: Campaign Code Required, you must still use at least one Guard in your rule set; otherwise, the rule set is not utilized. This can be either a specific Product or Country, or a check to ensure a field is not null. Or, if you wish a given rule to apply to ALL leads, you can select the attribute Creation

Date as a Guard, and set the operator to `Is Not Null` - this ensures the rule set is always used for each lead, for example, as a catch all rule.

Mapping Profile Options to Rule Engine Conditions

Each of the profile options is supported in the new Leads Rules Engine as Rule Conditions.

To map the profile options directly to conditions, you can create one qualification rule set that sets each of the qualifiers you require to <Qualifier> 'Is Not Null'.

For Rule Conditions, map the options as follows:

- If you set the profile OS: Project Name Required to `Yes`, set the attribute Project with operator `Is Not Null`.
- If you set the profile OS: Sales Channel Required to `Yes`, set the attribute Sales Channel with operator `Is Not Null`.
- If you set the profile OS: Decision Time Frame Required to `Yes`, set the attribute Purchase Timeframe with operator `Is Not Null`.
- If you set the profile OS: Total Budget Amount Required to `Yes`, set the attribute Total Budget with operator `Is Not Null`.
- If you set the profile OS: Contact Phone Required to `Yes`, set the attribute Contact Phone with operator `Is Not Null`.
- If you set the profile OS: Contact Role Required to `Yes`, set the attribute Contact Role with operator `Is Not Null`.
- If you set the profile OS: Budget status required to `Yes`, set the attributes Budget Status with operator `Is Not Null`.

Example:

Qualification Rule Set	Operator	Value
Name	-	Default Qualification Rule
Precedence	-	100
Status	-	Active
Currency Code	-	US Dollar
Guard = Campaign	Is Not Null	-

Qualification Rule Set	Operator	Value
Rule=	-	-
Sales Channel	Is Not Null	-
Purchase Time frame	Is Not Null	-
Total Budget	Is Not Null	-
Budget Status	Is Not Null	-

The above example maps the new Qualification Engine to the version 11.5.7 profile options. However, the new engine can define more flexible qualification conditions as well, using other operators, and other values.

Example:

Qualification Rule Set	Operator	Value
Name	-	New Qualification Rule
Precedence	-	100
Status	-	Active
Currency Code	-	US Dollar
Guard = Campaign	Equals	E-business
Rule =	-	-
Purchase Timeframe	Equals	1 to 3 months
Total Budget	Greater than	10,000
Budget Status	Equals	Approved

2.3.10.2 Migrating to the New Rating Rules Engine

In 11.5.7, the lead ranking process uses a scorecard with the following qualifiers:

- Organization
- Contact Role
- Timeframe
- Campaign Code
- Sales Channel

- Budget Status
- Budget Amount

These Name /value pairs are set to scores, which are then mapped to grades in the Rank Mapping.

This section discusses how to map a scorecard to rule sets in the new Rules Engine.

About Grading

Mapping the Scorecard and Rank Mapping framework to the new Rating Rules Engine is more complicated than qualification rule sets, because the new rule sets map conditions directly to grades (that is, no score total is used). This enables the user to map exact fields, and their different values, to grades, without merging different field/value logic accrued into a final score.

For example, a given lead can score low points on some fields, and high points on other fields. A total score merges this data, and therefore loses information about the relative weight of each field, which fields were missing, which answers were key, and so on. Or, a second given lead can score one high point but have low scores on all key fields, and derive a total high score. Again, the final total loses all distinctions.

The method in the new Rules Engine maps different fields, and their conditions, directly to grades. This way, each field and its value is controlled by the user in terms of its weight, its relative relevance to other fields, if it can be null or not, and so on.

For this reason, scores and their ranks cannot be directly mapped to rules, as more specific field rating logic is encapsulated in a rule.

Note: A Score attribute (from the Imports Interface table: Marketing_Score) is still supported in the new Rules Engine. This imported score can be mapped to grades in the rules engine along with any other fields. However, the score is not generated by the engine itself, and must be imported externally.

About Guards

Note that one Rating rule set can comprise the entire set of existing scorecard qualifiers; however, the new Rules Engine can also define multiple rule sets for different business scenarios and segmentation, via Guards.

As above, each rating rule set requires at least one Guard. On evaluation, Guards are used to select rule sets appropriate for each lead. For example, country-specific, product-specific, or campaign-specific. These, therefore enable the user to create multiple sets of rating rule sets, each of which applies to different leads.

Among the qualifiers currently supported, only Campaign Code is a Guard in the new Rating Engine. If you use the Campaign Code qualifier in your scorecard, you can add the attribute Campaign as a rule set Guard, and set the operator to Equals, and value to the value(s) you currently support. This way, all leads of the specified campaign are processed by this rule set.

> **Note:** You can add multiple campaigns per rule set, if all use the same conditions and precedence; or you can create multiple rule sets for different campaigns, each of which uses different rating logic.

If you do not use the Campaign qualifier, you must still use at least one Guard in your rating rule set; otherwise, the rule set will not be utilized. This can be either a specific Product or Country, or a check to ensure a field is not null. If you wish a given rule to apply to ALL leads, you can select the attribute Creation Date as a Guard, and set the operator to `Is Not Null` - this ensures the rule set is always used for each lead, for example, as a catch all rule set.

Mapping Score Qualifiers to Rule Conditions

Each of the Scorecard qualifiers is supported in the new Rating Rules Engine as Conditions.

To map the scorecard options directly to rule set conditions:

1. If you use the scorecard qualifier Organization, set the rating rule set attribute Customer Name with operator `Equals`, and value the same as you used in the scorecard.
2. If you use the scorecard qualifier Contact Role, set the rating rule set attribute Contact Role with operator `Equals`, and value the same as you used in the scorecard.
3. If you use the scorecard qualifier Timeframe, set the rating rule set attribute Purchase Timeframe with operator `Equals`, and value the same as you used in the scorecard.

4. If you use the scorecard qualifier Sales Channel, set the rating rule set attribute Sales Channel with operator `Equals`, and value the same as you used in the scorecard.
5. If you use the scorecard qualifier Budget Status, set the rating rule set attribute Budget Status with operator `Equals`, and value the same as you used in the scorecard.
6. If you use the scorecard qualifier Budget Amount, set the rating rule set attribute Total Budget with operator `Equals`, and value the same as you used in the scorecard.

To map a Scorecard and its Rank Mapping to the new engine, you need to create a rule set, and then translate the name/value pairs and their scores to the rule set's sub-rules. You define the outcome, or grade, of each sub-rule by the different field combinations you want to support.

For example, if you set up a scorecard as:

Qualifier	Score	Value
Organization	50	XYZ Company
Contact Role	20	Decision Maker
Time Frame	30	1-3 months
Campaign	20	E-business
Sales Channel	20	Indirect
Budget Status	10	Approved
Budget Amount	50	10,000 to 50,000

And your rank mapping is:

Min. Points	Max. Points	Rank
1	49	C
50	99	B
100	149	A
150	200	A+

You could set up one Rating rule set in the new engine as:

Rating Rule Set	Operator	Value
Name	-	New Rating Rule
Precedence	-	100
Status	-	Active
Currency Code	-	US Dollar
Guard	-	-
Campaign (was 20)	Equals	E-business
Rule 1 (A+)	-	-
Customer Name (was 50)	Equals	Decision Maker
Total Budget (was 50)	Between	10,000-50,000
Purchase Timeframe (was 30)	Equals	1-3 months
(total = 150)	-	-
Rule 2 (A)	-	-
Purchase Timeframe (was 30)	Equals	1-3 months
Sales Channel (was 20)	Equals	Indirect
Total Budget (was 50)	Between	10,000-50,000
(total = 120)	-	-
Rule 3 (B)	-	-
Budget Status (was 10)	Equals	Approved
Contact Role (was 20)	Equals	Decision Maker
Purchase Timeframe (was 30)	Equals	1-3 months
(total = 80)		

The table above is just one example. There are multiple ways to translate a scorecard into different grades. The following table also shows valid rule sets for a the above scorecard.

Rule Set	Operator	Value
Name	-	New Rating Rule
Precedence	-	100
Status	-	Active
Currency Code	-	US Dollar

Rule Set	Operator	Value
Guard	-	-
Campaign (was 20)	Equals	E-business
Rule 1 (A+)	-	-
Contact Role (was 20)	Equals	Decision Maker
Total Budget (was 50)	Between	10,000-50,000
Purchase Timeframe (was 30)	Equals	1-3 months
Sales Channel (was 20)	Equals	Indirect
Budget Status (was 10)	Equals	Approved
(total = 150)	-	-
Rule 2 (A)	-	-
Contact Role (was 20)	Equals	Decision Maker
Budget Amount (was 50)	Between	10,000 to 50,000
Budget Status (was 10)	Equals	Approved
(total = 120)		

2.3.10.3 Migrating to the New Channel Selection Engine

The Channel Selection Engine is new in 11.5.8. Define new rule sets as specified in Section 2.3.4, "Channel Selection Rule Sets".

2.3.11 Setting Up Automatic Lead Assignment

Use this procedure to set up the application to automatically assign resources to a lead whenever an agent or salesperson creates or updates the lead.

Prerequisites

None

Responsibility

Forms - Oracle Sales Administrator

Navigation

Forms - Profiles > System

Note: If the leads are processed by the Lead Rules Engine, a lead sales team is built only for qualified leads.

Steps

1. To enable automatic lead assignment, make sure that the system profile option OS: Assign New Lead is set to `N`. This is the default seeded value.

   When this profile is set to `N`, a call to the Territory Manager API automatically assigns resources to the lead using the territories defined in Territory Manager. The first person the program assigns becomes the lead owner. The rest of the resources in the territory become sales team members on the lead.

   If the lead creator is a valid sales agent or salesperson, the lead creator is added to the lead sales team when the lead is created.

2. For qualified and indirect leads, if the profile OS: Auto Convert Lead to Opportunity is set to `Y`, an opportunity is created, and partner matching workflow is launched.

3. For unqualified leads, if the channel is equal to the profile OS: Lead Incubation Channel, the lead owner is determined by immature lead assignment.

4. Set up the default resource to handle any leads that are not matched to any territory in the system profile option OS: Default Resource ID Used for Sales Lead Assignment.

   If there is no matching territory, then the application assigns the lead to the default resource set in this profile option. If you do not set this profile option, then the application assigns the lead to the agent or salesperson who created or updated the lead.

   Note: If both the resource in OS: Default Resource ID Used for Sales Lead Assignment and the user who created or updated the lead do not have a valid sales role assigned to them, then the leads you import will not be accessible from either Oracle Sales Online or Oracle TeleSales.

5. If the territories in your organization use agent availability as one of the criteria for assigning agents, then setting the profile OS: Calendar Setting Required for Assignment to `Yes` enables the automatic assignment of lead owners based on availability. If you are using agent availability for assigning lead ownership,

then you must make sure that each resource has a calendar set up for them. OS: Calendar Setting Required for Assignment is set to `No` by default.

For more details, see the *Oracle CRM Application Foundation Implementation Guide*, Implementing Calendar.

6. Optionally, you can implement custom rules for lead assignment by implementing the Lead Routing Engine user hook described below.

Lead Routing Engine

Hook Name: AS_LEAD_ROUTING_WF

Package Name: AS_LEAD_ROUTING_WF_CUHK

Purpose

If you are implementing custom lead routing rules, then create a package body according to these specifications.

> **Note:** Do not commit in this package body. After the transaction is complete, Oracle application code will issue a commit.

This user hook will be called when an agent or salesperson is creating and updating a lead in the Lead tab, and from the Import Sales Lead concurrent program whenever the routing engine is called.

The calling package is AS_LEAD_ROUTING_WF.GetOwner.

API name

Get_Owner_Pre

In Parameters

Table 2–21 lists the standard input parameters.

Table 2–21 Standard Input parameters

Parameter	Description
p_api_version_number	For 11*i* Oracle Sales applications, this is set to 2.0.
p_init_msg_list	Initialize message stack or not. Set to FND_API.G_FALSE by default.

Table 2–21 Standard Input parameters

Parameter	Description
p_validation_level	Validation level of pass-in value. Set to FND_API.G_VALID_LEVEL_FULL by default.
p_commit	Whether commit the whole API at the end of API. Set to FND_API.G_FALSE by default.

The following three parameters store the available resources for this customized package to decide the owner of the sales lead. Their data type is TABLE of NUMBERs.

- p_resource_id_tbl
- p_group_id_tbl
- p_person_id_tbl

Table 2–22 lists other input parameters.

Table 2–22 Other Input Parameters

Parameter	Description
p_resource_flag_tbl	This parameter specifies the source of the resource: ■ D: Default resource from the profile AS_DEFAULT_RESOURCE_ID, OS: Default Resource ID used for Sales Lead Assignment. ■ L: Login user. ■ T: Territory definition. If the sales lead matches any territory, the above parameters will include all the resources returned from territory engine and p_resource_flag_tbl will be all `T`. If the sales lead does not match any territory: **1.** Profile OS: Default Resource ID used for Sales Lead Assignment is set: ■ p_resource_id_tbl(1), p_group_id_tbl(1), p_person_id_tbl(1) is the default resource defined in this profile. ■ p_resource_flag_tbl(1)=`D` ■ p_resource_id_tbl(2), p_group_id_tbl(2), p_person_id_tbl(2) is the login user. ■ p_resource_flag_tbl(2)=`L` **2.** Profile OS: Default Resource ID used for Sales Lead Assignment is not set: ■ p_resource_id_tbl(1), p_group_id_tbl(1) ■ p_person_id_tbl(1) is the login user ■ p_resource_flag_tbl(1)=`L`
p_sales_lead_rec	This provides the whole definition of a sales lead. This record is provided to help an Oracle customer decide the sales lead owner.

Out Parameters

The following three parameters store the result of this user hook:

- x_resource_id
- x_group_id
- x_person_id

Together these set the sales lead owner.

If x_resource_id is NULL, the owner is decided based upon Oracle's logic.

For instance, x_resource_id=1001, x_group_id=10, x_person_id=100. The resource with the resource ID 1001, group ID 10, and person ID 100 is assigned as the owner of the sales lead.

Table 2–23 lists the standard output parameters.

Table 2–23 Standard Output Parameters

Parameter	Definition
x_return_status	The return status. If your code completes successfully, then FND_API.G_RET_STS_SUCCESS must be returned. If you get an expected error, then return FND_API.G_RET_STS_ERROR, otherwise return FND_API.G_RET_STS_UNEXP_ERROR.
x_msg_count	The message count. Call FND_MSG_PUB.Count_And_Get to get the message count and messages.
x_msg_data	The messages. Call FND_MSG_PUB.Count_And_Get to get the message count and messages.

2.3.12 Setting Up Immature Lead Assignment

Use this procedure to add one or more resources to manage immature leads.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Maturation Assignment

Steps

1. In the Add a resource field, enter the name of the resource.

   You can use the % wildcard.

2. Click **Go.**

   The Find a Sales Person page appears listing sales persons matching your search criteria.

3. Select one or more sales persons.

4. Click **Select**.

   The Immature Lead Assignment page appears, and the sales person(s) is added.

5. Select the Owner check box to indicate that a sales person must be the owner of the lead assigned.

6. Open the Country drop-down list, and select the country for this sales person.

7. In the Postal Code From and Postal Code To fields, enter the postal from and to codes.

8. Optionally, to remove a sales person, select the Remove check box.

   Click **Update**.

2.3.13 Setting Up Lead Status

Oracle TeleSales comes with a set of predefined lead statuses. These are:

- New
- In Progress
- Converted to Opportunity
- Dead Lead
- Loss

Use the procedure listed below to define alternate statuses. The procedure for creating lead statuses is almost identical to the procedure for setting up Opportunity Status.

Prerequisites

None

Responsibility

Forms - Oracle Sales Administrator

HTML - Oracle Sales Online Super User

Navigation

Forms - Administration > takes you to HTML

HTML - Administration > Sales > Opportunity > Status Code

Steps

1. Click **Create**.

   The Status Code Values page appears.

2. In the Status Code field, enter the status code name.

   Users cannot see this name. It is for internal use only.

3. In the Meaning field, enter the meaning.

   Users see this in the drop-down list when they select a lead status.

4. Add an optional description for the status.

5. Select the **Enabled** check box to enable this status.

   Note: Do not select either Open or Include in Forecast check boxes. These are reserved for setting up opportunity statuses.

6. In the Used for region, select the Sales Lead check box.

7. Select the Neither radio button in the Win Loss Indicator region.

   The other radio buttons are reserved for opportunity statuses.

8. Click **Create**.

2.3.14 Lead Status Changes for Customers Upgrading from Versions Prior to Version 1158

The progress of a lead is tracked by a combination of Lead Status and the Accepted and Qualified check boxes. In past releases, lead qualification depended on lead status only.

Lead statuses are upgraded for you automatically when you install the new version of the application.

Table 2–24 gives a list of old and new statuses. Here is an explanation of the table columns:

- **Old Status:** Status in previous releases.
- **New Status**: What the old status is being changed into.
- **Additional Action**: Additional updating by the script.

Table 2–24 Lead Status

Old Status	New Status	Additional Action
Qualified	In Progress	The Qualified check box is selected
Unqualified	In Progress	None
Accepted	In Progress	The Accepted check box is selected
Opportunity	Converted to Opportunity	None
Declined	New	None

2.3.15 Performing Mass Operations on Leads

You can mass update and evaluate a number of selected leads.

Mass Update

Using mass update, you can:

- **Mass Accept**

  Use this feature to select multiple leads, and accept them.

- **Mass Decline**

  Use this feature to select multiple leads, and decline them. You must provide a reason for declining the leads.

- **Mass Status Update**

  Use this feature to select multiple leads, and update their status. If the lead's current status is `Open`, and if you updated the status to `Close`, you are asked to state a reason for the same.

- **Mass Convert to Opportunity**

  Use this feature to select multiple leads, and convert them all to opportunities.

Mass Evaluate

You can select multiple leads and update them with the selected qualification, rank and channel values. If any of these fields are left blank, the corresponding rule engine is run and the lead is updated with the values.

You also have an option of selecting an owner and updating the list of leads with the new owner. If the owner is not selected, it is automatically selected and updated.

2.3.16 Setting Up Time Frames

This section includes:

- Section 2.3.16.1, "Time Frames Overview"
- Section 2.3.16.2, "Customizing Time Frames"

2.3.16.1 Time Frames Overview

Time Frames determine the expiration date of a lead. The expiration date assumes the maximum length of the time frame relative to the creation date. Previously, time frames were simply lookup codes. Now, mappings give a meaning to the time frame in terms of number of days.

Some seeded examples:

Within 1 week : 7 days

1-3 months : 90 days

For all seeded values, see Appendix B, "Seed Data".

2.3.16.2 Customizing Time Frames

The time frame periods can be customized to suit your requirements. However, the time frame itself cannot be modified, or new ones cannot be created. You must enable the time frames that you want to be used in your organization.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Setup > Timeframe

Steps

1. From the Time Frame column, select the time frame that you want to customize.
2. In the Days column, enter the new period.
3. Click **Update**.

   The time frame is added with the new period. There may be multiple periods defined for the same time frame.
4. Select the check box in the Enable column to enable this time frame.

   Note: Similar time frames cannot be enabled at the same time. Deselect any, if enabled.
5. Select a time frame from the Remove column to remove it.

   Note: Seeded time frames cannot be removed.
6. Click **Update**.
7. Click **Restore** to clear a time frame that you are customizing.

   The time frame is cleared.

2.4 Tracking

Sections in this topic include:

- Section 2.4.1, "Setting Up the Monitoring Engine"
- Section 2.4.2, "Administrative Operational Reports"

2.4.1 Setting Up the Monitoring Engine

This section includes:

- Section 2.4.1.1, "Monitoring Engine Overview"
- Section 2.4.1.2, "Scope of a Monitor"
- Section 2.4.1.3, "Concurrent Program"
- Section 2.4.1.4, "Creating a Monitor Rule"
- Section 2.4.1.5, "Monitoring Engine Conditions"

- Section 2.4.1.6, "Viewing the Monitor Log"
- Section 2.4.1.7, "Searching for a Monitor"

2.4.1.1 Monitoring Engine Overview

The Monitoring Engine enables the administrator to track the state of a lead, and ensure that each lead is acted upon in a timely manner. When a lead has not been acted upon, the monitoring engine can be set up to send reminders to the owner, or reroute the lead.

The lead monitor is set to trigger on two conditions:

- Lead is in Open status
- Lead is assigned a owner

When a condition is satisfied, the monitor is activated and a notification(s)/reminder(s) is sent to the lead owner and/or the owner's manager. If the lead is still not acted upon, the lead may be timed-out and rerouted.

The monitor conditions are the From and To States of a lead. The values supported for the From State are Creation and Assignment. Every time a new lead is created, the monitoring engine is called. If the lead matches a Creation From State monitor, the lead is attached to this monitor. For this lead, the monitors defined for Assignment From State are never used.

If the lead does not match any of the Creation From State monitors, then a matched Assignment From State monitor is chosen for the lead. Every time this lead is reassigned, the monitoring engine is called. At this time, any old Assignment From State monitors that may be active on the lead are terminated, and the workflow for the new monitor is launched. If a new monitor is not available for the new state, then no action is taken.

Based on the frequency specified, notifications or reminders are sent till the lead reaches the To State.

2.4.1.2 Scope of a Monitor

The scope of a monitor is defined by the Country and Rating attributes. A lead satisfying both the attributes is processed by the appropriate monitor. For example, if the scope of a monitor is Country=*US* and Rating=*A*, all leads that belong to the *US* and have an *A* rating are processed by this monitor.

A monitor with a single attribute is picked when no other monitors satisfy the condition. The preference though is always to select a monitor with both the attributes.

2.4.1.3 Concurrent Program

The Workflow Background Process concurrent program sends notifications and reminders from triggered monitors. The system administrator must schedule the concurrent program to run everyday, or twice a day, if required.

Details to run the concurrent program:

- Parameter - Monitoring Engine Workflow
- Process Deferred - Y
- Process Timeout - Y
- Process Stuck - Y

2.4.1.4 Creating a Monitor Rule

Use the following procedure to set up one or more monitoring rules for lead monitoring.

Prerequisites

Make sure the system profile OS:Max_Lead_Reroutes is set. The number assigned to this profile defines the maximum number of reroutes allowed to a lead that is timed out by the monitoring engine. If the lead does not have a owner after the maximum reroutes, it is routed to the default resource defined by the AS_DEFAULT_RESOURCE_ID profile.

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Monitoring Rules

Steps

1. Click **Create**.
2. In the Monitoring Rule Name field, enter the name by which the rule will be identified.
3. In the Description field, enter a brief description of the rule.
4. In the Valid From and To Date fields, enter dates.

   These are the dates between which the rule is valid.

5. Open the Status drop-down list, and select a status.
6. In the Owner field, enter a name.
   a. Click **Go**.
   b. Select the owner to own this rule.
7. In the Monitor Scope region:
   a. Open the Profile Attribute Name drop-down list, select the attribute name.
   b. Open the Condition drop-down list, select the condition.
   c. Open the Value drop-down list, select a Value for the attribute.

   For more information on the scope of a monitor, see Section 2.4.1.2, "Scope of a Monitor".
8. In the Monitor Conditions region, enter the conditions for the monitor.

   See Section 2.4.1.5, "Monitoring Engine Conditions".
9. In the Notification Recipients region:
   a. Select the Notify Owner check box.
   b. Select the Notify Owner Manager check box.
10. In the Reminders region:
    a. Select the No Reminder option.

       OR
    b. Select the Send number of Reminders option.
       – In the Number of Reminders field, enter a number.

         This indicates the number of reminders to send.
       – In the Frequency...Days field, enter a number.

         This indicates the frequency in which to send the reminders.
11. In the Timeout region:
    a. Select the No Timeout option.

       OR
    b. Select the Timeout after the following period option.

       In the Days field, enter the number of days.

This indicates the number of days after which the lead is timed out.

12. Click **Create**.

> **Note:** For a complete list of supported conditions, see Section 2.4.1.5, "Monitoring Engine Conditions".

2.4.1.5 Monitoring Engine Conditions

Table 2–25 lists the monitor conditions with their descriptions.

Table 2–25 Monitor Engine Conditions

Condition	Description
From State	The state of the lead (whether it was created or assigned) when it first qualifies for a monitor condition. The values supported for the From State are Creation and Assignment.
To State	The state of the lead till when the monitor conditions are applicable. The states supported are Accepted, In Progress, Closed, and Updated.
Time Lag Limit	Number of days after the From State changes when the first notification must be sent.
Relative to Expiration Date check box	Three possibilities exist: ■ If selected, and the expiration date is set for the lead, a notification is sent *n* days before the expiration date of the lead, where *n* = Time Lag Limit. ■ If selected, and the expiration date is not set for the lead, the monitor condition will never be satisfied by the lead. ■ If not selected, the first notification is sent *n* days after the lead creation date or the lead assignment date, where *n* = Time Lag Limit.
Total Reminders	Total number of reminders to be sent after the first notification.
Frequency	The number of days between two reminders, and between the notification and the first reminder.
Timeout Number	The number of days after the first notification when the lead must be timed out if it still satisfies the monitor condition.

2.4.1.6 Viewing the Monitor Log

The monitor log keeps track of all the leads that were processed by a monitor. For each monitor, you can view details such as the monitor triggered date, the notification and reminder recipient(s), and the status of a lead that triggered the monitor.

Prerequisites

None

Navigation

HTML - Administration > Leads > Setup > Monitoring Rules

Steps

1. From the Reports column, click the Reports icon for the monitor whose log you wish to view.

   The Monitor Log page appears.

2. Select the View From and To Dates.

3. Click **Run Report**.

   The summary appears in the Results region.

2.4.1.7 Searching for a Monitor

You can search for a monitor using the Search feature. You can search on the following attributes of the monitor - name, owner, status, rule set valid date, customer category, and country.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Processing Rules > Monitoring

Steps

1. In the Name field, enter the name of the monitor.
2. In the Owner field, enter the monitor owner name.
3. Open the Status drop-down list, and select the status of the monitor.
4. In the Start Date and End Date fields, select the dates between which the monitor is valid.
5. Open the Lead Rating drop-down list, and select the rating of the lead for the monitor that you are searching for.
6. Open the Country drop-down list, and select the country.
7. Click **Search**.

   The results are displayed based on your search criteria.
8. Optionally, click **Clear** to clear the values in all the Search fields.

2.4.2 Administrative Operational Reports

The Administration Leads module comprises four seeded reports for monitoring leads operations:

- Section 2.4.2.1, "Imports Records Manager Reports"
- Section 2.4.2.2, "Lead Processing History Reports"
- Section 2.4.2.3, "Rule Performance Reports"
- Section 2.4.2.4, "Rule Diagnostic Manager Reports"

2.4.2.1 Imports Records Manager Reports

The Import Records Manager report enables you to search all records imported as leads, to find import exceptions and recover from errors, and to track the number of leads imported by different sources.

To resolve errors in imported leads, see Section 2.1.2.3, "Checking for and Correcting Errors in Imported Leads".

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Operational Reports > Import Records Manager

Steps

To view details of an import, enter details in the following fields:

1. Open the Source System drop-down list, and select the source application from which the leads were generated.

   This is a mandatory field.

2. In the Source Code field, enter the product that the lead is interested in.
3. In the Import Interface ID field, enter the import interface ID that was assigned to the lead during import.
4. In the Batch ID field, enter the batch ID of the imported leads.

   You may have assigned a batch number to identify similar leads during lead import.

5. Open the Load Type drop-down list, and select the type of load used during the import.
6. Open the Load Status drop-down list, and select the status of the import operation.

   This is a mandatory field.

7. In the Load Date field, enter the date when the import operation was performed.

   Alternately, use the calendar tool to select a date.

8. In the Process Date field, enter the date when the imported records were processed.

   Alternately, use the calendar tool to select a date.

9. Open the Status Code drop-down list, and select the status code of the lead after it was processed.
10. In the Lead Description field, enter the description of the lead.

11. In the Customer field, enter the name of the customer organization to whom the leads belong.
12. In the First Name field, enter the first name of the contact representing the customer organization.
13. In the Last Name field, enter the last name of the contact representing the customer organization.
14. Click **Search**.

    The report appears below.
15. Optionally, click **Clear** to clear the entries in the fields.

2.4.2.2 Lead Processing History Reports

The Lead Processing History report provides the ability to view details of each execution of the Rules Engine per lead, and the state of the lead at different executions. Details such as the date and time, winning rule, and the outcome of each engine run for each lead is displayed.

The user can search by - Processing Stage (including all), Lead Name, Lead ID, Rule Set Name, Guard fields, Creation and Evaluation dates, and Result.

The report is grouped by unique Lead (ID and Name). All sorting is within a lead.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Operational Reports > Lead Processing History

Steps

To view details of a lead that was processed through the Rules Engine, enter details in the following fields.

1. Open the Processing Stage drop-down list, and select the processing stage of the leads - qualification, rating or channel selection.

2. In the Lead Number field, enter the number assigned to the lead to see its processing history.

   Leave this field blank if you do not wish to view the report for a specific lead.

3. In the Rule Set field, enter the name of the rule set if you wish to view the report based on a specific rule set.

4. In the Lead Name field, enter the name of the lead to see its processing history.

   Leave this field blank if you do not wish to view the report for a specific lead.

5. Open the Rule Result drop-down list, and select the rule result.

   The report displays processing history only for the selected rule result.

6. Open the Lead Status drop-down list, and select the status of the lead.

   The report displays processing history only for the selected status.

7. In the Lead Creation Date From field, enter the lead creation date from when the details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

8. In the Lead Creation Date To field, enter the lead creation date till when the details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

9. Open the Product Category drop-down list, and select a product category for which the leads were processed.

10. Open the Country drop-down list, and select the customer's country for whom the leads were processed.

11. In the Source Name field, enter the name of the campaign from which the leads were generated.

12. Click **Search**.

    The report appears below.

13. Optionally, click **Clear** to clear the entries in the fields.

2.4.2.3 Rule Performance Reports

The Rule Performance report provides the ability to view utilization and effectiveness of rule sets (for optimization of rule configuration). This stores the last engine run for each lead, the date and time, the winning rule, and the outcome.

Search by - Processing stage (including all), Lead Name and Lead ID, Rule Set Name, Guard fields, Creation and Evaluation dates, and Result.

Results grouped by - Stage, Rule Set, and Result, in that sequence.

Rollups:

- **Utilization** - Total number of leads processed (matching criteria)
- **Effectiveness**

  Total number and % of total of Status

  Total number and % total Upgraded or Downgraded

  Effectiveness - Total number and % of total Accepted.

For failed leads which use the default resource, the reports display the lead outcome (for example, grade) with no rule applicable. The administrator can drill down to view the winning sub-rule for each rule set and review the criteria.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Operational Reports > Rule Performance

Steps

To view details of a rule set to ascertain its performance, enter details in the following fields:

1. Open the Processing Stage drop-down list, and select the processing stage of the leads - qualification, rating or channel selection.
2. In the Lead Number field, enter the number assigned to the lead to see details when it was processed through a rule set.

   Leave this field blank if you do not wish to view the report for a specific lead.
3. In the Rule Set field, enter the name of the rule set if you wish to view the report based on a specific rule set.

4. In the Lead Name field, enter the name of the lead to see details when it was processed through a rule set.

   Leave this field blank if you do not wish to view the report for a specific lead.

5. Open the Rule Result drop-down list, and select the rule result.

   The report displays details only for the selected rule result.

6. Open the Lead Status drop-down list, and select the status of the lead.

   The report displays details only for the selected status.

7. In the Lead Creation Date From field, enter the lead creation date from when the details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

8. In the Lead Creation Date To field, enter the lead creation date till when the details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

9. Open the Product Category drop-down list, and select a product category for which the leads were processed.

10. Open the Country drop-down list, and select the customer's country for whom the leads were processed.

11. In the Source Name field, enter the name of the campaign from which the leads were generated.

12. Click **Search**.

    The report appears below.

13. Optionally, click **Clear** to clear the entries in the fields.

2.4.2.4 Rule Diagnostic Manager Reports

The Rule Diagnostic Manager report provides the ability to troubleshoot and manage rule configuration in the Leads Engine. This stores details of each engine run for each lead, and details include failure, tied, and winning executions of the rules and their precedence and guards for setup analysis.

Search by - Processing stage (including all), Lead Name and Lead ID, Rule Set Name, Guard fields, Creation and evaluation dates, and Result.

Results grouped by - Stage, then Rule Set.

Rollups

- Total number of executions where leads matched
- Total number of executions and % Failed
- Total number of executions and % Passed
- Total number of executions and % Tied
- Total number of executions and % Won

Each report shows record-by-record details of the executions.

Prerequisites

None

Responsibility

System Administrator

Navigation

HTML - Administration > Leads > Operational Reports > Rule Diagnostics

Steps

To view diagnostics of the execution of a rule set, enter details in the following fields.

1. Open the Processing Stage drop-down list, and select the processing stage of the leads - qualification, rating or channel selection.
2. In the Lead Number field, enter the number assigned to the lead to see details when it was processed through a rule set.

   Leave this field blank if you do not wish to view the report for a specific lead.
3. In the Rule Set field, enter the name of the rule set if you wish to view the report based on a specific rule set.
4. In the Lead Name field, enter the name of the lead to see details when it was processed through a rule set.

   Leave this field blank if you do not wish to view the report for a specific lead.
5. Open the Rule Result drop-down list, and select the rule result.

   The report will display details only for the selected rule result.
6. Open the Lead Status drop-down list, and select the status of the lead.

The report displays details only for the selected status.

7. In the Evaluation From Date field, enter the date from when the rule set evaluation details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

8. In the Evaluation To Date field, enter the date till when the rule set evaluation details must be displayed.

   Alternately, use the calendar tool to select a date. This is a mandatory field.

9. In the Lead Creation Date From field, enter the lead creation date from when the details must be displayed.

   Alternately, use the calendar tool to select a date.

10. In the Lead Creation Date To field, enter the lead creation date till when the details must be displayed.

    Alternately, use the calendar tool to select a date. This is a mandatory field.

11. Open the Product Category drop-down list, and select a product category for which the leads were processed.

12. Open the Country drop-down list, and select the customer's country for whom the leads were processed.

13. In the Source Name field, enter the name of the campaign from which the leads were generated.

14. Click **Search**.

    The report appears below.

15. Optionally, click **Clear** to clear the entries in the fields.

Part III

Appendixes

This section contains the following appendixes:

- Appendix A, "Oracle Leads Management Profile Options"
- Appendix B, "Seed Data"
- Appendix C, "Concurrent Programs"
- Appendix D, "Oracle Leads Management API Reference"

A

Oracle Leads Management Profile Options

This appendix provides a table of all of the system profile options used by the Oracle Leads Management.

The system profile options used by Oracle Leads Management have the following prefixes:

- OS: (Oracle Sales)
- OSO: (Oracle Sales Online)
- OOM: (Oracle Order Management)

Sections in this chapter include:

A.1 Setting Up Profiles

Use the following procedure to set up profiles.

Prerequisites

None

Responsibility

Forms - Oracle Sales Administrator

Navigation

Forms - Profiles > System

The following table details the system profiles to be set up for leads.

Here is an overview of the different columns of the table of profiles:

- **Req.?**: A Y in this column means that setting this profile is required. An N in this column means that setting this profile is optional.
- **New**: A Y in this column indicates this profile is new in release 11.5.9. An N means that it was introduced in prior releases.
- **Profile Option**: User name of the profile option.
- **Level**: Level at which this profile option can be set. A = Application, S = Site, R = Responsibility, U = User.
- **User**: This column indicates whether the user can view and modify the profile. Values are: V (Visible) = the user can see the profile only, U (Updatable) = users themselves can update the profile.
- **Program:** Lists the program that uses the profile.
- **Default Value**: Gives the seeded default.
- **Notes**: Usage notes.

A.2 Table of Oracle Leads Management Profiles

Table A–1 lists the profiles in Oracle Leads Management.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	Y	OS: Use DQM Rule code to match Party	A,S,R, U	U,V	DQM	None	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate.
Y	Y	OS: Use DQM Rule code to match Contact	A,S,R, U	U,V	DQM	None	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate.
Y	Y	OS: Use DQM Rule code to match Person	A,S,R, U	U,V	DQM	None	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	Y	OS: Run Lead Monitor Engine	S		Leads	Y	If set to `Y`, the lead is processed by the monitoring engine after it is created/updated.
Y	Y	OS:Maximum Reroutes per leads	S		Leads	3	The number assigned to this profile defines the maximum number of reroutes allowed to a lead that is timed out by the monitoring engine or during automatic assignment.
Y	Y	OS:Sales Lead Default Close Reason	S	VU	Leads	None	This profile is used by the Update Sales Lead API if the status is being set to that in the profile OS:Lead Link Status, and that in turn is a closed status. In that case, the API picks up this profile option value by default.
N	Y	OS:Lead New State Transition	S	VU	Leads	Y	If set to `Y`, new state transitioning logic controls the transition of lead state from one status to another, based on common business logic. Set to N if you do not want any additional restrictions.
Y	Y	OS:Auto Run Lead Engines While Update	S	V	Leads	Y	If set to `Y`, the lead engines are run automatically while updating a lead. If set to `N`, then the lead is processed manually. However, click **Run Engines** to process the lead via the lead engines.
N	Y	OS: Auto Convert Lead to Opportunity	S	V	Leads	Yes	If a channel is defined as INDIRECT in Channel Setup page, and this profile is set to 'Yes', an opportunity is created for the lead, and the partner matching workflow is started.
Y	N	OM: Item Validation Organization	SR	V	V	None	Manufacturing organization items are validated. This profile is used to set AS_SALES_LEAD_LINES.organization_id

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	Y	OS: Address Required for Person	S	None	V	'N	If set to 'Y', the Lead Contact page in Oracle Sales Online displays the Create Person button. The person or contact has to be created by clicking this button and entering address information on the Create Person page. If set to 'N', the user can enter contact information in empty rows.
N	Y	OS: Address Required for Sales Lead	S	None	V, U	No	Makes address entry mandatory for leads. If not set, then address is required. The API gives users an error if the address is required but not present in the record.
N	Y	OS: Assign New Lead	S	None	V	No	if this profile is set to No, then the application uses the Territory Manager to automatically assign resources to the lead. The first person the Territory Manager assigns becomes the owner. The rest of the resources in the territory become sales team members on the lead. if this profile is set to Yes, then the agent must enter the owner manually using the Owner drop-down list. If the agent does not make an entry, then the lead is assigned to the default user set in OS: Default Resource ID Used for Sales Leads. If no default resource is set, then application assigns ownership to the user updating or importing the lead. **Note**: If both the resource in this profile and the user who created or updated the lead do not have a valid sales role assigned to them, then the leads you import will not be accessible from either Oracle Sales Online or Oracle TeleSales.
N	Y	OS: Lead View scorecard data	A,S,R, U	U,V		N	Set to `Y` to view the old Score Card Mapping and rank data.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	N	OS: Auto assign from lead import (Obsolete)	-	-	-	-	-
N	N	OS: Auto ranking from lead import (Obsolete)	-	-	-	-	-
N	Y	OS: Auto Qualify Lead (Obsolete)	S	V	V	Yes	Turns automatic qualification for sales leads on or off. A setting of Yes, causes the application to attempt to qualify a lead when the Qualified check box is null. A setting of No means the user must qualify the lead manually by selecting the Qualified check box. If this profile is not set, it is defaulted to 'N'.
N	N	OS: Budget status required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a budget status to qualify the lead. If the profile option value is No, then lead qualification does not depend budget status entry.
N	N	OS: Calendar setting required for assignment	S	V	V	N	Calendar setting required for sales lead assignment. If set to `Yes`, then the routing APIs check availability of resource in JTF Calendar before assigning the lead owner.
N	N	OS: Campaign code required (Obsolete)	S	None	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a campaign code to qualify the lead. If the profile option value is No, then lead qualification does not depend on the campaign code.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	N	OS: Contact phone required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a contact phone to qualify the lead. If the profile option value is No, then lead qualification does not depend on the contact phone.
N	N	OS: Contact role required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a contact role to qualify the lead. If the profile option value is No, then lead qualification does not depend on the contact role.
Y	N	OS: Customer Access Privilege	S,R,U	None	V,U	Full	Security management for leads. See the section on security for details.
NOT USED	Y	OS: Dead Lead Status	S	None	V	Dead Lead	Not Used
N	N	OS: Decision time frame required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a decision time frame to qualify the lead. If the profile option value is No, then lead qualification does not depend on the decision time frame.
N	Y	OS: Default Budget Status for Leads	S, R , U	V, U	V, U	Pending	Defaults value in UI and API. If not set, this profile inserts a null value in the database table.
Y	Y	OS: Default Channel for Leads (Obsolete)	S, R , U	V, U	V, U	Direct	Used for defaulting value in UI and also in API. If not set, then the application inserts a null value and notifies the user of an error.
	11.5.8	OS: Default Channel for Lead Channel Selection Engine	S			Direct	This sets the default for each lead if no channel selection rules evaluate to true.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	Y	OS: Default Decision Timeframe for Leads	S, R , U	V, U	V, U	Within 1 week	Defaults value in UI and API. If not set, this profile inserts a null value in the database table.
N	Y	OS: Default Lead Contact Role	SRUA	VU	V	'END_ USER	Used for defaulting the contact role on the contact page.
Y	N	OS: Default Lead Marketing Owner	S	V,U		None	The immature lead owner assignment API will use the default marketing lead owner if owner cannot be found in the owner table.
n/a	n/a	OS: Default Lead Scorecard (obsolete)	SRA	VU	V	1	Scorecard to be used for lead ranking. If not set, the scoring engine will not function. While creating leads, the UI retrieves this value and passes it to the scoring APIs.
	11.5.8	OS: Default Rank for Lead Rating Engine	S	-	-	Cold Lead	This sets the default if no ranking rules evaluate to true.
N	N	OS: Default Resource ID Used for Sales Lead Assignment	S	None	V	None	The territory assignment program assigns ownership of a lead to this Resource ID if the program does not find any matching territory for the lead. If this profile is not set, then the ownership of any unassigned leads are automatically assigned to the user creating or importing the lead. This user must have a valid sales role assigned using Resource Manager. If the user does not have a valid sales role, then unassigned leads will not be accessible from Oracle Sales Online or Oracle TeleSales.
Y	Y	OS: Default Status for Leads	S, R , U	V, U	V, U	New	Used for defaulting value in UI and also in API. If not set, then the application inserts a null value and notifies the user of an error.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	Y	OS: Default Vehicle Response Code for Leads	S, R , U	V, U	V, U	User	Defaults value in UI and API. If not set, this profile inserts a null value in the database table.
		OS: Lead Incubation Channel	-	-	-	-	Immature lead owner assignment will be used if the sales channel in the profile value matches the channel code returned by the Channel Selection Engine.
Y	Y	OS: Lead Link Status	V	None	V	Converted to Opportunity	Used to set the status of the lead after linking the lead to an opportunity.
Y	Y	OS: Lead Routing Status (Obsolete)	S	None	V	New	When the routing engine finds a sales lead owner, then the sales lead status is reset to this profile
Y	Y	OS: Lead to Opportunity Move Sales Team	S		V, U	No	When converting a lead to an opportunity, you can govern whether all the sales team members are copied to the opportunity sales team by setting this profile. If set to Yes, all sales team members are copied to the new opportunity. If set to No, the sales team is limited to the sales representative who converted the lead, plus those sales reps who are included per the territory manager settings. The Keep flag is copied as is, and the Owner flag is copied as No.
Y	N	OS: Manager Update Access	S,R,U	None	V,U	View data	Security management for leads. See the section on security for details.
Y	N	OS: Opportunity Access Privilege	S,R,U	None	V,U	Global Full	Security management for leads. See the section on security for details.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	Y	OS: Privilege to Change Lead Owner	S, R, U	None	V,U	No	A user with this privilege can change the owner of a lead to which he/she has update access. Users who do not have this privilege can change owner of only those leads that they own. This profile is used to allow the user to change lead owner even if the user is not the current owner for lead. If not set, the application default this profile to No.
N	N	OS: Project name required (Obsolete)	S	V, U	V,U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a project name to qualify the lead. If the profile option value is No, then lead qualification does not depend project name.
N	Y	OS: Rank Lead (Obsolete)	S	None	V	System	The Rating Engine is run when this is set to SYSTEM, if the Rank ID is null.
	11.5.8	OS: Run New Lead Qualification, Rating, Channel Selection Engines (Obsolete)	S	-	-	Y	If set to `Y`, the lead is processed by the Leads Rules Engine. If set to N, qualification and ranking engines qualify and rank the lead based on profiles.
Y	N	OS: Sales Admin Update Access	S,R,U	None	V,U	View data	Security management for leads. See the section on security for details.
N	N	OS: Sales channel required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a sales channel to qualify the lead. If the profile option value is No, then lead qualification does not depend on sales channel entry.
Y	N	OS: Sales Lead Access Privilege	S, R , U	None	V,U	Full	<Access APIs>

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
N	Y	OS: Source Code Mandatory for Leads	S	V	V	No	Used to make source code (campaign) entry mandatory for leads. A Yes setting causes an error to appear if a user tries to save a lead without a source code. If this profile is not set, the application treats this as a Yes setting.
N	N	OS: Total budget amount required (Obsolete)	S	V, U	V, U	Yes	Used by auto qualification process in leads API. If value is Yes, then the user must enter a total budget amount to qualify the lead. If the profile option value is No, then lead qualification does not depend on the total budget amount.
Y	N	OSO: Default Country	SRAU	VU	VU	United States	Used for defaulting the country selected on the create lead page. If not set, the first country in the drop-down list will be chosen by default.
N	N	OSO: Default Person Relationship	SRU	VU	V	'EMPLOYEE_OF	The default value for contact relationship type.
N	N	OS: Customer Address Required for Customer, Opportunity and Lead	S	None	None	No	Indicates whether customer address is required for a customer.
Y	N	OS: Create Organization Privilege	S,R,U	None	None	Yes	If set to Yes, the Create Organization buttons are displayed on the Create page
Y	N	OS: Create Person Privilege	S,R,U	None	None	Yes	If set to Y, the Create Person buttons are displayed on the Create page.
Y	N	OS: Auto-relate Lead note to Customer	S,A,R,U	V,U	None	Yes	While creating a note, this profile is used to automatically relate the note to the customer.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y		OS: Auto-relate Lead note to Primary Contact	S,A,R, U	V,U	None	No	While creating a note, this profile is used to automatically relate the note to the primary contact.
Y	N	OS: Auto-relate Lead task to Primary Contact	S,A,R, U	V,U	None	No	While creating a task, this profile option is used to automatically relate the task to the primary contact.
Y	N	OS: Auto-relate Lead task to Customer	S,A,R, U	V,U	None	Yes	While creating a task, this profile option is used to automatically relate the task to the customer.
Y	N	OS: Sales Team Creator Keep Flag	S,A,R, U	V	None	Yes	Used for defaulting the keep flag in the lead sales team page.
Y	N	OS: Default Lead Marketing Owner	S	V,U	None	None	The immature lead owner assignment API will use the default marketing lead owner if owner cannot be found in the owner table.
Y	N	OTS: Default Country	S,A,R, U	V,U	None	US	Used for Lead Center in Oracle TeleSales.
Y	N	OTS: Default Party Type	S,A,R, U	V,U	None	PARTY_ RELATI ONSHIP	Used to default the party type in lead center.
Y	N	OTS: Default Tab in Sales Lead Center	S,A,R, U	V,U	None	AST_ SLC_ PUR	Default tab in lead center.
Y	N	OTS: Default Universal Search Tab	S,A,R, U	V,U	None	QUICK SEARCH	Default universal search tab.
Y	N	OTS: Default Universal Search Type	S,A,R, U	V,U	None	PARTY_ RELATI ONSHIP	Default universal search type.

Table A–1 Lead Profiles

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	N	OTS: Interactions-Enable Automatic Start	S,A,R,U	V	None	N	Used to automatically start interactions.
Y	N	OTS: Show Campaign and Offer Names instead of Codes	S,A,R,U	V,U	None	N	If set yo Y, Campaign Name is shown in the UI instead of Source Name.
N	N	OTS: Default User Role	S,A,R,U	V,U	None	None	Used to decide what the user's default role must be when the user logs in. Used by Oracle TeleSales.
N	N	OTS:Default User Sales Group	S,A,R,U	V,U	None	None	Used to decide what the user's default sales group must be when the user logs in. Used by Oracle TeleSales.
N	N	OTS: Minimum Number of Characters for Lookup	S,A,R,U	V	None	None	Min characters for long LOV lookup.

Table A–2 lists profiles related to leads that enable calculations in multiple currencies.

Table A–2 Lead Profiles Options for Multiple Currencies

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	N	OS: Forecast Calendar	S	None	V	None	The name of the calendar you are using to manage your forecasts. Used to get period set name used in as_period_rates table.
Y	N	OS: Daily Conversion Type	V	V,U	V,U	Corporate	Daily conversion type for currency conversion. Used in the amount calculation.
Y	N	OS: Date Mapping Type	S	VU	V	S	Used to get pseudo period rate coming from start or end of the period.

Table A–2 Lead Profiles Options for Multiple Currencies

Req.?	New	Profile Option	Level	User	Program	Default Value	Notes
Y	N	OS: Maximum Roll Days for Converting Amount	S	V	V	None	Maximum Roll Days for Converting Amount if No Conversion Rate exists for a Day. Used to get maximum rollup days for currency conversion
Y	Y	OS: Default Period Type for Currency Conversion	S	VU	V	None	Default period type for currency conversion. Used to get default period type for currency conversion.

B

Seed Data

This appendix chapter details seeded data in Oracle Leads Management under the following headings:

- Section B.1, "Seeded Data: Timeframe"
- Section B.2, "Seeded Data: Lead Rank"
- Section B.3, "Seeded Data: Lead Status"
- Section B.4, "Seed Data: Lead Attribute"

B.1 Seeded Data: Timeframe

Target Table/View: AML_SALES_LEAD_TIMEFRAMES

Table B–1 gives the seeded values for Timeframe.

Table B–1 Timeframe seeded values

Days	Timeframe Code
7	WITHIN 1 WEEK
30	WITHIN 1 MONTH
90	1 - 3 MONTHS
180	3 - 6 MONTHS
365	6 - 12 MONTHS
3650	MORE THAN 1 YEAR

B.2 Seeded Data: Lead Rank

Target Table/View: AS_SALES_LEAD_RANKS_VL.

Table B–2 gives the seeded values for Lead Rank.

Table B–2 Lead Rank seeded values

Description	Min Score	Max Score
Cold Lead	1	25
Low Lead	26	50
Medium Lead	51	75
Hot Lead	76	99

B.3 Seeded Data: Lead Status

Target Table/View: AS_STATUSES_VL.

Table B–3 gives the seeded values for Lead status.

Table B–3 Lead Status seeded values

Status Code	Open Status	Forecast Rollup	Win Loss	Usage Indicator	Meaning	Description
NEW	Y	N	N	ALL	New	New
LOSS	N	N	L	ALL	Loss	Loss
DEAD_LEAD	N	N	N	ALL	Dead Lead	Dead Lead
CONVERTED_TO_ OPPORTUNITY	N	N	N	-	Converted to Opportunity	Lead Converted to Opportunity
IN_PROGRESS	Y	Y	Y	ALL	In Progress	Progress

B.4 Seed Data: Lead Attribute

Target Table/View: PV_ATTRIBUTES_VL, PV_ENTITY_ATTRS.

Table B–4 gives the seeded values for Lead Attribute.

Table B–4 Lead Attribute seeded values

Attribute ID	Name	Description
4	Country	Country
9	Additional Channel Offering	Additional Channel Offering
16	Campaign	Campaign
20	State	State
500	Customer Name	Customer Name
501	Customer Address	Customer Address
502	Customer Annual Revenue	Customer Annual Revenue
503	Primary Contact	Primary Contact
504	Contact Role	Contact Role
505	Purchase Timeframe	Purchase Timeframe
506	Budget Status	Budget Status
507	Lead Score	Lead Score Description
508	Lead Status	Lead Status
509	Total Budget	Total Budget
510	Product Interest	Product Interest
511	Purchase Quantity - Product	Purchase Quantity - Product
512	Purchase Amount - Product	Purchase Amount - Product
513	Response Channel	Response Channel
514	Project	Project
517	Qualify Flag	Qualify Flag
518	Lead Rating	Lead Rating
519	Sales Channel	Sales Channel
520	Creation Date	Creation Date
522	Total Purchase Amount - Product	Total Purchase Amount - Product
523	Total Purchase Amount - Solutions	Total Purchase Amount - Solutions
524	Purchase Quantity - Solutions	Purchase Quantity - Solutions

Table B–4 Lead Attribute seeded values

Attribute ID	Name	Description
525	Purchase Amount - Solutions	Purchase Amount - Solutions
531	Primary Contact Phone	Primary Contact Phone
532	Area Code	Area Code
533	County	County
534	Province	Province
535	City	City
536	Postal Code	Postal Code
537	Primary Contact E-mail Address	Primary Contact E-mail Address
538	Primary Contact E-mail or Phone	Primary Contact E-mail or Phone
606	Customer/Account Type	Customer/Account Type
608	Customer Category	Customer Category

C

Concurrent Programs

This appendix provides details of all of the concurrent programs used by the Oracle Leads Management.

The system profile options used by Oracle Leads Management have the following prefixes:

- OS: (Oracle Sales)
- OSO: (Oracle Sales Online)
- OOM: (Oracle Order Management)

Sections in this chapter include:

C.1 Running Concurrent Programs

The procedure for running concurrent programs is the same for all Oracle applications.

Prerequisites

None

Login

Log in to Oracle Forms

Responsibility

Oracle Sales Administrator

Navigation

Concurrent Requests

Steps

1. Click **Run**.
2. Select the Single Request radio button.
3. Click **OK**.
4. Search for the concurrent request that you want to run.
5. Click **Submit**.
6. Select from the search results.
7. Click **OK**.
8. Click **OK** to run the concurrent request.

For a detailed description of the procedures, see *Oracle Applications System Administrator's Guide*.

C.2 Table of Concurrent Programs

The following table lists by name the concurrent programs used by Oracle Leads Management. The table includes the following columns from left to right:

- **Mandatory:** Is the program mandatory for the function?
- **Concurrent Program Name:** Name of the concurrent program.
- **Description:** Explains what the concurrent program does.

Mandatory	Concurrent Program Name	Description
No	Assign Territory Accesses	This concurrent program assigns new territory access to sales force employees. The program prepares database s for parallel processing. It requires setting three profile options: OS: Territory Minimum Number of Records for Parallel Processing OS: Territory Number of Child Processes OS: Territory Records to Open for Processing Changed Accounts Run this program after completing the setups in Setting Up Territory Management, and after the JTF Concurrent program Generate Territory Package has run. Parameters: Run Mode (New/Restart/Total) Lead Status: (All/Open/Closed) Previous Request ID for restart mode only.
No	Autocreate Opportunity from Sales lead	Creates opportunities from existing sales leads. This program should be run after the Import Sales Leads concurrent program. Parameters: Debug Mode Trace Mode
No	Generate Access Records	This is a child program of Assign Territory Accesses and does not need to be run separately.
No	Generate Territory Packages	This concurrent program, available by logging in under the CRM Administration responsibility, builds the API that returns the winning territories which are defined in territory setup. It must be run at least once before you import leads and each time after you modify the territory setup. You need not run this program every time you import leads.
Yes	Initial Build of Opportunity and Lead Bins	Used for the Opportunity and New Leads home page bins and reports. This program must be run initially before users can set up opportunity and new leads bins and reports. This program loads the opportunity and leads materialized view with sales credit information. A new parameter: Next Extent Size has been added. You can choose the value of the next extent to be allocated for all s and indexes created. Possible values for the parameter are: Small (1M), Medium (5M), Large (10M) **Note:** This program must be run anytime the as_sales_credit_denorm is rebuilt or when values such as sales stages and statuses have changed. You should also run when Refresh AS_PERIOD_DAYS is run.

Mandatory	Concurrent Program Name	Description
Yes	Initial Load for Lead Reports	This program is used to build Leads bins and reports. This should also be run following Refresh of Leads Bins Data so that you can see the latest values in leads reports.
No	Load Interest Types and Codes to Inventory Categories (Load Categories)	This concurrent program creates inventory categories under the inventory category set Oracle Sales and Marketing for each combination of interest types and codes. In order to use this concurrent program, the OS: Inventory Category Integration profile value must be set to Yes.
No	OTS: Load Sales Lead Interface from Flat File	Use this program to import sales leads from the interface. This program must be run before Auto Create Opportunity from Sales Lead. This moves data from the interface to AS_SALES_LEADS, AS_SALES_LEADS_LINES, and AS_SALES_LEAD_CONTACTS. If you want to import data from a flat file, you must first run OTS: Load Sales Lead, which will move data from the flat file to the interface .
No	Refresh of Leads Bin Data	This program is used to run an incremental refresh of the materialized view used for the Leads bins and reports. This program should be scheduled to run periodically.
No	Refresh of Opportunity Bins Data	This program is used to run an incremental refresh of the materialized view used for the opportunity bins and reports. This program should be scheduled to run periodically. **Note:** Users will not see opportunities in bins created after the last refresh of Refresh Sales Credit.
No	Setup Checking for Oracle Sales application	This program validates Sales Setups and produces an error log that the system administrator can use to diagnose invalid setups. Parameters: Upgrade - Yes/No
Y	Import Sales Leads	The lead import concurrent program allows you to import leads into Oracle Sales from other systems. While importing leads, the program also imports data on customers, addresses, and contacts into the customer model (TCA) tables.
Y	Workflow Background Process	This program sends notifications and reminders from triggered monitors. The system administrator must schedule it to run everyday, or twice a day, if required.
Y	Synchronization program	The synchronization concurrent program must be run after every lead import activity. This updates the DQM staging schema with new entries that were created during the lead import.
No	DQM Compile All Rules	This program must be run when any rules are modified in DQM.

Mandatory	Concurrent Program Name	Description
No	DQM Staging Program	The DQM Staging Program must be run when DQM is set up for the first time. It creates the staging schema this is vital for existence checking.

D

Oracle Leads Management API Reference

The public APIs provided by Oracle Leads Management are described and grouped according to procedures using them.

Sections in this chapter include:

D.1 Oracle Leads Management Procedures

Table D–1 lists the procedures which make up the Leads Public APIs.

Table D–1 Leads Public API Procedures

Procedure Name	Description
Create Sales Lead	Creates a new sales lead with the specified parameters. A unique sales lead ID will be created.
Update Sales Lead	Updates the sales lead record. The sales lead record being updated must refer to a valid sales lead ID in the database and must have a valid last_update_date (who column) passed in. This is used to check if the record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead record type, then these values will not be updated.
Create Sales Lead Lines	Creates one or more sales lead lines with the specified parameters. A table of sales lead line records with the parameters set needs to be passed in. A unique sales lead line ID will be created for each sales lead line. All the sales lead lines created will be associated with the sales lead ID which is passed in the sales lead line record.
Update Sales Lead Lines	Updates one or more sales lead lines. A table of sales lead line records with the parameters needs to be passed. Each sales lead line record must refer to a valid sales lead line ID in the database and must have a valid last_update_date (who column passed in). This is used to check if the sales lead line record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead line record type, then these will not be updated.
Delete Sales Lead Lines	Deletes one or more sales lead lines. A table of sales lead line records needs to be passed in . Each sales lead line record must refer to a valid sales lead line ID in the database.
Get Sales Lead Score	Gets the score and rank of the Sales Lead based on the Score card.
Run Lead Engines	Runs the qualification engine, rating engine, and channel selection engine.
Build Lead Sales Team	Builds lead sales team based on territory definition and adds lead creator as one of lead sales team members.
Rebuild Lead Sales Team	Rebuilds lead sales team to reflect the latest lead information.
Start Partner Matching	Start a workflow to do partner matching.

Table D–1 Leads Public API Procedures

Procedure Name	Description
Create Sales Lead Contacts	Creates one or more new sales lead contacts with the specified parameters. A table of sales lead contact records with the parameters set needs to be passed in. A unique lead contact ID will be created for each sales lead contact. All the sales lead contacts created will be associated with the sales lead ID which is passed in the sales lead contact record.
Update Sales Lead Contacts	Updates one or more sales lead contacts. A table of sales lead contacts records with the parameters needs to be passed in. Each sales lead contact record must refer to a valid sales lead contact ID in the database and must have a valid last_update_ date (who column passed in). This is used to check if the sales lead contact record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead contact record type, then these will not be updated.
Delete Sales Lead Contacts	Deletes one or more sales lead contacts. A table of sales lead contact records needs to be passed in . Each sales lead contact record must refer to a valid lead contact ID in the database.

D.2 Parameter Specifications

The specifications for the public APIs provided by the Oracle CRM Application Foundation 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.

D.2.1 Standard IN Parameters

Table D–2 describes standard IN parameters which are common to all APIs provided by Oracle Leads Management.

Table D–2 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.
p_init_msg_list	VARCHAR2	Yes	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false 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.
p_commit	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, the API commits before returning to the calling program. If set to false, then it is the calling program's responsibility to commit the transaction.
p_validation_ level	NUMBER	No	Level of validation required. If set to NONE, no validation will be done in the API. If set to FULL, all validations (item level and record level) will be performed.
P_check_access_ flag	VARCHAR2	No	If set to Y, access security check is performed. If set to N, access security check is not performed.
P_Admin_Flag	VARCHAR2	No	If set to Y, the current user has administrator privileges. If set to N, the current user does not have administrator privileges.
P_Admin_ Group_Id	NUMBER	No	If the current user has administrator privileges, the user’s SalesgroupID.

Table D–2 Standard IN Parameters

Parameter	Data Type	Required	Description
P_Identity_ salesforce_Id	NUMBER	No	Resource ID of the current user.

D.2.2 Standard OUT Parameters

Table D–3 describes standard OUT parameters, which are common to all public APIs provided by Oracle Leads Management.

Note: All standard OUT parameters are required.

Table D–3 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: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_return_status	VARCHAR2	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false 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.

Table D–3 Standard OUT Parameters

Parameter	Data Type	Description
x_msg_count	NUMBER	Holds the number of messages in the message list.If the error message returned is one, then the message count will be zero.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the number of messages is more than one, this parameter will be NULL and the messages must be extracted from the message stack.

D.3 Type Declarations

```
TYPE SALES_LEAD_Rec_Type IS RECORD
(
       SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
       LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
       LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
       CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
       CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
       LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
       REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
       LEAD_NUMBER                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
       STATUS_CODE                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
       CUSTOMER_ID                     NUMBER := FND_API.G_MISS_NUM,
       ADDRESS_ID                      NUMBER := FND_API.G_MISS_NUM,
       SOURCE_PROMOTION_ID             NUMBER := FND_API.G_MISS_NUM,
       INITIATING_CONTACT_ID           NUMBER := FND_API.G_MISS_NUM,
       ORIG_SYSTEM_REFERENCE           VARCHAR2(240) := FND_API.G_MISS_CHAR,
       CONTACT_ROLE_CODE               VARCHAR2(30) := FND_API.G_MISS_CHAR,
       CHANNEL_CODE                    VARCHAR2(30) := FND_API.G_MISS_CHAR,
       BUDGET_AMOUNT                   NUMBER := FND_API.G_MISS_NUM,
       CURRENCY_CODE                   VARCHAR2(15) := FND_API.G_MISS_CHAR,
       DECISION_TIMEFRAME_CODE         VARCHAR2(30) := FND_API.G_MISS_CHAR,
       CLOSE_REASON                    VARCHAR2(30) := FND_API.G_MISS_CHAR,
       LEAD_RANK_ID                    NUMBER := FND_API.G_MISS_NUM,
       LEAD_RANK_CODE                  VARCHAR2(30) := FND_API.G_MISS_CHAR,
       PARENT_PROJECT                  VARCHAR2(80) := FND_API.G_MISS_CHAR,
       DESCRIPTION                     VARCHAR2(2000) := FND_API.G_MISS_CHAR,
```

```
ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE11                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
ASSIGN_TO_PERSON_ID             NUMBER := FND_API.G_MISS_NUM,
ASSIGN_TO_SALESFORCE_ID         NUMBER := FND_API.G_MISS_NUM,
ASSIGN_SALES_GROUP_ID           NUMBER := FND_API.G_MISS_NUM,
ASSIGN_DATE                     DATE := FND_API.G_MISS_DATE,
BUDGET_STATUS_CODE              VARCHAR2(30) := FND_API.G_MISS_CHAR,
ACCEPT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
VEHICLE_RESPONSE_CODE           VARCHAR2(30) := FND_API.G_MISS_CHAR,
TOTAL_SCORE                     NUMBER := FND_API.G_MISS_NUM,
SCORECARD_ID                    NUMBER := FND_API.G_MISS_NUM,
KEEP_FLAG                       VARCHAR2(1) := FND_API.G_MISS_CHAR,
URGENT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
IMPORT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
REJECT_REASON_CODE              VARCHAR2(30) := FND_API.G_MISS_CHAR,
DELETED_FLAG                    VARCHAR2(1) := FND_API.G_MISS_CHAR,
OFFER_ID                        NUMBER := FND_API.G_MISS_NUM,
INCUMBENT_PARTNER_PARTY_ID      NUMBER := FND_API.G_MISS_NUM,
INCUMBENT_PARTNER_RESOURCE_ID   NUMBER := FND_API.G_MISS_NUM,
PRM_EXEC_SPONSOR_FLAG           VARCHAR2(1) := FND_API.G_MISS_CHAR,
PRM_PRJ_LEAD_IN_PLACE_FLAG      VARCHAR2(1) := FND_API.G_MISS_CHAR,
PRM_SALES_LEAD_TYPE             VARCHAR2(30) := FND_API.G_MISS_CHAR,
PRM_IND_CLASSIFICATION_CODE     VARCHAR2(30) := FND_API.G_MISS_CHAR,
QUALIFIED_FLAG                  VARCHAR2(1) := FND_API.G_MISS_CHAR,
ORIG_SYSTEM_CODE                VARCHAR2(30) := FND_API.G_MISS_CHAR,
PRM_ASSIGNMENT_TYPE             VARCHAR2(30) := FND_API.G_MISS_CHAR,
AUTO_ASSIGNMENT_TYPE            VARCHAR2(30) := FND_API.G_MISS_CHAR,
PRIMARY_CONTACT_PARTY_ID        NUMBER := FND_API.G_MISS_NUM,
PRIMARY_CNT_PERSON_PARTY_ID     NUMBER := FND_API.G_MISS_NUM,
PRIMARY_CONTACT_PHONE_ID        NUMBER := FND_API.G_MISS_NUM,
REFERRED_BY                     NUMBER := FND_API.G_MISS_NUM,
```

```
        REFERRAL_TYPE                   VARCHAR2(30) := FND_API.G_MISS_CHAR,
        REFERRAL_STATUS                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
        REF_DECLINE_REASON              VARCHAR2(30) := FND_API.G_MISS_CHAR,
        REF_COMM_LTR_STATUS             VARCHAR2(30) := FND_API.G_MISS_CHAR,
        REF_ORDER_NUMBER                NUMBER := FND_API.G_MISS_NUM,
        REF_ORDER_AMT                   NUMBER := FND_API.G_MISS_NUM,
        REF_COMM_AMT                    NUMBER := FND_API.G_MISS_NUM
        );
G_MISS_SALES_LEAD_REC          SALES_LEAD_Rec_Type;
TYPE  SALES_LEAD_Tbl_Type      IS TABLE OF SALES_LEAD_Rec_Type
                               INDEX BY BINARY_INTEGER;

G_MISS_SALES_LEAD_TBL          SALES_LEAD_Tbl_Type;

TYPE SALES_LEAD_LINE_Rec_Type IS RECORD
(
        SALES_LEAD_LINE_ID              NUMBER := FND_API.G_MISS_NUM,
        LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
        LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
        CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
        CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
        LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
        REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
        PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
        PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
        PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
        SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
        STATUS_CODE                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
        INTEREST_TYPE_ID                NUMBER := FND_API.G_MISS_NUM,
        PRIMARY_INTEREST_CODE_ID        NUMBER := FND_API.G_MISS_NUM,
        SECONDARY_INTEREST_CODE_ID      NUMBER := FND_API.G_MISS_NUM,
        INVENTORY_ITEM_ID               NUMBER := FND_API.G_MISS_NUM,
        ORGANIZATION_ID                 NUMBER := FND_API.G_MISS_NUM,
        UOM_CODE                        VARCHAR2(3) := FND_API.G_MISS_CHAR,
        QUANTITY                        NUMBER := FND_API.G_MISS_NUM,
        BUDGET_AMOUNT                   NUMBER := FND_API.G_MISS_NUM,
        SOURCE_PROMOTION_ID             NUMBER := FND_API.G_MISS_NUM,
        ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
        ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
```

```
       ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE11                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
       OFFER_ID                        NUMBER        := FND_API.G_MISS_NUM
       );

G_MISS_SALES_LEAD_LINE_REC          SALES_LEAD_LINE_Rec_Type;
TYPE  SALES_LEAD_LINE_Tbl_Type      IS TABLE OF SALES_LEAD_LINE_Rec_Type
                                    INDEX BY BINARY_INTEGER;

G_MISS_SALES_LEAD_LINE_TBL          SALES_LEAD_LINE_Tbl_Type;

TYPE SALES_LEAD_LINE_OUT_Rec_Type   IS RECORD
(
        SALES_LEAD_LINE_ID             NUMBER,
        RETURN_STATUS                  VARCHAR2(1)
);

TYPE SALES_LEAD_LINE_OUT_Tbl_Type  IS TABLE OF SALES_LEAD_LINE_OUT_Rec_Type
                                   INDEX BY BINARY_INTEGER;

TYPE SALES_LEAD_CONTACT_Rec_Type IS RECORD
(
       LEAD_CONTACT_ID                 NUMBER := FND_API.G_MISS_NUM,
       SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
       CONTACT_ID                      NUMBER := FND_API.G_MISS_NUM,
       LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
       LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
       CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
       CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
       LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
       REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
       PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
       ENABLED_FLAG                    VARCHAR2(1) := FND_API.G_MISS_CHAR,
       RANK                            VARCHAR2(30) := FND_API.G_MISS_CHAR,
       CUSTOMER_ID                     NUMBER := FND_API.G_MISS_NUM,
       ADDRESS_ID                      NUMBER := FND_API.G_MISS_NUM,
       PHONE_ID                        NUMBER := FND_API.G_MISS_NUM,
```

```
        CONTACT_ROLE_CODE               VARCHAR2(30) := FND_API.G_MISS_CHAR,
        PRIMARY_CONTACT_FLAG            VARCHAR2(1) := FND_API.G_MISS_CHAR,
        ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
        ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE11                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
        CONTACT_PARTY_ID                NUMBER := FND_API.G_MISS_NUM
        );

G_MISS_SALES_LEAD_CONTACT_REC           SALES_LEAD_CONTACT_Rec_Type;

TYPE  SALES_LEAD_CONTACT_Tbl_Type       IS TABLE OF SALES_LEAD_CONTACT_Rec_Type
                                        INDEX BY BINARY_INTEGER;

G_MISS_SALES_LEAD_CONTACT_TBL           SALES_LEAD_CONTACT_Tbl_Type;
TYPE SALES_LEAD_CNT_OUT_Rec_Type   IS RECORD
(
        LEAD_CONTACT_ID                 NUMBER,
        RETURN_STATUS                   VARCHAR2(1)
);

TYPE SALES_LEAD_CNT_OUT_Tbl_Type   IS TABLE OF SALES_LEAD_CNT_OUT_Rec_Type
                                 INDEX BY BINARY_INTEGER;   camp_rec_type IS
TYPE LEAD_ENGINES_OUT_Rec_Type   IS RECORD
(
       qualified_flag                VARCHAR2(1),
       lead_rank_id                  NUMBER,
       channel_code                  VARCHAR2(30),
       indirect_channel_flag         VARCHAR2(1),
       sales_team_flag               VARCHAR2(1)
);
```

D.4 Create Sales Lead

The Create Sales procedure creates a sales lead with the parameters specified. A unique sales lead ID is created.

D.4.1 Procedure Specification

```
PROCEDURE Create_sales_lead(
    P_Api_Version_Number     IN   NUMBER,
    P_Init_Msg_List          IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                 IN   VARCHAR2     := FND_API.G_FALSE,
    P_Validation_Level       IN   NUMBER       := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag      IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag             IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id         IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id IN   NUMBER       := FND_API.G_MISS_NUM,
   P_Sales_Lead_Profile_Tbl IN  AS_UTILITY_PUB.Profile_Tbl_Type := ASUTILITY_
                                  PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_Rec         IN   SALES_LEAD_Rec_Type  := G_MISS_SALES_LEAD_REC,
    P_SALES_LEAD_LINE_tbl    IN   SALES_LEAD_LINE_tbl_type
                                  DEFAULT G_MISS_SALES_LEAD_LINE_tbl,
    P_SALES_LEAD_CONTACT_tbl IN   SALES_LEAD_CONTACT_tbl_type
                                  DEFAULT G_MISS_SALES_LEAD_CONTACT_tbl,
    X_SALES_LEAD_ID          OUT  NUMBER,
    X_SALES_LEAD_LINE_OUT_Tbl OUT SALES_LEAD_LINE_OUT_Tbl_Type,
    X_SALES_LEAD_CNT_OUT_Tbl OUT  SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status          OUT  VARCHAR2,
    X_Msg_Count              OUT  NUMBER,
    X_Msg_Data               OUT  VARCHAR2
    );
```

D.4.2 Current Version

2.0

D.4.3 Parameter Descriptions

Notes

- A unique sales lead ID is generated from the sequence.
- In P_Sales_Lead_Rec , the required parameters are status_code, customer_id and source_promotion_id (based on profile)

- If P_sales_lead_line_tbl and/or P_sales_lead_contact_tbl is passed in, then the appropriate create APIs for sales lead lines and sales lead contacts is also called after creating the sales lead header.

Table D–4 describes the IN parameters for the Create Sales Lead procedure.

Table D–4 IN Parameters

Parameter	Data Type	Required	Description
P_Sales_Lead_ Profile_Tbl	AS_UTILITY_ PUB.Profile_ Tbl_Type	No	Data type to store the access security related profile values (they can be cached mid-tier and passed to the API)
P_sales_lead_ rec	AS_SALES_ LEADS_ PUB.SALES_ LEAD_REC_ TYPE	No	Sales lead record
P_Sales_lead_ line_tbl	AS_SALES_ LEADS_ PUB.SALES_ LEAD_LINE_ TBL_TYPE	No	Table of sales lead line records
P_Sales_lead_ contact_tbl	AS_SALES_ LEADS_ PUB.SALES_ LEAD_ CONTACT_ TBL_TYPE	No	Table of sales lead contact records

Table D–5 describes the OUT parameters for the Create Sales Lead procedure.

Table D–5 OUT Parameters

Parameter	Data Type	Description
X_sales_lead_ID	NUMBER	Sales lead ID of the sales lead just created
X_sales_lead_line_ out_tbl	AS_SALES_ LEADS_ PUB.SALES_ LEAD_LINE_ OUT_TBL_TYPE	Table of sales lead line out record types. Each record type consists of the sales lead line ID created and the return status.

Table D–5 OUT Parameters

Parameter	Data Type	Description
X_sales_lead_ contact_out_tbl	AS_SALES_ LEADS_ PUB.SALES_ LEAD_ CONTACT_OUT_ TBL_TYPE	Table of sales lead contact out record types. Each record type consists of the lead contact ID created and the return status.

D.5 Update Sales Lead

This procedure updates a sales lead with the parameters specified. A valid sales lead ID must be passed in.

D.5.1 Procedure Specification

```
PROCEDURE Update_sales_lead(
    P_Api_Version_Number     IN   NUMBER,
    P_Init_Msg_List          IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                 IN   VARCHAR2     := FND_API.G_FALSE,
    P_Validation_Level       IN   NUMBER       := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag      IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag             IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id         IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id IN   NUMBER       := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl IN   AS_UTILITY_PUB.Profile_Tbl_Type := AS_UTILITY_
PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_Rec         IN   SALES_LEAD_Rec_Type
                                      DEFAULT G_MISS_SALES_LEAD_REC,
    X_Return_Status          OUT  VARCHAR2,
    X_Msg_Count              OUT  NUMBER,
    X_Msg_Data               OUT  VARCHAR2
    );
```

D.5.2 Current Version

2.0

D.5.3 Parameter Descriptions

Notes

- A valid sales lead ID must be passed in the sales lead record type.

- Last_update_date must be passed in.

Table D–6 describes the IN parameters for the Update Sales Lead procedure.

Table D–6 IN Parameters

Parameters	Data Type	Required	Description
P_Sales_Lead_Profile_Tbl	AS_UTILITY_PUB.Profile_Tbl_Type	No	Data type to store the access security related profile values (they can be cached in the mid tier and passed to the API)
P_sales_lead_rec	AS_SALES_LEADS_PUB.SALES_LEAD_REC_TYPE	No	Sales lead record type for update

Note: This procedure does not have OUT parameters.

D.6 Create Sales Lead Lines

This procedure creates one or more sales lead lines with the parameters specified and associates them with the sales lead ID passed in.

D.6.1 Procedure Specification

```
PROCEDURE Create_sales_lead_lines(
    P_Api_Version_Number      IN   NUMBER,
    P_Init_Msg_List           IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                  IN   VARCHAR2     := FND_API.G_FALSE,
    p_validation_level        IN   NUMBER       := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag       IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag              IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id          IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id  IN   NUMBER       := FND_API.G_MISS_NUM,
  P_Sales_Lead_Profile_Tbl   IN   AS_UTILITY_PUB.Profile_Tbl_Type
                                  := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl     IN   SALES_LEAD_LINE_Tbl_Type
                                   := G_MISS_SALES_LEAD_LINE_Tbl,
    P_SALES_LEAD_ID           IN   NUMBER,
    X_SALES_LEAD_LINE_OUT_Tbl OUT  SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status           OUT  VARCHAR2,
```

```
X_Msg_Count                    OUT  NUMBER,
X_Msg_Data                     OUT  VARCHAR2
);
```

D.6.2 Current Version

2.0

D.6.3 Parameter Descriptions

Notes

- A unique sales lead line ID is generated from the sequence for each sales lead line.
- In P_SALES_LEAD_LINE_REC, the required parameters are sales_lead_id, source_promotion_id and either product category (interest_type_id, primary_interest_code_id, secondary_interest_code_id) or inventory item (inventory_item_id and organization_id).

Table D–7 describes the IN parameters for the Create Sales Lead Lines procedure.

Table D–7 IN Parameters

Parameters	Data Type	Required	Description
P_Sales_Lead_Profile_Tbl	AS_UTILITY_PUB.Profile_Tbl_Type	No	Data type to store the access security related profile values (they can be cached in the mid tier and passed to the API)
P_sales_lead_ID	NUMBER	Yes	Sales lead ID for which the sales lead lines are being created.
P_Sales_lead_line_tbl	AS_SALES_LEADS_PUB.SALES_LEAD_LINE_TBL_TYPE	No	Table of sales lead line records

Table D–8 describes the OUT parameters for the Create Sales Lead Lines procedure.

Table D–8 OUT Parameters

Parameter	Data Type	Description
X_sales_lead_ line_out_tbl	AS_SALES_LEADS_ PUB.SALES_LEAD_ LINE_OUT_TBL_ TYPE	Table of sales lead line out record types. Each record type consists of the sales lead line ID created and the return status.

D.7 Update Sales Lead Lines

This procedure updates one or more sales lead lines with the parameters specified.

D.7.1 Procedure Specification

```
PROCEDURE Update_sales_lead_lines(
    P_Api_Version_Number         IN   NUMBER,
    P_Init_Msg_List              IN   VARCHAR2   := FND_API.G_FALSE,
    P_Commit                     IN   VARCHAR2   := FND_API.G_FALSE,
    p_validation_level           IN   NUMBER     := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN   VARCHAR2   := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN   VARCHAR2   := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN   NUMBER     := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN   NUMBER     := FND_API.G_MISS_NUM,
   P_Sales_Lead_Profile_Tbl    IN  AS_UTILITY_PUB.Profile_Tbl_Type := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl        IN   SALES_LEAD_LINE_Tbl_Type,
    X_SALES_LEAD_LINE_OUT_Tbl    OUT  SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status              OUT  VARCHAR2,
    X_Msg_Count                  OUT  NUMBER,
    X_Msg_Data                   OUT  VARCHAR2
    );
```

D.7.2 Current Version

2.0

D.7.3 Parameter Descriptions

Notes

- In P_SALES_LEAD_LINE_REC, the required parameters are sales_lead_id, source_promotion_id and either product category (interest_type_id, primary_

interest_code_id, secondary_interest_code_id) or inventory item (inventory_item_id and organization_id).

- last_update_date must be passed in for each sales lead line being updated.

Table D–9 describes the IN parameters for the Update Sales Lead Lines procedure.

Table D–9 IN Parameters

Parameter	Data Type	Required	Description
P_Sales_Lead_Profile_Tbl	AS_UTILITY_PUB.Profile_Tbl_Type	No	Data type to store the access security related profile values (they can be cached in the mid tier and passed to the API)
P_Sales_lead_line_tbl	AS_SALES_LEADS_PUB.SALES_LEAD_LINE_TBL_TYPE	No	Table of sales lead line records

Table D–10 describes the OUT parameters for the Update Sales Lead Lines procedure.

Table D–10 OUT Parameters

Parameter	Data Type	Description
X_sales_lead_line_out_tbl	AS_SALES_LEADS_PUB.SALES_LEAD_LINE_OUT_TBL_TYPE	Table of sales lead line out record types. Each record type consists of the sales lead line ID processed and the return status.

D.8 Delete Sales Lead Lines

This procedure deletes one or more sales lead lines.

D.8.1 Procedure Specification

```
PROCEDURE Delete_sales_lead_lines(
    P_Api_Version_Number         IN   NUMBER,
    P_Init_Msg_List              IN   VARCHAR2   := FND_API.G_FALSE,
    P_Commit                     IN   VARCHAR2   := FND_API.G_FALSE,
    p_validation_level           IN   NUMBER     := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN   VARCHAR2   := FND_API.G_MISS_CHAR,
```

```
    P_Admin_Flag                IN   VARCHAR2    := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id            IN   NUMBER      := FND_API.G_MISS_NUM,
    P_identity_salesforce_id    IN   NUMBER      := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl    IN   AS_UTILITY_PUB.Profile_Tbl_Type := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl       IN   SALES_LEAD_LINE_Tbl_type,
    X_SALES_LEAD_LINE_OUT_Tbl   OUT  SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status             OUT  VARCHAR2,
    X_Msg_Count                 OUT  NUMBER,
    X_Msg_Data                  OUT  VARCHAR2
    );
```

D.8.2 Current Version

2.0

D.8.3 Parameter Descriptions

Note: In P_SALES_LEAD_LINE_REC, the required parameters are sales_lead_line_id.

Table D–11 describes the IN parameters for the Delete Sales Lead Lines procedure.

Table D–11 IN Parameters

Parameter	Data Type	Required	Description
P_Sales_ Lead_Profile_ Tbl	AS_UTILITY_ PUB.Profile_ Tbl_Type	No	Data type to store the access security related profile values (they can be cached in the mid tier and passed to the API)
P_Sales_lead_ line_tbl	AS_SALES_ LEADS_ PUB.SALES_ LEAD_LINE_ TBL_TYPE	No	Table of sales lead line records

Table D–12 describes the OUT parameters for the Delete Sales Lead Lines procedure.

Table D–12 OUT Parameters

Parameter	Data Type	Description
X_sales_lead_ line_out_tbl	AS_SALES_LEADS_ PUB.SALES_LEAD_ LINE_OUT_TBL_ TYPE	Table of sales lead line out record types. Each record type consists of the sales lead line ID processed and the return status.

D.9 Get Sales Lead Score

This procedure calculates the score and rank of the Sales Lead based on the score card and returns them

D.9.1 Procedure Specification

```
Procedure Get_Score (
    p_api_version             IN  NUMBER := 2.0,
    p_init_msg_list           IN  VARCHAR2 := FND_API.G_FALSE,
    p_commit                  IN  VARCHAR2 := FND_API.G_FALSE,
    p_validation_level        IN  NUMBER := AS_UTILITY_PUB.G_VALID_LEVEL_ITEM,
    P_Check_Access_Flag       IN  VARCHAR2 := FND_API.G_MISS_CHAR,
    p_sales_lead_id           IN  NUMBER,
    p_scorecard_id            IN  NUMBER,
    p_marketing_score         IN  NUMBER := 0,
    p_identity_salesforce_id  IN  NUMBER,
    p_admin_flag              IN  VARCHAR2,
    p_admin_group_id          IN  NUMBER,
    x_rank_id                 OUT NUMBER,
    X_SCORE                   OUT NUMBER,
    x_return_status           OUT VARCHAR2,
    x_msg_count               OUT NUMBER,
    x_msg_data                OUT VARCHAR2 );
```

D.9.2 Current Version

2.0

D.9.3 Parameter Descriptions

Table D–13 describes the IN parameters for the Get Sales Lead Scores procedure.

Table D–13 IN Parameters

Parameter	Data Type	Required	Description
p_sales_ lead_id	NUMBER	Yes	The Unique Identifier of the Sales Lead
p_scorecard_ id	NUMBER	Yes	The Unique Identifier of the Score card
p_ marketing_ score	NUMBER	No	The additional score which could be used to alter the total score of the sales Lead in turn to change the rank of the sales lead.

Table D–14 describes the OUT parameters for the Get Sales Lead Scores procedure.

Table D–14 OUT Parameters

Parameter	Data Type	Description
X_rank_id	NUMBER	The rank ID of the Sales Lead
X_score	NUMBER	The total score of the Sales Lead

D.10 Run Lead Engines

This API should be called after lead is created, or Run Engine button is clicked. If user doesn't specify qualified flag, rank, or sales channel, and profile setting is to do them automatically, this API will run qualification engine, rating engine, and channel selection engine. Besides, if user sets profile to run old 11.5.7 engines, it will run 11.5.7 qualification and ranking engines.

D.10.1 Procedure Specification

```
PROCEDURE Run_Lead_Engines (
    P_Api_Version_Number      IN  NUMBER,
    P_Init_Msg_List           IN  VARCHAR2 := FND_API.G_FALSE,
    p_Commit                  IN  VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level        IN  NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id          IN  NUMBER   := FND_API.G_MISS_NUM,
    P_identity_salesforce_id  IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Salesgroup_id           IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id           IN  NUMBER,
    X_Lead_Engines_Out_Rec    OUT LEAD_ENGINES_OUT_Rec_Type,
    X_Return_Status           OUT VARCHAR2,
    X_Msg_Count               OUT NUMBER,
```

```
X_Msg_Data                  OUT VARCHAR2
);
```

D.10.2 Current Version

2.0

D.10.3 Parameter Descriptions

Notes

- Api_version_number will be set to 2.0.
- If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
- If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
- If a flag column is not passed in, default it to 'Y' or 'N'.

Table D–15 describes the IN parameters for the Run Lead Engines API.

Table D–15 IN Parameters

Parameter	Data Type	Required	Description
P_sales_lead_id	NUMBER	Yes	Sales Lead Identifier that user wants to build sales team for

Table D–16 describes the OUT parameters for the Run Lead Engines API.

Table D–16 OUT Parameters

Parameter	Data Type	Description
X_Lead_ Engines_Out_ Rec	LEAD_ ENGINES_ OUT_Rec_Type	Result of qualification, rating, and channel selection engines.

D.11 Build Lead Sales Team

This API should be called after Run_Lead_Engines API is called. It builds lead sales team based on territory definition and adds lead creator as one of lead sales team member.

D.11.1 Procedure Specification

```
PROCEDURE Build_Lead_Sales_Team(
    P_Api_Version_Number      IN  NUMBER,
    P_Init_Msg_List           IN  VARCHAR2 := FND_API.G_FALSE,
    p_Commit                  IN  VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level        IN  NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id          IN  NUMBER   := FND_API.G_MISS_NUM,
    P_identity_salesforce_id  IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Salesgroup_id           IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id           IN  NUMBER,
    X_Return_Status           OUT VARCHAR2,
    X_Msg_Count               OUT NUMBER,
    X_Msg_Data                OUT VARCHAR2
);
```

D.11.2 Current Version

2.0

D.11.3 Parameter Descriptions

Notes

- Api_version_number will be set to 2.0.
- If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
- If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
- If a flag column is not passed in, default it to 'Y' or 'N'.

Table D–17 describes the IN parameters for the Build Lead Sales Team API.

Table D–17 IN Parameters

Parameter	Data Type	Required	Description
P_sales_lead_id	NUMBER	Yes	Sales Lead Identifier that user wants to build sales team for

Note: This procedure does not have OUT parameters.

D.12 Rebuild Lead Sales Team

This API should be called after lead header is updated and lines are created/updated/deleted. When user does change to the lead, the lead may not match the territory it originally met, and match other territories. This API will rebuild lead sales team to reflect the latest lead information.

D.12.1 Procedure Specification

```
PROCEDURE Rebuild_Lead_Sales_Team(
    P_Api_Version_Number      IN  NUMBER,
    P_Init_Msg_List           IN  VARCHAR2 := FND_API.G_FALSE,
    p_Commit                  IN  VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level        IN  NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id          IN  NUMBER   := FND_API.G_MISS_NUM,
    P_identity_salesforce_id  IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Salesgroup_id           IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id           IN  NUMBER,
    X_Return_Status           OUT VARCHAR2,
    X_Msg_Count               OUT NUMBER,
    X_Msg_Data                OUT VARCHAR2
    );
```

D.12.2 Current Version

2.0

D.12.3 Parameter Descriptions

Notes

- Api_version_number will be set to 2.0.
- If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
- If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
- If a flag column is not passed in, default it to 'Y' or 'N'.

Table D–18 describes the IN parameters for the Rebuild Lead Sales Team API.

Table D–18 IN Parameters

Parameter	Data Type	Required	Description
P_sales_lead_ id	NUMBER	Yes	Sales Lead Identifier that user wants to build sales team for

Note: This procedure does not have OUT parameters.

D.13 Start Partner Matching

This API is a workflow to do partner matching. Since partner matching may take a while, it defers the process and exits immediately. User won't have to wait until partner matching process completes.

D.13.1 Procedure Specification

```
PROCEDURE Start_Partner_Matching(
    P_Api_Version_Number      IN  NUMBER,
    P_Init_Msg_List           IN  VARCHAR2 := FND_API.G_FALSE,
    P_Commit                  IN  VARCHAR2 := FND_API.G_FALSE,
    P_Validation_Level        IN  NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id          IN  NUMBER   := FND_API.G_MISS_NUM,
    P_Identity_Salesforce_Id  IN  NUMBER,
    P_Salesgroup_Id           IN  NUMBER,
    P_Lead_id                 IN  NUMBER,
    X_Return_Status           OUT VARCHAR2,
    X_Msg_Count               OUT NUMBER,
    X_Msg_Data                OUT VARCHAR2
    );
```

D.13.2 Current Version

2.0

D.13.3 Parameter Descriptions

Notes

- Api_version_number will be set to 2.0.
- If p_salesgroup_id is not passed in, this API will find a group_id for the current user.

- If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
- If a flag column is not passed in, default it to 'Y' or 'N'.

Table D–19 describes the IN parameters for the Start Partner Matching API.

Table D–19 IN Parameters

Parameter	Data Type	Required	Description
P_lead_id	NUMBER	Yes	Lead Identifier that user wants to do partner matching

Note: This procedure does not have OUT parameters.

D.14 Create Sales Lead Contacts

This API creates sales lead contacts for an existing sales lead.

D.14.1 Procedure Specification

```
PROCEDURE Create_sales_lead_contacts(
    P_Api_Version_Number       IN   NUMBER,
    P_Init_Msg_List            IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                   IN   VARCHAR2     := FND_API.G_FALSE,
    p_validation_level         IN   NUMBER       := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag        IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag               IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id           IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id   IN   NUMBER       := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl   IN   AS_UTILITY_PUB.Profile_Tbl_Type := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl   IN   SALES_LEAD_CONTACT_Tbl_Type
                                         := G_MISS_SALES_LEAD_CONTACT_Tbl,
    p_SALES_LEAD_ID            IN   NUMBER,
    X_SALES_LEAD_CNT_OUT_Tbl   OUT  SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status            OUT  VARCHAR2,
    X_Msg_Count                OUT  NUMBER,
    X_Msg_Data                 OUT  VARCHAR2
    );
```

D.14.2 Current Version

1.0

D.14.3 Parameter Descriptions

Table D–20 describes the IN parameters for the Create Sales Lead Contacts API.

Table D–20 IN Parameters

Parameter	Data Type	Required	Descriptions
p_api_version	Number	Y	Caller version number. This will be compared against the API version number to detect incompatibility.
p_init_msg_ list	VarChar2	N	Flag to indicate if the message stack should be initialized. Default : FND_API.g_false.
p_commit	VarChar2	N	Flag to indicate if the changes should be committed on success. Default : FND_API.g_ false.
p_validation_ level	Number	N	Level of validation required. NONE means no validation will be done in the API, and FULL means all the validations (item level, record level) will be performed.
p_check_ access_flag	VarChar2	N	-
p_admin_flag	VarChar2	N	-
p_admin_ group_id	VarChar2	N	-
p_identity_ salesforce_id	Number	N	-
p_sales_lead_ profile_tbl	AS_ UTILITY_ PUB.Profile_ tbl_type	N	-
P_sales_lead_ id	Number	Y	The unique identifier of the sales lead.

Table D–21 describes the OUT parameters for the Create Sales Lead Contacts API.

Table D–21 OUT Parameters

Parameter	Data Type	Descriptions
x_return_ status	VarChar2	See Section D.2.2, "Standard OUT Parameters".
x_msg_count	Number	See Section D.2.2, "Standard OUT Parameters".
x_msg_data	VarChar2	See Section D.2.2, "Standard OUT Parameters".
X_sales_lead_ cnt_out_tbl	sales_Lead_ cnt_out_tbl_ type	Contains the record which contains the Lead Contact ID for the sales lead.

D.15 Update Sales Lead Contacts

This API is used to update sales contact information in the table. The API will raise an exception if the record matching the Sales Lead Contact ID and Object Version Number passed does not exist.

D.15.1 Procedure Specification

```
PROCEDURE Update_sales_lead_contacts(
    P_Api_Version_Number         IN   NUMBER,
    P_Init_Msg_List              IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                     IN   VARCHAR2     := FND_API.G_FALSE,
    p_validation_level           IN   NUMBER       := FND_API.G_VALID_LEVEL_
FULL,
    P_Check_Access_Flag          IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN   NUMBER       := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN   AS_UTILITY_PUB.Profile_Tbl_Type := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl     IN   SALES_LEAD_CONTACT_Tbl_Type,
    X_SALES_LEAD_CNT_OUT_Tbl     OUT  SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status              OUT  VARCHAR2,
    X_Msg_Count                  OUT  NUMBER,
    X_Msg_Data                   OUT  VARCHAR2
    );
```

D.15.2 Current Version

1.0

D.15.3 Parameter Descriptions

Notes

Raise an exception if the object_version_number does not match.

Table D–22 describes the IN parameters for the Update Sales Lead Contacts API.

Table D–22 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	Number	Y	Caller version number. This will be compared against the API version number to detect incompatibility .
p_init_msg_list	VarChar2	N	Flag to indicate if the message stack should be initialized. Default : FND_API.g_false.
p_commit	VarChar2	N	Flag to indicate if the changes should be committed on success. Default : FND_API.g_false.
p_validation_level	Number	N	Level of validation required. NONE means no validation will be done in the API and FULL means all the validations (item level, record level) will be performed.
p_check_access_flag	VarChar2	N	-
p_admin_flag	VarChar2	N	-
p_admin_group_id	VarChar2	N	-
p_identity_salesforce_id	Number	N	-
p_sales_lead_profile_tbl	AS_UTILITY_PUB.Profile_tbl_type	N	-
P_sales_lead_id	Number	Y	The unique identifier of the sales lead.

Table D–23 describes the OUT parameters for the Update Sales Lead Contacts API.

Table D–23 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See Section D.2.2, "Standard OUT Parameters".
x_msg_count	Number	See Section D.2.2, "Standard OUT Parameters".
x_msg_data	VarChar2	See Section D.2.2, "Standard OUT Parameters".
X_sales_lead_cnt_out_tbl	sales_Lead_cnt_out_tbl_type	Contains the record which contains the Lead Contact ID for the sales lead.

D.16 Delete Sales Lead Contacts

This API calls the table handler Delete_Sales_Lead_Contacts and then calls a procedure to update the AS_SALES_LEAD_CONTACT table, if the primary contact is deleted and other contact is marked as primary.

D.16.1 Procedure Specification

```
PROCEDURE Delete_sales_lead_contacts(
    P_Api_Version_Number         IN   NUMBER,
    P_Init_Msg_List              IN   VARCHAR2     := FND_API.G_FALSE,
    P_Commit                     IN   VARCHAR2     := FND_API.G_FALSE,
    p_validation_level           IN   NUMBER       := FND_API.G_VALID_LEVEL_
FULL,
    P_Check_Access_Flag          IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN   VARCHAR2     := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN   NUMBER       := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN   NUMBER       := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN   AS_UTILITY_PUB.Profile_Tbl_Type := AS_
UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl     IN   SALES_LEAD_CONTACT_Tbl_type,
    X_SALES_LEAD_CNT_OUT_Tbl     OUT  SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status              OUT  VARCHAR2,
    X_Msg_Count                  OUT  NUMBER,
    X_Msg_Data                   OUT  VARCHAR2
    );
```

D.16.2 Current Version

1.0

D.16.3 Parameter Descriptions

Notes

Raise an exception if the object_version_number does not match.

Table D–24 describes the IN parameters for the Delete Sales Lead Contacts API.

Table D–24 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	Number	Y	Caller version number. This will be compared against the API version number to detect incompatibility.
p_init_msg_ list	VarChar2	N	Flag to indicate if the message stack should be initialized. Default : FND_API.g_false.
p_commit	VarChar2	N	Flag to indicate if the changes should be committed on success. Default : FND_API.g_ false.
p_validation_ level	Number	N	Level of validation required. NONE means no validation will be done in the API and FULL means all the validations (item level, record level) will be performed.
p_check_ access_flag	VarChar2	N	-
p_admin_flag	VarChar2	N	-
p_admin_ group_id	VarChar2	N	-
p_identity_ salesforce_id	Number	N	-
p_sales_lead_ profile_tbl	AS_ UTILITY_ PUB.Profil e_tbl_type	N	-
P_sales_lead_ id	Number	Y	The unique identifier of the sales lead.

Table D–25 describes the OUT parameters for the Delete Sales Lead Contacts API.

Table D–25 OUT Parameters

Parameter	Data Type	Description
x_return_ status	VarChar2	See Section D.2.2, "Standard OUT Parameters".
x_msg_count	Number	See Section D.2.2, "Standard OUT Parameters".
x_msg_data	VarChar2	See Section D.2.2, "Standard OUT Parameters".
X_sales_lead_ cnt_out_tbl	sales_Lead_ cnt_out_tbl_ type	Contains the record which contains the Lead Contact ID for the sales lead.

Index

C

D

G

I

L

M

O

Q

R

S

T