

Oracle® Leads Management

Implementation and Administration Guide

Release 12.1

Part No. E13448-02

April 2009

Oracle Leads Management Implementation and Administration Guide, Release 12.1

Part No. E13448-02

Copyright © 2003, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Debjit Nag

Contributor: Ed Carter

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

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

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

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Introduction to Oracle Leads Management

Oracle Leads Management Overview..... 1-1

2 Business Flows

Business Flows in Oracle Leads Management..... 2-1

Capturing and Cleaning Leads..... 2-1

Processing Leads..... 2-3

About the Leads Processing Engine..... 2-4

Scenario Of Qualifying Leads..... 2-4

The Qualification Engine..... 2-4

The Rating Engine..... 2-5

The Channel Selection Engine..... 2-6

Creating Rule Sets..... 2-7

3 Capturing and Cleaning Leads

Leads from Multiple Sources..... 3-2

The Lead Import Process..... 3-3

Importing Leads..... 3-4

Importing leads from the HTML Interface..... 3-4

Evaluating Errors During Lead Import..... 3-5

Correcting Errors During Lead Import..... 3-6

Importing Leads From a Flat File..... 3-6

Imported Leads and Oracle Sales Tables.....	3-9
The Import Sales Lead Concurrent Program.....	3-10
Import Sales Lead Concurrent Program Flow.....	3-11
Import Lead Inactive Parties Flow.....	3-13
Before Running the Concurrent Program.....	3-14
Setting Up DQM Match Rules.....	3-15
Setting Profiles Used by the Import Sales Lead Concurrent Program.....	3-16
Creating Valid Lookup Codes.....	3-17
Setting up Territories.....	3-17
Loading the Import Interface Tables.....	3-18
Running the Import Sales Lead Concurrent Program.....	3-18
Enhancing the Performance of the Import Sales Lead Concurrent Program.....	3-20
Limitation of the Import Sales Lead Concurrent Program.....	3-21
Data Quality.....	3-22
Customer Data Quality.....	3-22
Setting Up DQM Staging Schema.....	3-27
Designing Matching Rules to Detect Duplicate Customer or Person.....	3-28
Leads Data Quality.....	3-43
Customizing the Deduplication Rule.....	3-47
Custom Codes with the Lead Import Program.....	3-49
Seeded Business Events.....	3-49
Custom User Hook.....	3-54
Purging Staged Lead Records.....	3-56

4 Linking Interactions to Leads

Interaction Matching Engine Overview.....	4-1
Interaction Types and Interaction Scores.....	4-2
Interaction Score Threshold.....	4-3
Functioning of the Interaction Matching Engine.....	4-3
The Mining Behavior of the Interaction Matching Engine.....	4-4
Linking an Interaction with Existing Leads.....	4-5
Linking if No Lead Exists for the Interaction.....	4-6
Running the Concurrent Program.....	4-6
Setting Up Interaction Types.....	4-7
Interaction Selection Rule Sets.....	4-7
Creating an Interaction Rule Set.....	4-7

5 Processing Leads

Leads Processing Engine Overview.....	5-1
Rule Set Details.....	5-2

Rule.....	5-2
Attributes.....	5-3
The Qualification Engine.....	5-3
The Rating Engine.....	5-4
Setting Up Ratings.....	5-4
The Channel Selection Engine.....	5-5
Setting Up Channels.....	5-5
Best Practices.....	5-6
Rule Flows.....	5-6
Purging Unqualified Leads.....	5-7
Setting Up Lead Assignments.....	5-7
Setting Up Automatic Lead Assignment.....	5-8
Setting Up Immature Lead Assignment.....	5-9
Routing Leads Using a User Hook.....	5-9
Setting Up Lead Status.....	5-16
Using Custom Attributes.....	5-16
Setting Up Time Frames.....	5-16
Customizing Time Frames.....	5-17
 6 Monitoring Leads	
Monitoring Engine Overview.....	6-1
Scope of a Monitor.....	6-2
Smart Time Frames and Expiration Dates.....	6-3
Timeouts in the Monitoring Engine.....	6-3
Running the Workflow Background Process Concurrent Program.....	6-5
Creating a Monitor Rule.....	6-5
Monitoring Engine Conditions.....	6-5
Viewing the Monitor Log.....	6-7
 7 Operational Reports	
Imports Records Manager Reports.....	7-1
Lead Processing History Reports.....	7-1
Rule Performance Reports.....	7-2
Rule Diagnostics Reports.....	7-2
Lead Linking Report.....	7-3
 A System Profile Options	
Setting System Profile Options.....	A-1
System Profile Options in Oracle Leads Management.....	A-1

B Seeded Data

Seeded Attributes	B-1
Seeded Lead Attributes.....	B-1
Seeded Interaction Matching Engine Attributes.....	B-6
Seeded Lookups	B-6
Time Frame.....	B-6
Lead Rank.....	B-7
Lead Status.....	B-7

C Concurrent Programs

Running Concurrent Programs	C-1
Concurrent Programs in Oracle Leads Management	C-1

D Oracle Leads Management API Reference

Oracle Leads Management Procedures	D-2
Parameter Specifications	D-4
Standard IN Parameters	D-4
Standard OUT Parameters	D-6
Type Declarations	D-8
Create Sales Lead	D-16
Procedure Specification	D-16
Parameter Descriptions	D-18
Update Sales Lead	D-19
Procedure Specification	D-19
Parameter Descriptions	D-20
Update Sales Lead Lines	D-21
Procedure Specification	D-21
Parameter Descriptions	D-22
Delete Sales Lead Lines	D-23
Procedure Specification	D-23
Lead Process After Create	D-24
Procedure Specification	D-24
Parameter Descriptions	D-25
Lead Process After Update	D-26
Procedure Specification	D-26
Parameter Descriptions	D-27
Run Lead Engines	D-27
Procedure Specification	D-27
Parameter Descriptions	D-28

Build Lead Sales Team.....	D-29
Procedure Specification.....	D-29
Parameter Descriptions.....	D-30
Rebuild Lead Sales Team.....	D-30
Procedure Specification.....	D-30
Parameter Descriptions.....	D-31
Update Sales Lead Contacts.....	D-32
Procedure Specification.....	D-32
Parameter Descriptions.....	D-33
Delete Sales Lead Contacts.....	D-35
Procedure Specification.....	D-35
Parameter Descriptions.....	D-36

Glossary

Index

Send Us Your Comments

Oracle Leads Management Implementation and Administration Guide, Release 12.1

Part No. E13448-02

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

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

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

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

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

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

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

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

Preface

Intended Audience

Welcome to Release 12.1 of the *Oracle Leads Management Implementation and Administration Guide*.

This guide is intended for sales and marketing users who are responsible for processing leads. This guide assumes you have a working knowledge of the following:

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

- Oracle Self-Service Web Applications
- The Oracle Applications graphical user interface.

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

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

How To Use This Guide

The Oracle Leads Management Implementation and Administration Guide contains the information you must understand to use Oracle Leads Management. This guide contains the following chapters:

- Chapter 1 provides an overview of Oracle Leads Management.
- Chapter 2 provides business flows and scenarios that help you to understand

Oracle Leads Management.

- Chapter 3 describes the setup tasks that you need to perform, to process leads.
- Chapter 4 describes the Interaction Matching Engine.
- Chapter 5 describes how leads are processed.
- Chapter 6 describes the Monitoring Engine.
- Chapter 7 describes the various operational reports 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 is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

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

Other Information Sources

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

If this guide refers you to other Oracle Applications documentation, use only the Release 12 versions of those guides.

Online Documentation

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

- **PDF Documentation**- See the Documentation CD provided with each release for current PDF documentation for your product. This Documentation CD is also available on My Oracle Support and is updated frequently.
- **Online Help** - You can refer to Oracle Applications Help for current HTML online help for your product. Oracle provides patchable online help, which you can apply to your system for updated implementation and end user documentation. No system downtime is required to apply online help.

- **12 Release Content Document** - Refer to the Release Content Document for new features listed release. The Release Content Document is available on My Oracle Support.

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

TTY Relay Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at <http://www.consumer.att.com/relay/tty/standard2.html>. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

1 Introduction to Oracle Leads Management

2	Business Flows
3	Capturing and Cleaning Leads
4	Linking Interactions to Leads
5	Processing Leads
6	Monitoring Leads
7	Operational Reports
A	System Profile Options
B	Seeded Data
C	Concurrent Programs
D	Oracle Leads Management API Reference
	Glossary

Related Information Sources

Integration Repository

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

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

Online Documentation

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

- **PDF** - PDF documentation is available for download from the Oracle Technology Network at <http://otn.oracle.com/documentation>.
- **Online Help** - Online help patches (HTML) are available on My Oracle Support.
- **About Documents and Release Notes** - Refer to the About Document or Release Notes for the mini-pack or family pack that you have installed to learn about new documentation or documentation patches that you can download. About Documents and Release Notes are available on My Oracle Support.
- **My Oracle Support Knowledge tab** - The My Oracle Support Knowledge tab lets you browse the knowledge base by technology, industry, integration, application, documentation, training, and services, to find all documents for a product area. Use the Knowledge Home to search for release-specific information, such as FAQs, recent patches, alerts, white papers, troubleshooting tips, and other archived documents.

- **Oracle E-Business Suite Electronic Technical Reference Manuals** - Each Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications and integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on My Oracle Support as part of Online Documentation.

Guides Related to All Products

Oracle Applications User's Guide

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

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

Guides Related to This Product

Oracle Trading Community Architecture Administration Guide

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

Oracle Trading Community Architecture User Guide

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

Installation and System Administration

Oracle Applications Concepts

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12, or contemplating significant changes to a configuration. After describing the Oracle Applications architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus the installation and configuration choices that may be available.

Oracle Applications Installation Guide: Using Rapid Install

This book is intended for use by anyone who is responsible for installing or upgrading Oracle Applications. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle Applications Release 12, or as part of an upgrade from Release 11*i* to Release 12. The book also describes the steps needed to install the technology stack components only, for the special situations where this is applicable.

Oracle Applications Maintenance Utilities

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 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 forms so that they integrate with Oracle Applications. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Oracle Applications User Interface Standards for Forms-Based Products

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

Forms.

Other Implementation Documentation

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 Applications Flexfields Guide

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

Oracle E-Business Suite Diagnostics User's Guide

This guide contains information on implementing, administering, and developing diagnostics tests in the Oracle E-Business Diagnostics framework.

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

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

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide

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

Training and Support

Training

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

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

Support

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

Do Not Use Database Tools to Modify Oracle Applications Data

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

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

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

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

record of changes.

Introduction to Oracle Leads Management

This chapter covers the following topics:

- Oracle Leads Management Overview

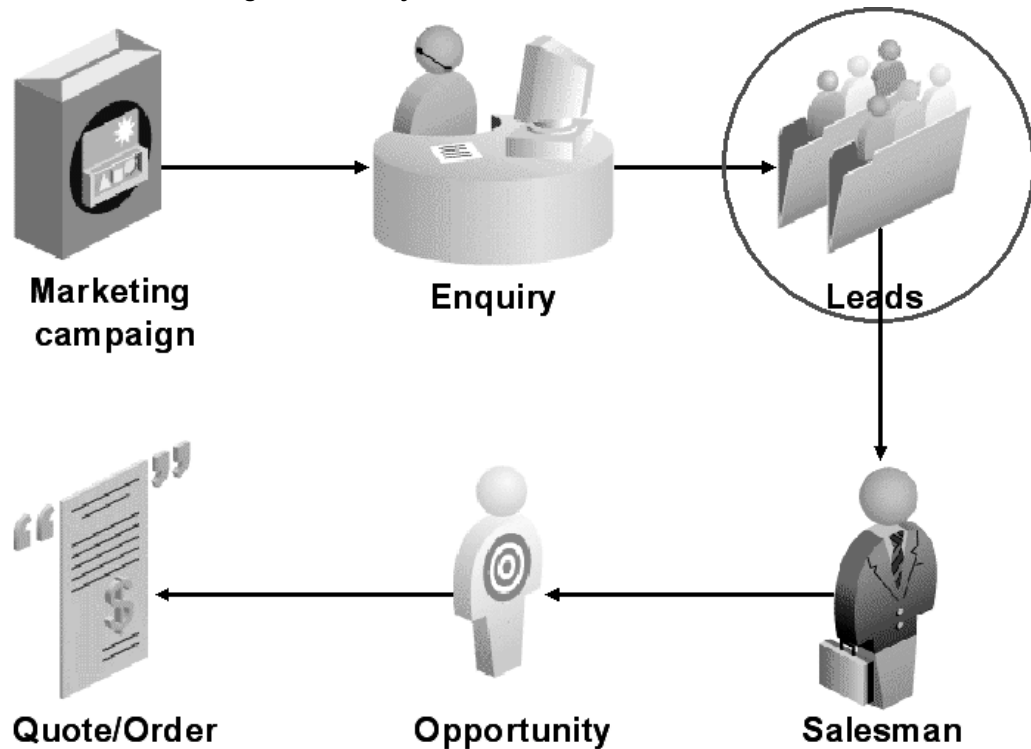
Oracle Leads Management Overview

Oracle Leads Management provides solutions to automate and optimize prospect-to-sales conversion across the enterprise. It provides a staging area for all prospective leads. In the staging area, the leads go through data quality processing, qualification, and prioritization before being assigned to sales teams for conversion.

Leads in the Marketing and Sales Cycles

Where does a lead appear in the Marketing and Sales cycles? Figure: Leads in the Marketing and Sales Cycles, page 1-2 illustrates this.

Leads in the Marketing and Sales Cycles



1. **Marketing Campaign:** The marketing department kicks off a campaign by targeting prospective customers.
2. **Enquiry:** In response to the campaign, interested prospects make an enquiry. At this point, the details of the prospects are captured.
3. **Leads:** The details of prospects who respond to a campaign or who make an enquiry are stored as leads. Leads are prospective customers. After the leads are processed, they are assigned to appropriate sales teams.
4. **Opportunity:** The sales teams convert some leads to opportunities. An opportunity is still a prospect, but more mature. An opportunity is more likely to be converted to an order.
5. **Quote/Order:** Promising opportunities receive a quote from the sales team. When the quote is accepted, the opportunity becomes an order.

Oracle Leads Management Features

The Oracle Leads Management features include:

- **Capture of leads** - Leads are captured from various sources such as marketing

campaigns, events, and referrals.

- **Customer and lead data quality** - Captured records are processed for customer and leads quality.
- **Interactions linked to leads** - Relevant interactions are attached to leads, and wherever appropriate, new leads are generated from interactions.
- **Real time flexible rules engine** - The Leads Processing Engines can be set up for lead filtering, evaluation and distribution based on business rules.
- **Integrated monitoring** - Sales-ready leads are monitored for prompt action by the sales team.
- **Lead utilization and effectiveness analysis** - Using the operational reports, the effectiveness of the engines and the status of the leads processed by them are tracked.

Business Flows

This chapter covers the following topics:

- Business Flows in Oracle Leads Management
- About the Leads Processing Engine
- Scenario Of Qualifying Leads
- The Qualification Engine
- The Rating Engine
- The Channel Selection Engine
- Creating Rule Sets

Business Flows in Oracle Leads Management

In Oracle Leads Management, leads are captured and processed before they are assigned to appropriate sales channels. The following sections give a high-level flow of a lead in Oracle Leads Management. Each of the processes are discussed at length in their respective chapters.

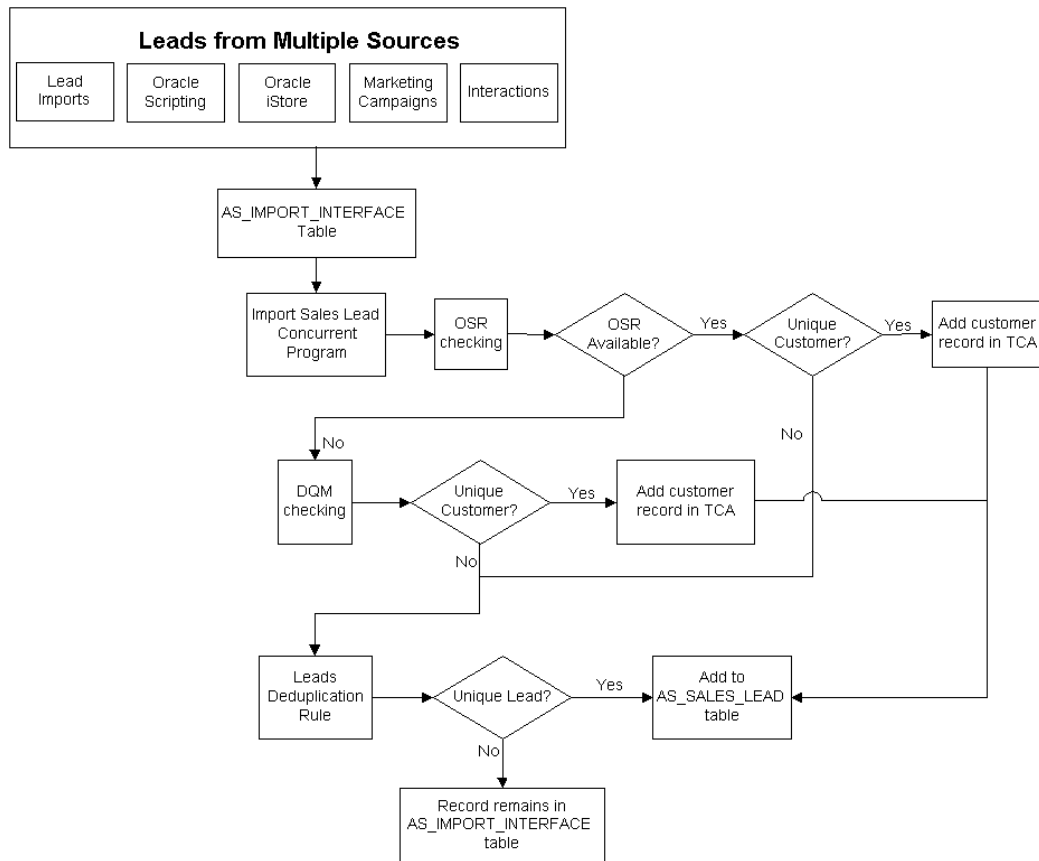
Topics in this section include:

- Capturing and Cleaning Leads
- Processing Leads

Capturing and Cleaning Leads

Figure: Business Flow for Capturing and Cleaning Leads, page 2-2 gives the flow of leads after they are stored in the `AS_IMPORT_INTERFACE` table. For more information on the import process, see Capturing and Cleaning Leads.

Business Flow for Capturing and Cleaning Leads

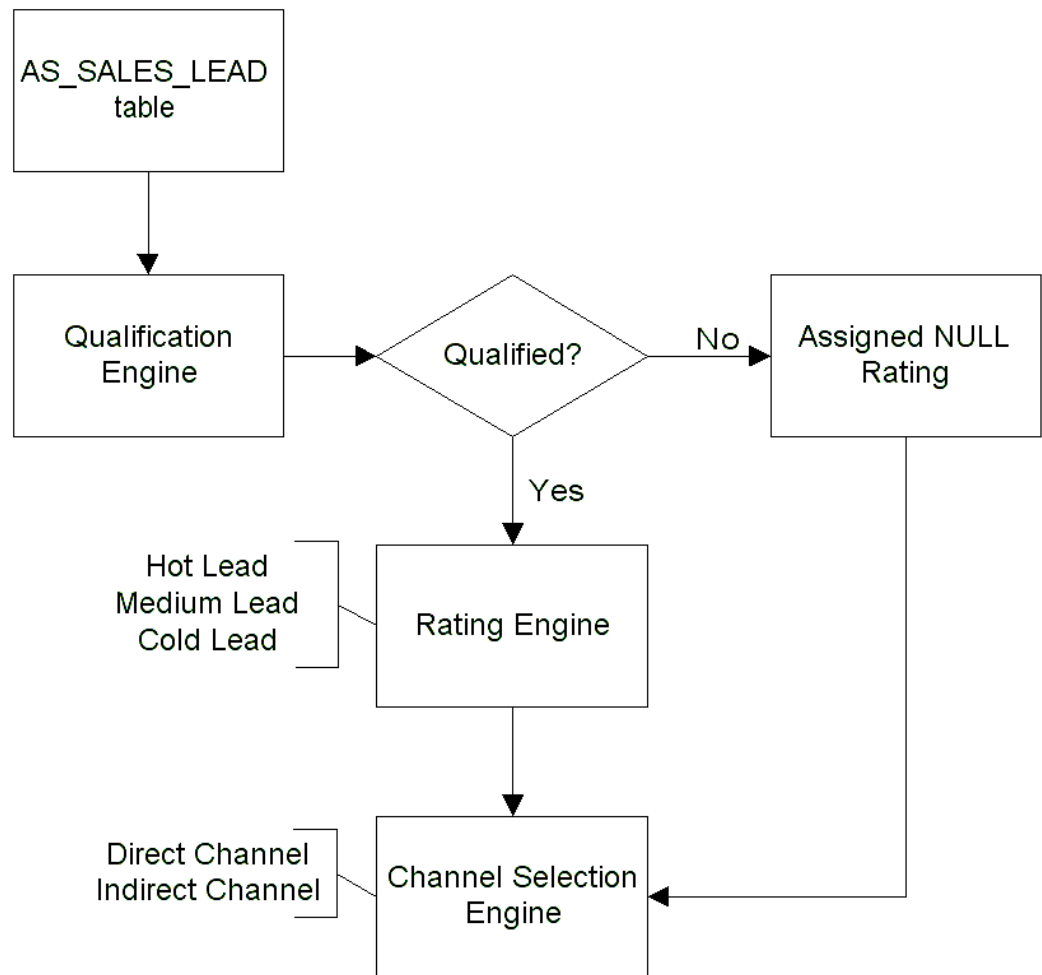


1. Leads enter Oracle Leads Management from multiple sources: Lead Imports, Oracle Scripting, Oracle *iStore*, Marketing Campaigns, and Interactions. The lead records are stored in the `AS_IMPORT_INTERFACE` table.
2. The Import Sales Lead concurrent program processes each lead record in the `AS_IMPORT_INTERFACE` table in the following order and manner:
 1. The lead record is checked for the existence of Original System Reference.
 2. Next, Data Quality Management (DQM) checks if a customer record for the lead already exists in TCA. If a record does not exist, then a new record is created in TCA.
 3. After DQM, the record is run through the Leads Deduplication rule to check if the lead record exists in the `AS_SALES_LEAD` table. If it does not exist, then the record is added to the `AS_SALES_LEAD` table.

Processing Leads

Figure: Business Flow for Processing Leads, page 2-3 gives the flow of leads after they enter the AS_SALES_LEAD table. For more information on the processing of leads, read the **Processing Leads** chapter.

Business Flow for Processing Leads



1. The leads in the AS_SALES_LEAD table are processed by the Lead Processing Engine.
2. The first engine that processes leads is the Qualification Engine. If the lead is qualified, then it is routed to the Rating Engine. If not, it is graded NULL, and routed to the Channel Selection Engine.
3. The Rating Engine assigns a rating to all the qualified leads. Examples of the grades could be Hot, Medium, Cold, and so on.

4. Based on the rating assigned to the lead, the Channel Selection engine assigns a sales channel to the lead. Examples of channel are Direct, Indirect, and so on.

About the Leads Processing Engine

The Qualification, Rating, and Channel Selection engines make up the Leads Processing Engine. The following scenario helps you to understand how to set up the Leads Processing Engine.

- Scenario Of Qualifying Leads, page 2-4

Scenario Of Qualifying Leads

In this scenario, the qualification engine is used to qualify leads.

Vision Enterprises is into the business of selling computers and computer accessories to organizations. Recently, they have run a road show called Vision Hardware to showcase sleek monitors, lightweight laptops, and high-end computers. Participating organizations were asked to fill forms with details such as product interest, budget status, and so on. These details have been captured, and they must now be processed by Oracle Leads Management so that the leads are followed up appropriately.

The Qualification, Rating, and Channel Selection engines must be set up. The following section details rule sets set up by Vision Enterprises to process leads. Create rule sets based on the following model.

- The Qualification Engine
- The Rating Engine
- The Channel Selection Engine
- Creating Rule Sets

The Qualification Engine

You can use the Qualification Engine to either qualify or disqualify leads. We use it to qualify leads in this scenario. A lead can be qualified when its attributes suggest interest in your products or services.

In this scenario, because we are using the engine only to qualify leads, the leads that are not processed by the engine are routed to the Channel Selection Engine. Qualified leads are routed to the Rating Engine.

Table: Qualification Rule Set Conditions, page 2-5 details the conditions for the qualification engine rule set.

Qualification Rule Set Conditions

Rule Set Component	Conditions	Remarks
Guards	Country=US	Guards define the domain of the rule set.
	Product Category=Monitors or Laptops or Computers	Only those leads from the US and with interest in monitors, laptops, or computers will be processed by this rule set.
Rules	Qualified Flag = YES	The Qualification Flag for all qualified leads is set to Yes. This is applicable to both Rule 1 and Rule 2.
Rule 1	Purchase Timeframe=Within 1 month, 1-3 months, 3-6 months	Rules determine the conditions and action to be performed on the lead.
	Purchase Amount Greater Than or Equals 50,000	Leads satisfying this criteria are qualified.
Rule 2	Purchase Timeframe=More than 1 Year	Leads satisfying this criteria are qualified.
	Purchase Amount Less Than 50,000	

The Rating Engine

The leads that are qualified are routed to the Rating Engine. Based on the attributes of the lead, Vision Enterprises uses the following ratings: A, B, C and D to assign a rating.

Table: Rating Engine Rule Set Conditions, page 2-6 details the conditions for the rating engine rule set.

Rating Engine Rule Set Conditions

Rule Set Component	Conditions	Remarks
Guards	Country=US Product Category=Monitors or Laptops or Computers	Guards define the domain of the rule set. Only those leads from the US and with interest in monitors, laptops, or computers will be processed by this rule set.
Rule 1	Purchase Timeframe=Within 1 month, and Budget Status=Approved: Grade A	Rules determine the conditions and action to be performed on the lead. Leads satisfying this criteria are rated Grade A.
Rule 2	Purchase Timeframe=1-3 months, and Budget Status=Approved: Grade B	Leads satisfying this criteria are rated Grade B.
Rule 3	Purchase Timeframe=3-6 months, and Budget Status=Approved: Grade C	Leads satisfying this criteria are rated Grade C.
Rule 4	Purchase Timeframe=3-6 months, and Budget Status=Pending: Grade D	Leads satisfying this criteria are rated Grade D.

The Channel Selection Engine

You can use the Channel Selection Engine to assign a sales channel to the leads. The Territory Assignment Program decides the sales teams to assign the leads to using the sales channel. Vision Enterprises uses the Direct and Indirect Channels.

Table: Channel Selection Engine Rule Set Conditions, page 2-7 details the conditions for the channel selection engine rule set.

Channel Selection Engine Rule Set Conditions

Rule Set Component	Conditions	Remarks
Guards	Country=US	Guards define the domain of the rule set. Only those leads from the US will be processed by this rule set.
Rule 1	Qualified Flag = Yes and Purchase Amount > 50,000, route to Direct Channel	Rules determine the conditions and action to be performed on the lead. Leads satisfying this criteria are routed to the Direct Channel.
Rule 2	Qualified Flag = Yes, Purchase Amount < 50,000, and Lead Rating Equals A, route to Indirect Channel A	Leads satisfying this criteria are routed to Indirect Channel A.
Rule 3	Qualified Flag = Yes, Purchase Amount < 50,000, and Lead Rating Not Equals A, route to Indirect Channel B	Leads satisfying this criteria are routed to Indirect Channel B.

Creating Rule Sets

Rule Sets are mediums through which the Qualification, Rating, and Channel Selection engines process leads. Rule sets contain guards and rules which are matched against lead attributes.

The following section guides you to create a rule set for any of the engines - Qualification, Rating, or Channel Selection, and add guards and rules to it.

Navigation: Log in as an administrator, and navigate to Administration > Leads > Processing Rules > Qualification/Rating/Channel Selection.

Notes

- **Start Date and End Date:**Dates between which the rule set is valid.
- **Status:** By default, the Status is Draft.

The status can be changed after the rule set is created.

- **Precedence:** Enter a number in the Precedence field. Each rule set can have a different precedence, to define the order of importance for evaluation (where 100 is higher than 1).
- **Guards:** To create guards, add attributes by clicking Add Attributes.
- **Rules:** Enter a name for the rule, and specify its order of evaluation.
 - If this is a qualification rule set: From the "If the condition is met Qualified Flag is set to" drop-down list, select 'Yes' if this rule set is to qualify leads, and 'No' if it is to disqualify leads. The Qualified Flag attribute in the lead record will get this value.
 - If this is a rating rule set: From the "If the condition is met Lead Rank is set to" drop-down list, select a rating. This rating is assigned to the lead if its attributes match the rule.
 - If this is a channel selection rule set: From the "If the condition is met Sales Channel is set to" drop-down list, select a sales channel. This channel is assigned to the lead if its attributes match the rule.

At any point in time, it is recommended that you do not have rule sets to both qualify and disqualify leads.

Capturing and Cleaning Leads

This chapter covers the following topics:

- Leads from Multiple Sources
- The Lead Import Process
- Importing Leads
- Importing leads from the HTML Interface
- Evaluating Errors During Lead Import
- Correcting Errors During Lead Import
- Importing Leads From a Flat File
- Imported Leads and Oracle Sales Tables
- The Import Sales Lead Concurrent Program
- Import Sales Lead Concurrent Program Flow
- Import Lead Inactive Parties Flow
- Before Running the Concurrent Program
- Setting Up DQM Match Rules
- Setting Profiles Used by the Import Sales Lead Concurrent Program
- Creating Valid Lookup Codes
- Setting up Territories
- Loading the Import Interface Tables
- Running the Import Sales Lead Concurrent Program
- Enhancing the Performance of the Import Sales Lead Concurrent Program
- Limitation of the Import Sales Lead Concurrent Program
- Data Quality
- Customer Data Quality

- Setting Up DQM Staging Schema
- Designing Matching Rules to Detect Duplicate Customer or Person
- Leads Data Quality
- Custom Codes with the Lead Import Program
- Seeded Business Events
- Custom User Hook
- Purging Staged Lead Records

Leads from Multiple Sources

Leads are captured into Oracle Leads Management from various sources. The primary sources are:

Oracle Scripting

To develop a personalized relationship with customers, marketing organizations use the branching functionality in Oracle Scripting. The branching functionality responds differently to the input of customers based on their profiles or the answers that they provide to questions. When a customer expresses interest in a product, a lead is created and managed by Oracle Leads Management.

Oracle Scripting contains seeded scripts focused on various marketing activities. These scripts can be used with minimum configuration and can be deployed as call-center scripts or web surveys. In addition to simplifying processes, scripts can help to ensure communication consistency. For more information on implementing seeded scripts, see the *Oracle Marketing Implementation Guide* and the *Oracle Scripting Implementation Guide*.

Oracle iStore

Customers who use Oracle iStore for purchases can be mined in as leads for a cross-sell or an up-sell. Also, when customers abandon a shopping cart before making the final purchase, leads are created from such records, and followed up.

Marketing Campaigns

The marketing department may run several campaigns, and capture leads. These leads may be imported into Oracle Leads Management from a .csv, .txt or a flat file. For more information, see Importing Leads, page 3-4. The marketing campaign generates leads based on the Installbase.

Interactions

An interaction is a record of communication between a potential customer and a

company representative. An interaction is generally timed and has an outcome or result that can be tracked. These interactions are tracked and leads are created from them.

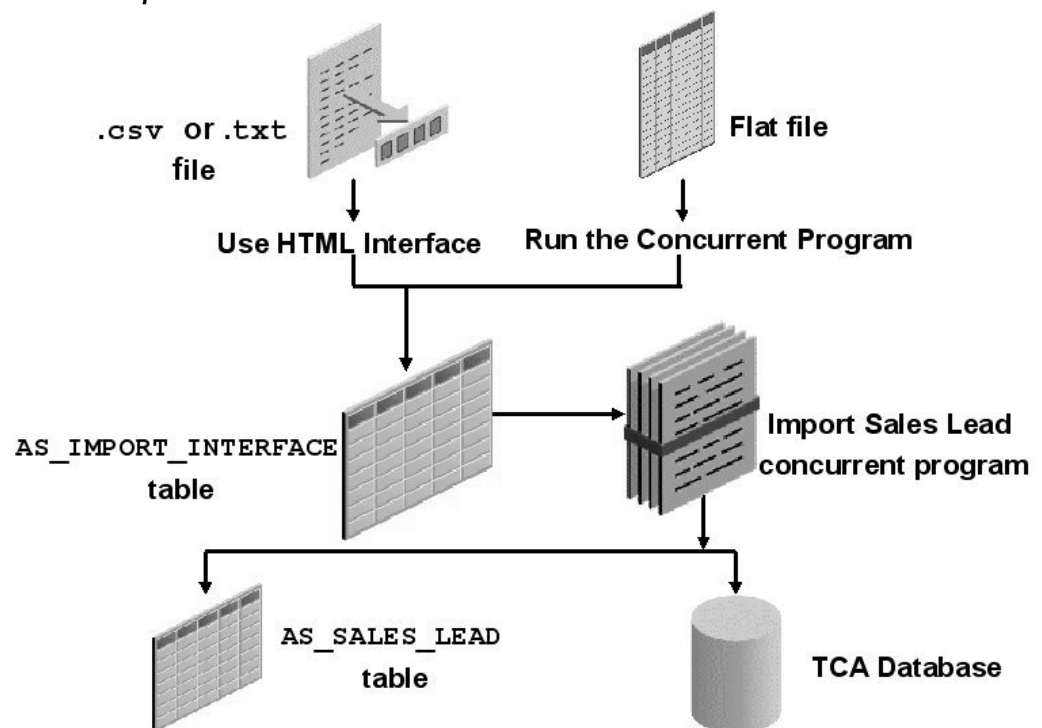
Partner Referrals

The Referral Management feature in Oracle Partner Management allows a partner to submit referrals to the vendor. After the vendor accepts the referrals, the referral becomes a lead. For more information about Referral Management, see the *Oracle Partner Management Partner User Guide*.

The Lead Import Process

Importing leads is one of the sources to capture leads into Oracle Leads Management. Figure: The Lead Import Process, page 3-3 depicts the methods of importing the leads and the processing that takes place after they are imported.

The Lead Import Process



You can import leads into Oracle Leads Management in two ways:

- Import data from a .csv or a .txt file, using the Lead Import utility in the HTML interface.
- Import data from a flat file by running the Lead Sales Table from Flat File concurrent program.

The imported records are stored in the `AS_IMPORT_INTERFACE` table. This table is an intermediary table that stages all lead records before they are refined and cleaned.

The Import Sales Lead concurrent program picks every lead from the `AS_IMPORT_INTERFACE` table, and runs it through Data Quality Management (DQM) to identify unique customer records, and through the Leads Deduplication rule to remove duplicate lead records.

All unique customer records identified by the DQM process are stored in the Trading Community Architecture (TCA) database. This database is a central repository that is accessible to all Oracle's E-Business Suite and ERP applications.

All unique lead records that are identified by the Leads Deduplication process are stored in the `AS_SALES_LEAD` table.

Importing Leads

Leads may be imported from a .csv, .txt, or from a flat file.

Topics in this section include:

- Importing Leads from the HTML Interface, page 3-4
- Evaluating Errors During Lead Import, page 3-5
- Correcting Errors During Lead Import, page 3-6
- Importing Leads From a Flat File, page 3-6
- Imported Leads and Oracle Sales Tables, page 3-9

Importing leads from the HTML Interface

You can import leads stored in a .csv or .txt file using the Import wizard from the HTML interface. Use the following procedure to import leads.

Prerequisite: A .csv or .txt file containing data for import is required.

Navigation: Log in as an administrator, and navigate to Audience Dashboard > Import.

Notes:

- **Source File:** 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.

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

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 a .csv or a .txt file, a .csv file or a

.txt file.

- **Column Delimiter:** Use the Column Delimiter list to select the delimiter used in the file to distinguish between two columns. Choose tilde (~) unless you are using SQL Loader.
- **Field Enclosed By:** Use the Field Enclosed By list to select the character that encloses each field in the file. This is required when the data in your file has special characters that must not be mistaken for the column delimiter.

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

- **Source Fields:** Source Fields are columns in your import file.

Target Fields: Target Fields are columns present in the table.

Select a Source field and a corresponding Target field.

>: The mapped fields appear in the Mapped Source Target fields section. Ensure that all mandatory fields are mapped.

After reviewing the details, you can import the lead. The lead is then processed.

Evaluating Errors During Lead Import

The Import Sales Lead concurrent program stores errors that occur during lead import in the AS_LEAD_IMPORT_ERRORS table.

Table: Status and Descriptions for Lead Import Errors in the HTML Interface, page 3-5
Status and Descriptions for Lead Import Errors in the HTML Interface gives the status and descriptions for the lead import errors that you can see in the HTML interface.

Status and Descriptions for Lead Import Errors in the HTML Interface

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

Status	Description
Duplicate	All records are duplicates.

Note: Do not change the status of an imported lead from `Success` to `New`. A status of `Success` means that a lead has been successfully imported and cannot be imported again.

Correcting Errors During Lead Import

To check for errors detected by the Import Sales Lead concurrent program during lead import, and correct them, as a prerequisite, you must have imported records into the `AS_IMPORT_INTERFACE` table.

Navigation: Log in as an administrator, and navigate to Audience Dashboard > Import.

Notes:

- Click the Import Source Line ID link, and modify the column which has an error and click update

Note: When you modify any of the details for a record, the original record is not updated. Instead, a new record is created and updated with the modifications.

Importing Leads 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`). Note the following points:

- To distinguish between imports, use different batch IDs.
- There are some mandatory columns in the `AS_IMPORT_INTERFACE` table. For such columns, a `null` or incorrect entry in the field results in a database error. You must provide valid values to these columns. Run SQL*Plus queries to obtain some of the values required by this table.
- If you have flexfields set up in your application, you must also populate the `AS_IMP_SL_FLEX` table. See Flexfields, page 3-9 for more information.

There are two ways to import leads into the `AS_IMPORT_INTERFACE` table from a flat file.

- Importing Leads by Running the Concurrent Program, page 3-7
- Importing Leads by Running SQL Loader Manually, page 3-7

Importing Leads by Running the Concurrent Program

Use the following details to run the Load Sales Lead Interface Table from Flat File concurrent program.

- You must have read and write permissions for the directory on the server.
- The data in your import file must contain the required fields and the Load Status of each record must be `NEW`.

Prerequisite: FTP the tilde(~) delimited flat file with lead data to a directory on the server. The file must have the .dat extension.

Responsibility: Oracle Sales Administrator

Name of program: Load Sales Lead Interface Table from Flat File

Parameter:

P_DATAFILE - Name of the .dat file

Schedule - Once

For steps to run the concurrent program, see Running Concurrent Programs., page C-1

Note: The Load Sales Lead Interface Table from Flat File 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.

Related Topics

- For a sample data file, see Sample Flat File, page 3-8.
- For information on flexfields, see Flexfields, page 3-9.

Importing Leads by Running SQL Loader Manually

If you have access to Oracle SQL Loader, you can import details from the flat file without running the Load Sales Lead Interface Table from Flat File concurrent program. You must upload the flat file to a server, and create the SQL Loader file which will import the records from the flat file.

Prerequisites:

- 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.
1. FTP the tilde(~) delimited flat file with lead data to a directory on the server. The file must have the .dat extension.
 - The directory on the server must have read and write permissions.
 - The data in your import file must contain the required fields and the Load Status of each record must be NEW.

1. Create an Oracle SQL Loader parameter file. Here is what a sample SQL Loader parameter file will look like:

```
userid=<username>/<password>
control=ASTSLIMP.CTL

data=<path><lead import data file name>.dat
```

2. Upload lead records using this parameter file as command line parameter to SQL Loader.

Records from the flat file are imported and processed by the Import Sales Lead concurrent program.

Sample Flat File

Below is a sample flat file for loading the AS_IMPORT_INTERFACE table. 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
~15000~5000000~6000000~DECISIONMAKER~1023472~1900~N~M~MR~~Hislas
t~Hisfirst~A~AccountsPayableSupervisor~ARC~101~926-2667~GEN~650~
123~926-2600~650~www.xyz.com~abc@xyz.com~Y~Y~N~N~NEW~DECISION_MA
KER~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~~~USERENTERED~XYZHQ~~~10~~~~~
~94065-1282~~~HQ~~~~~1008~~~~~XYZ~~~~~15-Aug-01~~~~~N~19
87SIC~~101~550000~~~N~MARKET~~10~~~Importantcontact~~Y~Informati
onTechnology~IT~DECISION_MAKER~10588~N~Y~USER_ENTERED~1~1~~~~MAI
LHTML~123~~~~~TSTENH
```


Flexfields

Use the AS_IMP_SL_FLEX table to store the flexfield 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 obsolete. Hence, the Import Sales Lead concurrent program does not support these columns. For information about how to plan and set up flexfields, see the *Oracle Applications Flexfields Guide*.

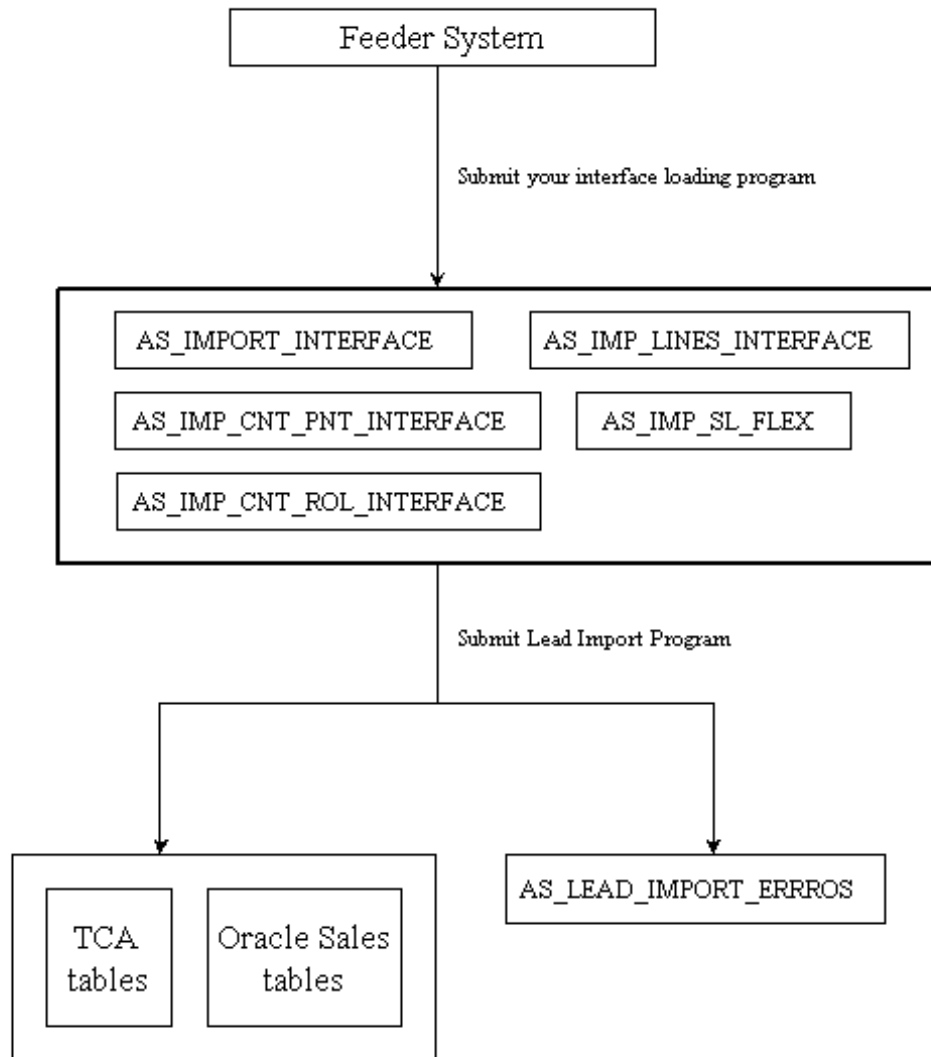
Imported Leads and Oracle Sales Tables

Figure: Leads Data Imported into Oracle Sales Tables, page 3-10 shows how lead information is imported into Oracle Sales tables.

From the feeder system, imported leads are stored in the interface tables. These are the AS_IMPORT_INTERFACE, AS_IMP_CNT_PNT_INTERFACE, AS_IMP_CNT_ROL_INTERFACE, AS_IMP_LINES_INTERFACE, and the AS_IMP_SL_FLEX tables.

After the Import Sales Lead concurrent program is run, appropriate records are created in the TCA database, Oracle Sales tables, and imported records that resulted in errors are stored in the AS_LEAD_IMPORT_ERRORS table.

Leads Data Imported into Oracle Sales Tables



The Import Sales Lead Concurrent Program

The Import Sales Lead concurrent program runs every record in the `AS_IMPORT_INTERFACE` table through the DQM and Leads deduplication processes. The TCA database is updated with any unique customer records. Unique lead records are stored in the `AS_SALES_LEADS` table. The concurrent program calls the leads processing engines to filter, qualify, rate and channel these leads to the sales teams.

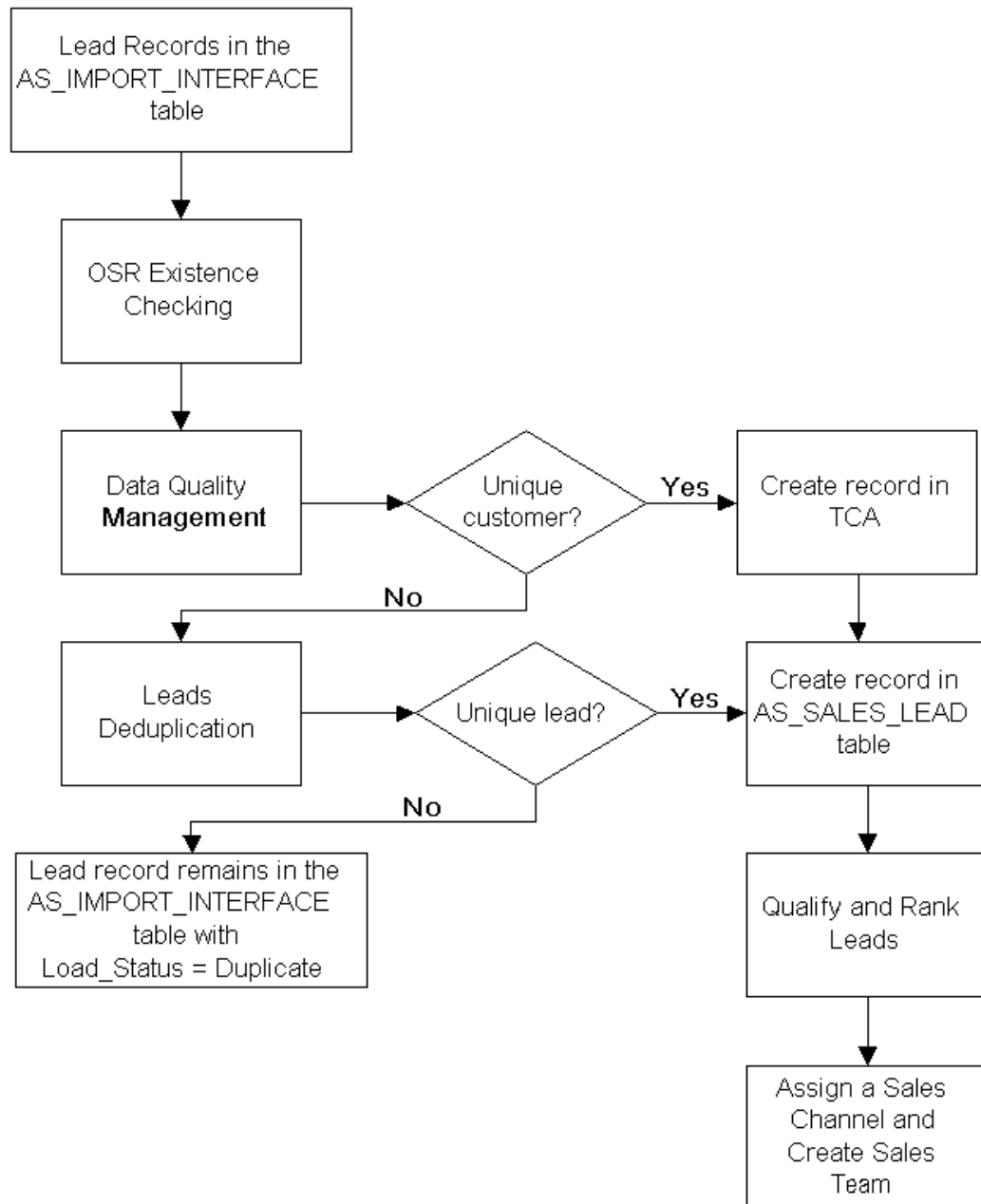
To set up, manage, and run the concurrent program, follow the procedures in these sections:

- Import Sales Lead Concurrent Program Flow, page 3-11
- Before Running the Concurrent Program, page 3-14
- Running the Import Sales Lead Concurrent Program, page 3-18
- Enhancing the Performance of the Import Sales Lead Concurrent Program, page 3-20
- Limitation of the Import Sales Lead Concurrent Program, page 3-21

Import Sales Lead Concurrent Program Flow

Figure: Import Sales Lead Concurrent Program Flow, page 3-12 illustrates the manner and sequence in which the Import Sales Lead concurrent program processes records.

Import Sales Lead Concurrent Program Flow



The Import Sales Lead concurrent program does the following:

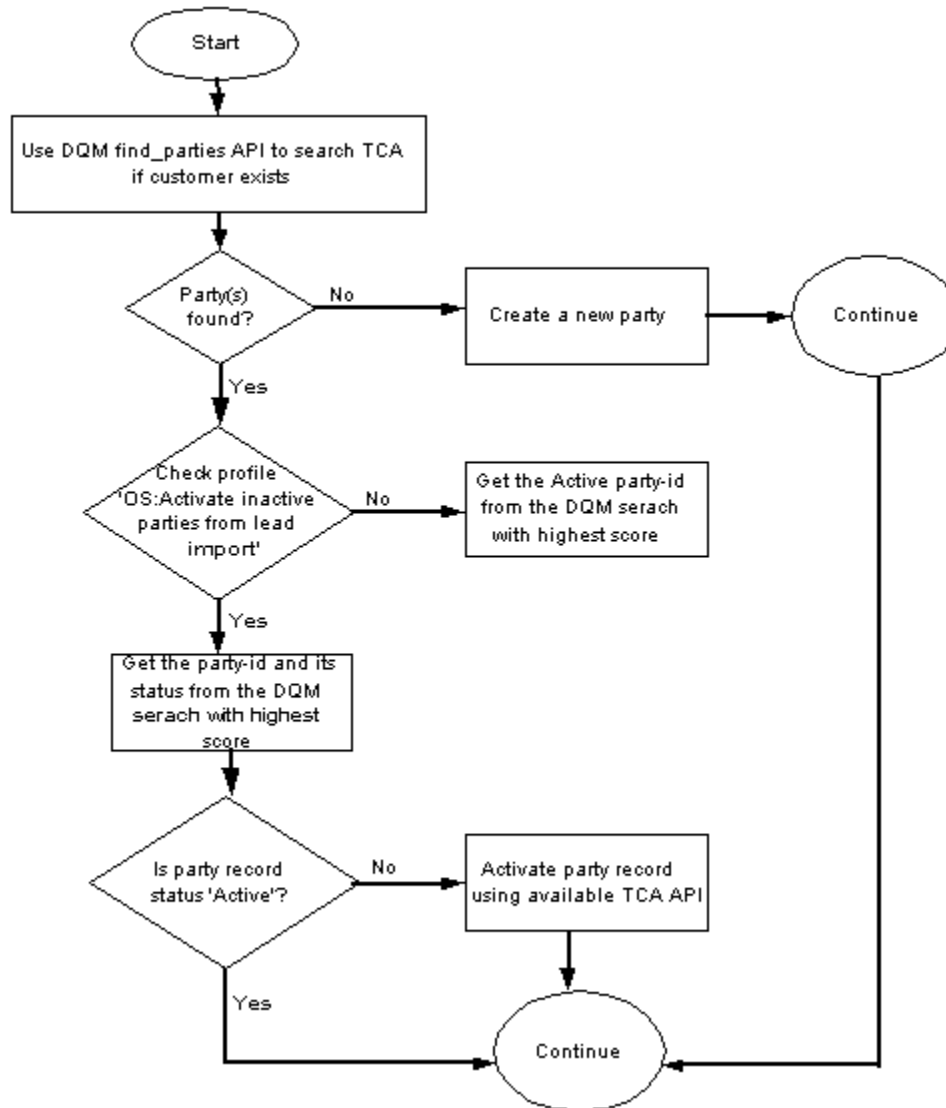
1. Checks for the existence of Original System Reference (OSR) using Leads Data Quality. See [Checking for Duplicate Original System Reference](#), page 3-23.
2. Checks for the existence of customer, address, contact, and contact points using DQM Data Quality. See [Customer Data Quality](#), page 3-22.

3. Creates a record in the TCA database, if the imported record is unique.
4. Checks for duplicate leads using Leads Data Quality. See Leads Data Quality, page 3-43.
5. Creates a lead in the AS_SALES_LEAD table, if the imported record is unique.
6. Qualifies and ranks the lead using the Leads Processing Engine.
7. Identifies the sales team, and assigns the lead to the owner of the sales team.
8. Creates a sales team to interact with the customer (the lead's organization), if required.

Import Lead Inactive Parties Flow

Figure: Lead Import - Inactive Parties Flow, page 3-14 illustrates the manner and sequence in which the Import Lead concurrent program activates 'Inactive' records.

Import Lead Inactive Parties Flow



Lead import will consider 'Inactive' parties returned by DQM search to create a lead. If the selected party is 'Inactive' lead import will call TCA API to activate it. This is controlled by the profile 'OS: Activate inactive parties from lead import' as some customers might not want 'Inactive' parties to get activated by the Lead Import program.

Before Running the Concurrent Program

The procedures in the following sections must be complete before running the Import Sales Lead concurrent program:

- Setting Up DQM Match Rules, page 3-28
- Setting Profiles Used by the Import Sales Lead Concurrent Program, page 3-16
- Creating Valid Lookup Codes, page 3-17
- Setting up Territories, page 3-17
- Loading the Import Interface Tables, page 3-18

Setting Up DQM Match Rules

All customer records are stored in the TCA database. DQM checks if a customer already exists in the TCA database. If a customer in the import record already exists in TCA, DQM returns the party_id of the customer. If not, the Import Sales Lead concurrent program creates a new customer record.

DQM also returns 'Inactive' party records from the search. Lead import considers these 'Inactive' party records and creates leads against them and also activates the party. Lead import calls TCA API to activate the same. This is controlled by the profile 'OS: Activate inactive parties from lead import'.

DQM checks the following attributes to identify unique records - Customers, Addresses, Contacts, and Contact Points.

DQM uses the matching rules to decide if a customer record exists in TCA. You can create a rule based on the business requirements in your organization.

The DQM match rules that will be used are dependent on the following profiles:

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

Use the following procedure to create a sample DQM rule to find duplicate contact records. The procedure is based on the sample rule explained in Identify Duplicate Contacts , page 3-38.

Navigation: Log in to Oracle Forms with the Trading Community Manager responsibility and navigate to Data Quality Management > Setup > Match Rules.

Notes

- **Purpose:** Select Identify Duplicates in the Purpose area.

Acquisition tab: In the Attribute Name column, enter Name. The Entity column displays Party.

In the Attribute Name column, enter Contact Name. The Entity column displays Contacts, and the Type column displays Custom Attribute.

In the Attribute Name column, enter Phone Number Flexible Format. The Entity column displays Contact_Points.

Transformation tab: Select each attribute, and select a transformation.

Scoring tab: In the Match Threshold field, enter 110.

In the Attribute Name column, enter Contact Name. The Entity column displays Contacts, and the Type column displays Custom Attribute.

Contact Name attribute: Select Exact String, and enter 100 for Weight (%) and select WR Person + Cleanse, and enter 90 for Weight (%).

e-Mail Address attribute: Select Exact (E-mail), and enter 100 for Weight (%).

Phone Number Flexible Format attribute: Select Exact, and enter 100 for Weight (%).

URL attribute: Select Cleanse (URL), and enter 100 for Weight (%).

Click Compile.

Related Topics

For other sample matching rules, see **Designing Matching Rules to Detect Duplicate Customer or Person**.

Setting Profiles Used by the Import Sales Lead Concurrent Program

Because the Import Sales Lead concurrent program triggers other programs as part of its flow, the following profiles must be set before it is run.

- OS: Use DQM Rule code to match Party
- OS: Use DQM Rule code to match Party Site
- OS: Use DQM Rule code to match Person
- OS: Use DQM Rule code to match Contact
- OS: Default Resource ID Used for Sales Lead Assignment

Prerequisites: Create DQM matching rules.

Navigation: Log in to Oracle Forms with the System Administrator responsibility and navigate to Profile > System > Open.

Notes

- **Site level:**

- **OS: Use DQM Rule code to match Contact:** Associate it with rules that find matching records based on the Contact in the imported record.
- **OS: Use DQM Rule code to match Party:** Associate it with rules that find matching records based on the Party ID in the imported record.
- **OS: Use DQM Rule code to match Party Site:** Associate it with rules that find matching records based on the Party Site ID in the imported record.
- **OS: Use DQM Rule code to match Person:** Associate it with rules that find matching records based on the Person in the imported record.
- **OS: Default Resource ID Used for Sales Lead Assignment:** Set it to the resource who will handle any leads that are not assigned to any current territory.

Creating Valid Lookup Codes

Lookup codes map to drop-down lists in the User Interface. The `SOURCE_SYSTEM` lookup type identifies the source of the leads. For example, lead sources could be from a marketing campaign or a partner referral.

The `SOURCE_SYSTEM` lookup type categorizes the leads in the system, and helps you to track them. Seeded values in the `SOURCE_SYSTEM` lookup type are Interaction, Marketing, New, Referral, Sales_Campaign, Store, and User.

Navigation: Log in to Oracle Forms with the Oracle Sales Administrator responsibility and navigate to Oracle Sales Setup > Lookup Codes > Sales and select View > Query By Example > Enter.

Notes

- **Meaning:** The meaning is displayed as one of the values in the drop-down list. For example, the Meaning 'Yes' is displayed for Code Y. The code is stored in a hidden field.
- **Description:** The description along with the meaning gives more information about your lookup code.
- **Tag:** The tag can be used to categorize lookup values. This field is optional.

Setting up Territories

A territory refers to the geographical location of a lead and a sales team. Setting up a territory is important so that the lead is assigned to the right sales team in the correct geographical location.

Create territories in the Oracle Sales and TeleSales node on the territory setup form of

Territory Manager. Territory Manager is part of the CRM Foundation module.

Territory Rule Refresh Concurrent Program

Run the Territory Rule Refresh concurrent program. This concurrent program builds the API that returns the winning territories which are defined in territory setup. Run the program 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. See *Oracle Territory Management Implementation Guide*.

Loading the Import Interface Tables

Load the `AS_IMPORT_INTERFACE` table before running the Import Sales Lead concurrent program

AS_IMPORT_INTERFACE (mandatory): 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.

The following are auxiliary tables. Load data into these tables using a custom program.

- **AS_IMP_LINES_INTERFACE (optional):** This interface table can be used to hold lead lines information, in case you have more than five line items for a lead.
- **AS_IMP_CNT_ROL_INTERFACE (optional):** This interface table is used to hold contact roles information to be imported.
- **AS_IMP_CNT_PNT_INTERFACE (optional):** 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 (optional):** This interface table is to store the flexfields values.

Running the Import Sales Lead Concurrent Program

The Import Sales Lead concurrent program must be scheduled to run at particular intervals. As a result of the concurrent program, unique leads are stored in the `AS_SALES_LEAD` table, and if any of these lead records are unique to the TCA database, they are added to it.

Use the following details to run the Import Sales Lead concurrent program.

Prerequisite: Complete all tasks covered in Before Running the Concurrent , page 3-14 Program.

Responsibility: Oracle Sales Administrator

Parameters:

- Lead Source System - NEW

- Show Debug Message - N
- Batch Id - Batch number if you have imported leads in batches

Schedule: Periodically

For the steps to run the concurrent program, see Running Concurrent Programs, page C-1.

Import Sales Lead Concurrent Program Parameters

The following table lists the parameters for the Import Sales Lead concurrent program.

Import Sales Lead Concurrent Program Parameters

Parameter	Req?	Lookup	Remarks
Lead 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 ViewLog 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.

Parameter	Req?	Lookup	Remarks
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.

Note: The Import Sales Lead concurrent program validates currency codes from the FND_CURRENCIES table while on the HTML UI, the currency codes are picked from the AS_LOOKUP table (lookup type = REPORTING_CURRENCY). The currency codes in both the places must be synchronized to import a lead successfully. If the currency code is not found in the FND_CURRENCIES table, then the currency value set in the JTF_Profile_Default_Currency profile is used.

Important: Users must run the full synchronization program after running the Import Sales Leads concurrent program.

Enhancing the Performance of the Import Sales Lead Concurrent Program

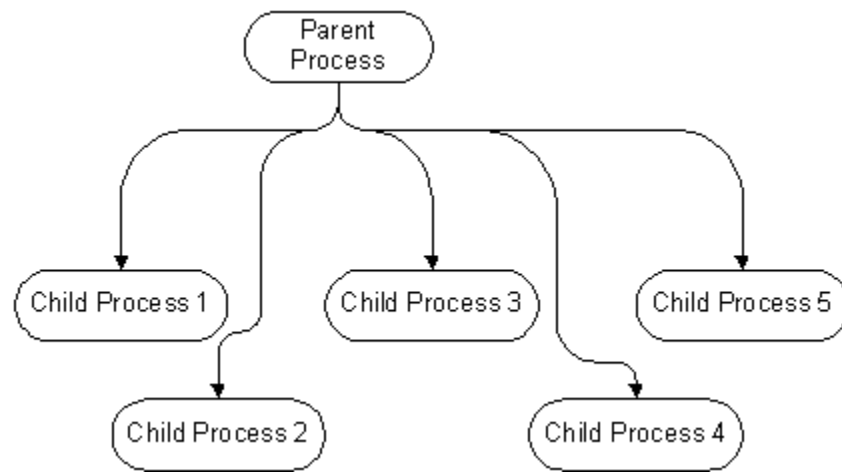
The Import Sales Lead concurrent program processes a number of records in a batch which may be time consuming. The performance of the concurrent program has been improved by running multiple concurrent programs in parallel - each processing fewer number of records.

Parallel Lead Import

In order to improve the performance, the Import Sales Lead concurrent program, itself being the parent, spawns multiple child requests to process the imported records in the AS_IMPORT_INTERFACE table. Since the child requests run in parallel, significant performance improvement is achieved. The concurrent program raises the Lead Import - Pre event before spawning child processes and then raises Lead Import - Post event after all the child processes are complete. For more information on the Pre and Post events, see Seeded Business Events, page 3-49.

The Import Sales Lead concurrent program splits into a parent process and multiple child processes. Figure: Parent and Child Processes, page 3-21 illustrates the relationship between the parent and child processes.

Parent and Child Processes



The OS:Minimum Number of Records for Parallel Processing in Lead Import profile governs the number of child processes that are spawned. For each n number of records in the AS_IMPORT_INTERFACE table, a new request is placed. The default value for this profile is 400.

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, and contact point before creating new records. However, the DQM logic has one limitation.

While importing leads, if the concurrent program creates new records such as Party, Contact, Party Site and Contact Points, the new entries are not reflected in the DQM staging schema. 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 DQM Synchronization concurrent program must be run after the first import and before the next import.

However, lead deduplication can still fail if there are duplicate leads in a single set of imported records. See Custom User Hook, page 3-54 to avoid this.

Running the DQM Synchronization Program

Use the following details to run the DQM Synchronization Program concurrent program.

Responsibility: Trading Community Manager

Schedule: Once

For the steps to run the concurrent program, see Running Concurrent Programs, page C-1.

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:

- Customer Data Quality - performed by Data Quality Management (DQM).
- Leads Data Quality - performed by the Leads Deduplication rule.

Table: Checking for Data Quality in imported Lead Records, page 3-22 gives the sequence in which the Customer and Leads data quality checks are performed.

Checking for Data Quality in imported Lead Records

Task Performed	Program
Checking for Duplicate Original System , page 3-23Reference	Import Sales Lead concurrent program
Checking for Duplicate Customers	Import Sales Lead concurrent program using DQM
Checking for Duplicate Addresses	Import Sales Lead concurrent program using DQM
Checking for Contacts and Contact Points	Import Sales Lead concurrent program using DQM
Leads Data Quality	Import Sales Lead concurrent program using the Lead Deduplication Rule

Customer Data Quality

The Import Sales Lead concurrent program uses the rule-based DQM tool to identify existence of customer records in the TCA database. It uses customer entities like parties (both organization and person), party site, contacts and contact points information to match a record.

When a lead record is imported, it is important to find if a record for this customer already exists in your database. The DQM program matches the imported record with the records in the TCA database to find a matching customer record. If a match is not found, a customer record is created by the Import Sales Lead concurrent program in the TCA registry database.

The DQM program uses rules to identify a matching record. The rules that are used are dependent upon the profiles that are set. See *Setting Up DQM Match Rules*, page 3-28 and *"Setting Profiles Used by the Import Sales Lead Concurrent Program*, page 3-16.

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

Checking for Duplicate Original System Reference

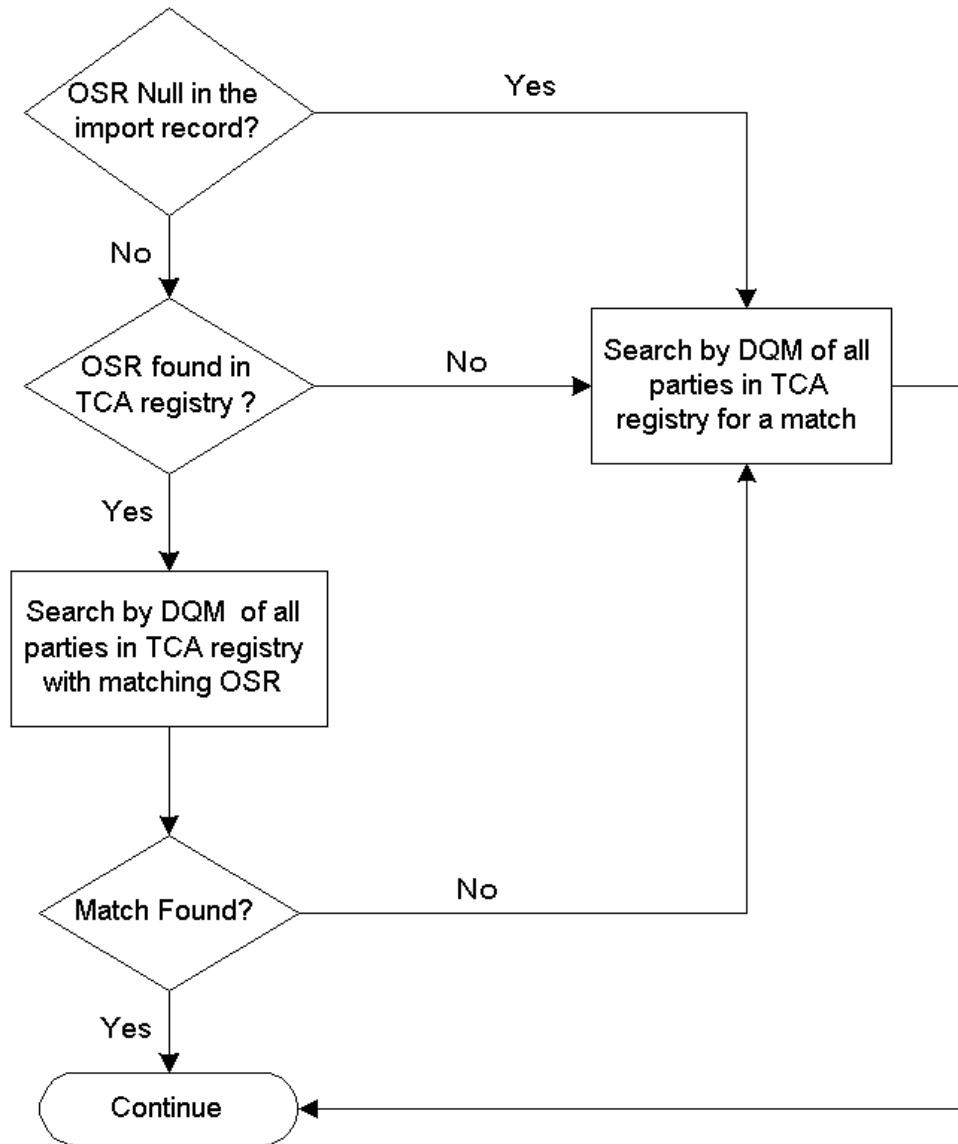
When leads are imported from a list generated by a third-party data source, each entry carries the ID of the party record in that third party database. This ID is referred to as Original System Reference (OSR).

The OSR is carried over to the party record in TCA thus maintaining a correlation between the TCA Party ID recorded on the lead and the party record in the external data source. If the existing party in TCA has a different OSR, the record is updated with the latest one.

OSR is not included in the list of attributes checked by DQM. Therefore, the OSR existence checking is done by the Import Sales Lead concurrent program itself.

Figure: OSR Logic Used by Import Sales Lead Concurrent Program, page 3-24 gives the logic followed to check for duplicate OSR before DQM starts matching party and address.

OSR Logic Used by Import Sales Lead Concurrent Program



If the OSR is available in the import record, then the Import Sales Lead concurrent program uses it to find a set of parties with the same OSR in the TCA database. If a matching set is found, the DQM matching rules are applied on this set alone to find a matching party.

If OSR is available and no matching party set is found or if the OSR itself is not available in the import record, then the DQM matching rules are applied to all the records in the TCA database to find a matching party.

If a matching party is found by using the DQM matching rules, then the matched party_id is reused. If a matching party is not found, the Import Sales Lead concurrent program 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.

Checking for Duplicate Customers

After the Import Sales Lead concurrent program checks for the existence of OSR, the DQM program starts checking for a matching customer in the TCA database. Depending on whether the import record is an organization or a person, the matching rule created to identify duplicate Party or Person is used.

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 (Party) or first name and last name (Person) 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. If the lead is created for Person, the contact information is not passed in.

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.

Checking for Duplicate Addresses

When a matching party is identified, the addresses between the import record and matching party are compared to see if they match as well.

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, the location and party site in the imported record are used.

Checking for Contacts and Contact Points

If the lead is created for Person and an existing party is found, the contact information is checked. The matching rule created to identify duplicate contacts is used. The party ID of the person along with the contact points are passed while calling the `get_matching_contact_points` API. The matched contact points are found in the `HZ_MATCHED_CPTS_GT` table.

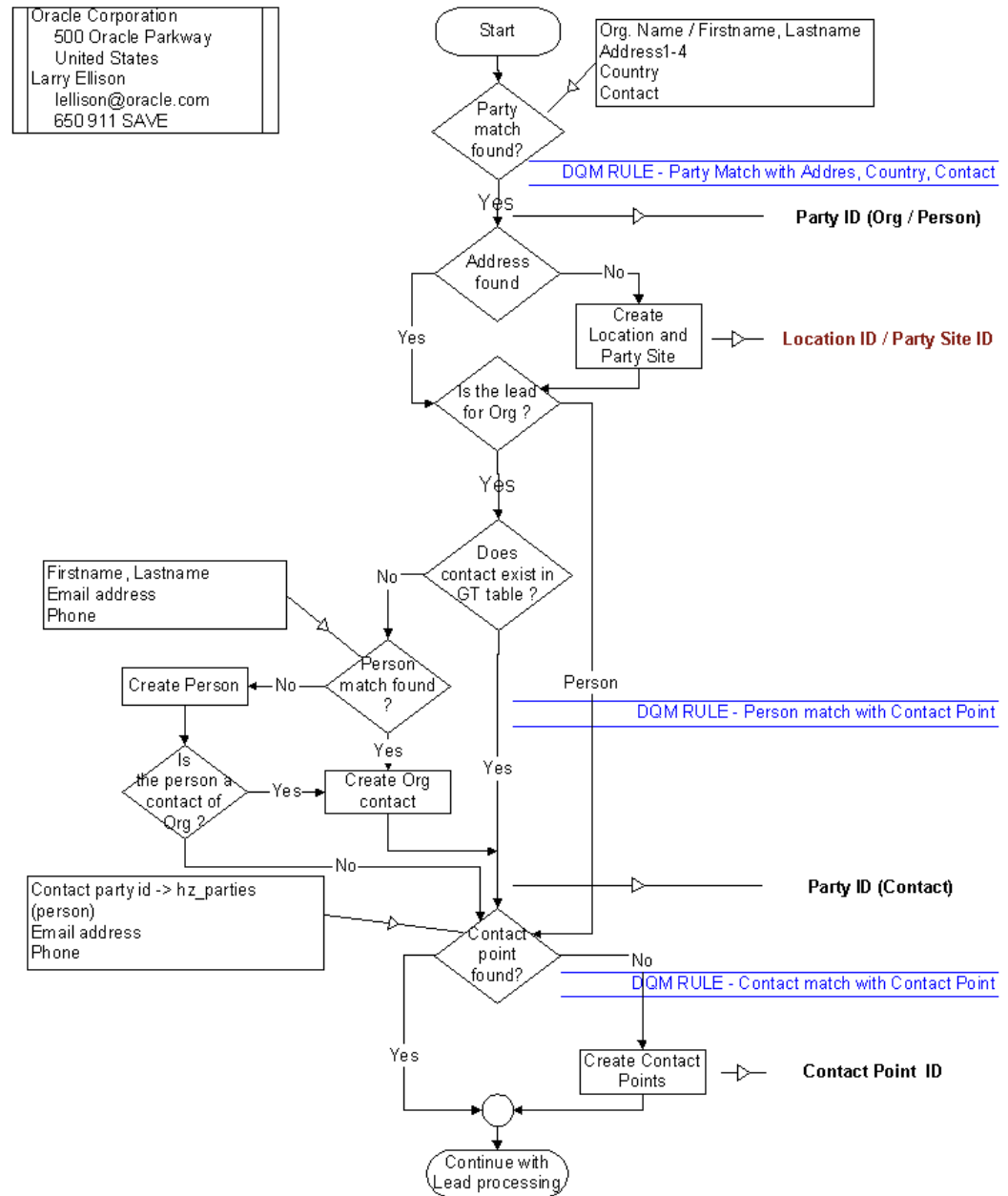
If the lead is created for an Organization and there is no contact Person found for the Organization, a search is performed to find a matching contact Person existing in the database. In order to find a match, the contact person's first name, last name and contact

points are passed in while calling the `HZ_PARTY_SEARCH.FIND_PARTIES` API.

To find a duplicate contact person, you must create a new rule which takes the details of the contact person name and the contact points.

Figure: How DQM Checks for Duplicate Entities, page 3-27 depicts how the DQM program checks for duplicate entities in the database.

How DQM Checks for Duplicate Entities



Setting Up DQM Staging Schema

Use the DQM Staging Program to create the staged schema and *interMedia* indexes. This program applies transformation functions to a portion of the data contained in the TCA registry and generates a separate schema with the transformed and standardized data. The time that the program takes to create the staged schema depends on the size

of your database and the number of attributes and transformation functions that you defined.

Reference

Oracle Trading Community Architecture Data Quality Management User Guide

Use the following details to run the DQM Staging program concurrent program.

Prerequisites: Define attributes and transformation functions in DQM. For sample transformation functions, see *Designing Matching Rules to Detect Duplicate*, page 3-28 Customer or Person.

Responsibility: Trading Community Manager

Parameters:

- Number of Parallel Staging Workers - 1
- Staging Command - STAGE_ALL_DATA
- Continue Previous Execution - No

For the steps to run the concurrent program, see *Running Concurrent Programs*, page C-1

Note: Run the DQM Staging Program every time you add or modify the attributes or the transformation functions.

Designing Matching Rules to Detect Duplicate Customer or Person

You need three DQM match rules for customer or person existence checking during the lead import process. Matches are initially identified using the Acquisition Attributes, and a score is assigned to each match based on the scoring attributes. The party with the highest score is matched with the import lead record.

Use the following as samples while designing DQM matching rules for the Import Sales Lead concurrent 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*. For steps to create the sample rules, see *Setting Up DQM Match Rules*, page 3-28.

Identify Duplicate Parties and Party Sites

The first match rule is used to identify the organization party and address. It is assigned to the profile OS:Use DQM Rule Code to Match Party.

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

attributes.

Figure: Acquisition Attributes for Party Existence Checking, page 3-29 depicts a screen shot of the sample rule with the acquisition attributes for Party existence checking.

Acquisition Attributes for Party Existence Checking

Match Rules

Rule Name: **LEAD_DUP_ORG** Status: **COMPILED** Compile

Description: **Finds duplicates based on Party Name & A** Unlock

Purpose: ☐ Search ☒ Identify Duplicates Copy

Acquisition **Scoring**

Attribute Name	Entity	Type	Filter
<input checked="" type="checkbox"/> Party Type	PARTY	Lookup	<input type="checkbox"/>
<input checked="" type="checkbox"/> Name	PARTY		<input type="checkbox"/>
<input checked="" type="checkbox"/> State	PARTY_SITES		<input type="checkbox"/>
<input checked="" type="checkbox"/> Country	PARTY_SITES	Lookup	<input type="checkbox"/>
<input checked="" type="checkbox"/> Contact Name	CONTACTS	Custom Attr	<input type="checkbox"/>

Attribute Match

☒ Match Any Attribute

☐ Match All Attributes

Transformations

Transformation Name	Description
<input checked="" type="checkbox"/> EXACT	Removes non-alphanumeric characters; f
<input type="checkbox"/>	
<input type="checkbox"/>	

Figure: Scoring Attributes for Party Existence Checking, page 3-30 depicts a screen shot of the sample rule with the scoring attributes for Party existence checking.

Scoring Attributes for Party Existence Checking

Match Rules

Rule Name: **LEAD_DUP_ORG** Status: **COMPILED** [Compile] [Unlock] [Copy]

Description: **Finds duplicates based on Party Name & A**

Purpose: ☐ Search ☒ Identify Duplicates

Acquisition Scoring

Match Threshold: **120** Override Threshold: [] Automatic Merge Threshold: []

Attribute Name	Entity	Type	Score
Name	PARTY		100
Address	PARTY_SITES	Custom Attr	20
Postal Code	PARTY_SITES		20
State	PARTY_SITES		20
Contact Name	CONTACTS	Custom Attr	40

Transformations

Transformation Name	Description	Weight (%)	Exact	Similarity (%)
EXACT STRING	Captures exact string; removes non-alph	100	<input checked="" type="radio"/>	<input type="radio"/>
WR NAMES + CLEANSE	Person and org. name word replacemen	90	<input checked="" type="radio"/>	<input type="radio"/>

Match Rule Name: LEAD_DUP_ORG

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

Purpose: Identify duplicate organizations and addresses.

Table: Existence Checking for Party: Acquisition Attributes, page 3-30, Table: Existence Checking for Party: Matching Attributes, page 3-31, and Table: Existence Checking for Party: Scoring Attributes, page 3-32 list the attributes for this rule.

Existence Checking for Party: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Type	PARTY	EXACT	Catches format errors	Lookup

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Name	PARTY	WR NAMES + CLEANSE	Captures the exact string, removes non-alphanumeric characters, forces upper case, removes vowels, and double letters	-
State	PARTY_SITES	WR STATE	Word replacement	-
Country	PARTY_SITES	EXACT	Captures the exact string, removes non-alphanumeric characters, forces upper case, and catches format errors.	Lookup
CONTACT NAME	CONTACTS	WR+CLEANSE+ REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	Custom Attribute

Existence Checking for Party: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match Any	120	<null>	<null>

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 CLEANSE & REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-	90
Address	PARTY_SITES	20	EXACT	Removes non-alphanumeric characters and forces upper case.	Custom Attribute	100
-	-	-	WR ADDRESS + CLEANSE	Address with word replacement, and removes vowels and double letters.	-	90

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Postal Code	PARTY_SITES	20	EXACT	Removes non-alphanumeric characters and forces upper case.	-	100
State	PARTY_SITES	20	WR STATE	State word replacements		100
Contact Name	CONTACTS	40	EXACT STRING	Captures the exact string, removes non-alphanumeric characters, and forces upper case.	Custom Attribute	100
-	-	-	WR CLEANSE + REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-	90

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.

Identify Duplicate Persons

The second match rule is used to identify the person and address. It is assigned to the profile OS:Use DQM Rule Code to Match Person.

Use the following sample rule to identify existence of party (Person) and addresses. Because 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. The PARTY_SITES entity attributes must be specified because the same rule is used to identify existence of addresses for a specified person.

Figure: Acquisition Attributes for Person Existence Checking, page 3-34 depicts a screen shot of the sample rule with the acquisition attributes for Person existence checking.

Acquisition Attributes for Person Existence Checking

Match Rules

Rule Name: **LEAD_DUP_PERSON** Status: **COMPILED** Compile

Description: **Finds duplicates based on Person Name &** Unlock

Purpose:
☐ Search
☒ Identify Duplicates Copy

Acquisition **Scoring**

Attribute Name	Entity	Type	Filter
Name	PARTY		<input type="checkbox"/>
State	PARTY_SITES		<input type="checkbox"/>
Country	PARTY_SITES	Lookup	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

Attribute Match
☐ Match Any Attribute
☒ Match All Attributes

Transformations

Transformation Name	Description
WR NAMES + CLEANSE	Person and org. name word replacement

Figure: Scoring Attributes for Person Existence Checking, page 3-35 depicts a screen shot of the sample rule with the scoring attributes for Person existence checking.

Scoring Attributes for Person Existence Checking

Match Rules

Rule Name

LEAD_DUP_PERSON

Status

COMPILED

Compile

Description

Finds duplicates based on Person Name &

Purpose

Search

Identify Duplicates

Unlock

Copy

Acquisition

Scoring

Match Threshold

130

Override Threshold

Automatic Merge Threshold

Attribute Name	Entity	Type	Score
Name	PARTY		100
Address	PARTY_SITES	Custom Attr	30
Postal Code	PARTY_SITES		10
State	PARTY_SITES		10

Transformations

Transformation Name	Description	Weight (%)	Exact	Similarity	Similarity (%)
EXACT STRING	Captures exact string; removes non-alpha	100	<input checked="" type="radio"/>	<input type="radio"/>	
WR NAMES + CLEANSE	Person and org. name word replacement	90	<input checked="" type="radio"/>	<input type="radio"/>	

Match Rule Name: LEAD_DUP_PERSON

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

Purpose: To identify duplicate persons and addresses

The following tables list the attributes for this rule.

Existence Checking for Person: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Name	PARTY	WORD REPLACE + CLEANSE + REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-

Acquisition Attributes	Entity	Transformation Function	Description	Type
State	PARTY_SITES	WORD REPLACE + EXACT	Word replacement of Person and Organization names, removes non-alphanumeric characters, and forces upper case.	-
Country	PARTY_SITES	EXACT	Removes non-alphanumeric characters, and forces upper case.	Lookup

Existence Checking for Person: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match All	130	<null>	<null>

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

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
-	-	-	WR CLEANSE & REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-	90
Address	PARTY_SITES	30	EXACT	Removes non-alphanumeric characters, and forces upper case.	Custom Attribute	100
-	-	-	WR ADDRESS + CLEANSE	Word replacement of Person and Organization names, removes vowels and double letters.	-	90
Postal Code	PARTY_SITES	10	EXACT	Removes non-alphanumeric characters, and forces upper case.	-	100

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
State	PARTY_SITES	10	WORD REPLACE + CLEANSE	Word replacement of State name, removes vowels and double letters.	-	100

Identify Duplicate Contacts

The third match rule is used to identify a contact of the organization with relationship type of *Contact of* only. No other relationships types (such as Employee Of, Consumer Of) are considered. This rule is assigned to the profile OS:Use DQM Rule Code To Match Contact.

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

The following figure depicts a screen shot of the sample rule with the acquisition attributes for Contact existence checking.

Acquisition Attributes for Contact Existence Checking

The screenshot shows a 'Match Rules' configuration window. At the top, the 'Rule Name' is 'LEAD_DUP_CONTACT' and the 'Status' is 'COMPILED'. The 'Description' is 'Finds duplicates based on Contact Name &'. Below this, the 'Purpose' section has two radio buttons: 'Search' and 'Identify Duplicates', with 'Identify Duplicates' selected. To the right of the 'Purpose' section are three buttons: 'Compile', 'Unlock', and 'Copy'. Below the 'Purpose' section are two tabs: 'Acquisition' and 'Scoring', with 'Scoring' selected. The 'Scoring' tab contains a table with columns 'Attribute Name', 'Entity', 'Type', and 'Filter'. The table has three rows: 'Name' with 'PARTY' and 'Type' empty; 'Contact Name' with 'CONTACTS' and 'Custom Attr'; and 'Phone Number Flexible Forma' with 'CONTACT_POINTS' and 'Custom Attr'. To the right of the table is an 'Attribute Match' section with two radio buttons: 'Match Any Attribute' and 'Match All Attributes', with 'Match All Attributes' selected. Below the table is a 'Transformations' section with a table with columns 'Transformation Name' and 'Description'. The table has one row: 'WR NAMES + CLEANSE' with 'Person and org. name word replacement'.

Attribute Name	Entity	Type	Filter
Name	PARTY		<input type="checkbox"/>
Contact Name	CONTACTS	Custom Attr	<input type="checkbox"/>
Phone Number Flexible Forma	CONTACT_POINTS	Custom Attr	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

Attribute Match

☐ Match Any Attribute

☒ Match All Attributes

Transformations

Transformation Name	Description
WR NAMES + CLEANSE	Person and org. name word replacement

The following figure depicts a screen shot of the sample rule with the scoring attributes for Contact existence checking.

Scoring Attributes for Contact Existence Checking

Match Rules

Rule Name: **LEAD_DUP_CONTACT** Status: **COMPILED** [Compile] [Unlock] [Copy]

Description: **Finds duplicates based on Contact Name &**

Purpose: ☐ Search ☒ Identify Duplicates

Acquisition **Scoring**

Match Threshold: **110** Override Threshold: [] Automatic Merge Threshold: []

Attribute Name	Entity	Type	Score
Contact Name	CONTACTS	Custom Attr	100
e-mail Address	CONTACT_POINTS		10
Phone Number Flexible Format	CONTACT_POINTS	Custom Attr	10
URL	CONTACT_POINTS		10

Transformations

Transformation Name	Description	Weight (%)	Exact	Similarity (%)
EXACT STRING	Captures exact string; removes non-alpha	100	<input checked="" type="radio"/>	<input type="radio"/>
WR PERSON + CLEANSE	Person name word replacement + Remo	90	<input checked="" type="radio"/>	<input type="radio"/>

Match Rule Name: LEAD_DUP_CONTACT

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

Purpose: To identify duplicate contact and contact points

The following tables list the attributes for this rule.

Existence Checking for Contact: Acquisition Attributes

Acquisition Attributes	Entity	Transformation Function	Description	Type
Party Name	PARTY	WORD REPLACE + CLEANSE + REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-

Acquisition Attributes	Entity	Transformation Function	Description	Type
Contact Name	CONTACTS	WORD REPLACE + CLEANSE + REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	Custom Attribute
Phone Number Flexible Format	CONTACT_POI NTS	EXACT	Removes non-alphanumeric characters and white spaces.	Custom Attribute

Existence Checking for Contact: Matching Attributes

Attribute Match	Match Threshold	Override Threshold	Automatic Merge Threshold
Match All	110	<null>	<null>

Existence Checking for Contact: Scoring Attributes

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
Contact Name	CONTACT S	100	EXACT	Captures the exact string, removes non-alphanumeric characters, and forces upper case	Custom Attribute	100

Scoring Attribute	Entity	Score	Transformation Function	Description	Type	Weight(%)
-	-	-	WORD REPLACE + CLEANSE + REVERSE	Word replacement of Person and Organization names, removes vowels and double letters, reorders first word to the back.	-	90
E-mail Address	CONTACT_POINTS	10	EXACT (E-mail)	Forces uppercase.	-	100
-	-	-	CLEANSE (E-mail)	Removes vowels and double letters.	-	90
Phone Number Flexible Format	CONTACT_POINTS	10	EXACT	Removes non-alphanumeric characters and white spaces.	-	100
URL	CONTACT_POINTS	10	CLEANSE (URL)	Removes non-alphanumeric characters, white spaces, vowels and double letters.	-	100

Note: For existence checking of Party (Organization & 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.

Leads Data Quality

When lead records are imported, they are temporarily stored in the `AS_IMPORT_INTERFACE` table. After the records are processed for lead data quality, unique records are transferred to the `AS_SALES_LEAD` table.

The Import Sales Lead concurrent program uses the Deduplication Rule to identify duplicate lead records between the `AS_IMPORT_INTERFACE` table and the existing leads in the `AS_SALES_LEAD` table. The lead is identified as duplicate based on a set of attributes. If a lead is identified as a duplicate, the status of the lead is marked as Duplicate, and the record is not transferred to the `AS_SALES_LEAD` table. The PV: Run Lead Deduplication Rule profile must be set to Y for the concurrent program to run the deduplication rule.

While creating the deduplication rule, you can check for duplicate leads using the following lead attributes:

Mandatory Attributes

Customer

Optional Attributes Part of Seeded Rule

- Primary Contact
- Customer Address
- Campaign
- Response Channel
- Lead Note/Type

Other Available Optional Attributes

- Product Category
- Project
- Total Budget
- Total Purchase Amount - Product
- Budget Status
- Purchase Timeframe

The Deduplication Rule

The deduplication rule offers you the flexibility to decide the attributes that will identify a duplicate lead for the requirements in your organization. You can also specify the number of days within which the lead should have been created. By default, all leads created in the last 7 days are checked for duplicates.

There is a seeded rule available, which you can customize. This is the algorithm that the seeded rule follows:

1. Look for duplicate customer (last name, first name). If duplicate, continue checking. Else unique lead.
2. Look at contact (name, address, and country). If duplicate, continue checking. Else unique lead.
3. Look at Campaign. If the matching lead(s) are created as a result of the same campaign, continue checking. Else unique lead.
4. Look at all other fields (vehicle response, first lead note). If all are duplicate, lead is duplicate. Else unique lead.

Custom Deduplication Using User Hooks

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

The custom function is executed only if the lead is identified as unique by the Import Sales Lead concurrent program.

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 the `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

Procedure Specification

```
CREATE or REPLACE PACKAGE as_import_sl_cuhk IS
PROCEDURE Is_Duplicate_Lead_Pre(
  p_api_version_number    IN    NUMBER,
  p_init_msg_list         IN    VARCHAR2      := FND_API.G_FALSE,
  p_validation_level      IN    NUMBER        := FND_API.G_VALID_LEVEL_FULL,
  p_commit                IN    VARCHAR2      := FND_API.G_FALSE,
  p_import_interface_id   IN    NUMBER,
  x_duplicate_flag        OUT NOCOPY VARCHAR2,
  x_return_status         OUT NOCOPY VARCHAR2,
  x_msg_count             OUT NOCOPY NUMBER,
  x_msg_data              OUT NOCOPY VARCHAR2
);
END as_import_sl_cuhk;
/
```

Procedure Body

```
CREATE or REPLACE PACKAGE BODY as_import_sl_cuhk AS
PROCEDURE Is_Duplicate_Lead_Pre(
  p_api_version_number    IN    NUMBER,
  p_init_msg_list         IN    VARCHAR2      := FND_API.G_FALSE,
  p_validation_level      IN    NUMBER        := FND_API.G_VALID_LEVEL_FULL,
  p_commit                IN    VARCHAR2      := FND_API.G_FALSE,
  p_import_interface_id   IN    NUMBER,
  x_duplicate_flag        OUT    VARCHAR2,
  x_return_status         OUT    VARCHAR2,
  x_msg_count             OUT    NUMBER,
  x_msg_data              OUT    VARCHAR2
) IS
BEGIN
  /*
  Custom code goes here for lead de-duplication
  Assign:
  x_duplicate_flag = 'Y' if lead is to be marked 'DUPLICATE'
  else
  */

```

```

x_duplicate_flag = 'N'
*/
null;
END;
END as_import_sl_cuhk;
/

```

In Parameters

The four parameters below are standard inputs:

Parameter	Description
p_api_version_number	For the Oracle Sales 12 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. This is 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. This is the default value.

The following parameter does not have a standard input:

Parameter	Description
p_import_interface_id	The import interface identifier. Pass the 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	Indicates the status of the lead. 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.

Customizing the Deduplication Rule

To customize the deduplication rule to your requirements.

- Log in as an administrator and navigate to Administration > Leads > Processing > Deduplication.
- Leads created within the last n number of days are checked for duplicates.
- To add additional attributes for the rule, select lead attributes from the drop-down lists in the Matching Attributes region. These attributes are used to check for duplicates.

The Deduplication Rule Flow

The flow of the deduplication rule is explained in the following table using the settings in the seeded deduplication rule. The Result column in the table gives the outcome for each instance.

Seeded Deduplication Rule Flow

Customer	Vehicle Response	Campaign	Customer Address	Primary Contact Last Name	Primary Contact First Name	Lead Note/Type	Result
Digital Harvester	E-mail	Laptop Cross Sell	500 Oracle Pkwy	Lorna	Bennie	Note1	Master Lead
Digital Harvester	E-mail	Laptop Cross Sell	500 Oracle Pkwy	Lorna	Bennie	Note1	Exact Duplicate Lead
Digital Harvester	E-mail	Laptop Cross Sell	500 Oracle Pkwy	Irvin	Bennie	Note1	Unique Lead (fails first duplicate check)
Digital Harvester	Phone	Laptop Cross Sell	500 Oracle Pkwy	Lorna	Bennie	Note1	Unique Lead (fails Vehicle Response duplicate check)
Digital Harvester	E-mail	Printer Cross Sell	500 Oracle Pkwy	Lorna	Bennie	Note1	Unique Lead (fails Campaign duplicate check)
Digital Harvester	E-mail	Laptop Cross Sell	500 Oracle Pkwy	Lorna	Bennie	New Note1	Unique Lead (fails first note check)

Customer	Vehicle Response	Campaign	Customer Address	Primary Contact Last Name	Primary Contact First Name	Lead Note/Type	Result
Digital Harvester	E-mail	Laptop Cross Sell	500 Oracle Pkwy	Lorna	Bennie	Note1New Note2	Duplicate Lead (second note not checked)

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

Contact Restrictions

The Import Sales Lead concurrent program allows you to set restrictions for the `do_not_phone_flag`, `do_not_fax_flag`, `do_not_email_flag`, and `do_not_mail_flag` flags for the contact (relationship `party_id`), and the `do_not_mail_flag` for the address (`party_site`). If you want to set the restrictions, set these flag values to `Y`.

Contact Points

The Import Sales Lead concurrent program creates the Phone, E-mail, Web, and Fax contact points.

Custom Codes with the Lead Import Program

In addition to the lead import process, you can subscribe to business events that will execute prior to and after the Import Sales Lead concurrent program is run. You can also customize the user hook provided to check for a party in TCA.

- Seeded Business Events, page 3-49
- Custom User Hook, page 3-54

Seeded Business Events

Subscribe to business events when you want additional processing on lead records, or when you want a specific result at the end of the lead import process. You can specify the function codes to run by associating them with a business event, and subscribing to the event.

What is a Business Event?

A business event is an occurrence of any logical event in the application. Examples of business events are Creating a lead, Importing a lead, or Converting a lead to opportunity.

What is Event Subscription?

Event subscription is a pointer to a function code or a workflow. With every business event, you can register one or more subscriptions. When a business event occurs, the subscription(s) associated with the event are called, and the associated function code is executed. If there are more than one subscriptions associated with an event, then the order of execution of these subscriptions is decided from the phase number associated with each subscription.

A Pre function code is executed prior to running the Oracle code and a Post function code is called after the Oracle code is executed. In Oracle Leads Management, the Lead Import Event - Pre and Lead Import Event - Post functions are seeded for the Importing a Lead event.

Before starting to import a batch of records, the Import Sales Lead concurrent program raises the Lead Import Event - Pre function using the `WF_EVENT.Raise ()` call. After the import process for a lead is completed, the Lead Import Event - Post function is executed. The same parameters that are passed to the Import Sales Lead concurrent program are passed to these functions as well. These parameters are `batch_id`, `source_system`, `debug_flag`, and `purge_error_flag`. For information about these parameters, see "Import Sales Lead Concurrent Program Parameters., page 3-19

If there is any exception raised from the function's code, the lead import process for the particular record is terminated, and the `load_status` for the record is marked as `ERROR`.

In general, there can be more than one function subscribed to an event. If there are no functions subscribed, then the control is returned to the calling program without raising an error or exception.

Alternatively, a workflow process can also be associated with an event. For sample subscription codes, see *Sample Function Codes for the Business Event*, page 3-51 Subscriptions. To subscribe to a business event, see the *Oracle Workflow Developer's Guide*.

Error Handling

If there are any exceptions generated from the functions, the Import Sales Lead concurrent program terminates abnormally and any further execution is marked as `ERROR`. The functions must be able to handle these exceptions themselves. If the Business Event system itself raises any errors while calling the `WF_EVENT.Raise()` method, then the error is placed on the `WF_ERROR` queue and a notification is sent to the System Administrator.

Execution Control

If multiple subscriptions are defined for the same event, you can control the order in which the Event Manager executes the subscriptions by specifying a phase number for each subscription. Subscriptions are executed in ascending phase order. For example, you can enter 10 for the subscription that you want to execute first when an event

occurs, 20 for the subscription that you want to execute second, and so on. You can use phases to ensure that different types of actions are performed in the appropriate order, such as executing subscriptions that perform validation before subscriptions that perform other types of processing. If you enter the same phase number for more than one subscription, the Event Manager may execute them in any order, relative to each other. However, the Event Manager will execute that group of subscriptions in their specified place in the phase order, relative to subscriptions with other phase numbers.

You can also use the phase number to control whether the subscription is executed immediately or is deferred. The Event Manager treats subscriptions with a phase number of 100 or higher as deferred subscriptions (asynchronous). Subscriptions with a phase number from 1 to 99 are executed immediately (synchronously). The phase number for a lead import subscription must always be between 1 and 99. The phase number 0 (zero) is reserved for Oracle Workflow seeded subscriptions and should not be used.

Sample Function Codes for the Business Event Subscriptions

Use the following code as a sample to create your pre and post business event functions.

Sample Package Specification

```
CREATE OR REPLACE PACKAGE aml_import_event IS

FUNCTION Pre(p_subscription_guid in raw,

p_event in out wf_event_t) return varchar2;

FUNCTION Post(p_subscription_guid in raw,

p_event in out wf_event_t) return varchar2;

END;
```

Sample Package Body

```
CREATE OR REPLACE PACKAGE BODY aml_import_event IS
```

Sample Function Code - Pre

```
FUNCTION Pre(p_subscription_guid in raw,

p_event in out wf_event_t) return varchar2

IS

l_parameter_list wf_parameter_list_t;
```

```

l_batch_id VARCHAR2(15);

l_source_system VARCHAR2(30);

l_debug_msg_flag VARCHAR2(1);

l_PURGE_ERROR_FLAG VARCHAR2(1);

l_string VARCHAR2(50);

BEGIN

--Get parameters:

l_parameter_list := WF_PARAMETER_LIST_T();

l_parameter_list := p_event.GetParameterList;

l_source_system :=
wf_event.GetValueForParameter('P_SOURCE_SYSTEM',l_parameter_list);

l_batch_id :=
wf_event.GetValueForParameter('P_BATCH_ID',l_parameter_list);

l_debug_msg_flag :=
wf_event.GetValueForParameter('P_DEBUG_MSG_FLAG',l_parameter_list);

l_PURGE_ERROR_FLAG :=
wf_event.GetValueForParameter('P_PURGE_ERROR_FLAG',l_parameter_list);

/*

<-- CUSTOM GOES HERE -->

l_string := substr('Pre-
'||l_batch_id||'-'||l_source_system||'-'||l_debug_msg_flag

||'-'||l_PURGE_ERROR_FLAG,1,50);

commit;

<-- END CUSTOM CODE -->

*/

return 'SUCCESS';

END Pre;

```

Sample Function Code - Post

```

FUNCTION Post(p_subscription_guid in raw,

p_event in out wf_event_t) return varchar2

IS

l_parameter_list wf_parameter_list_t;

l_batch_id VARCHAR2(15);

l_source_system VARCHAR2(30);

l_debug_msg_flag VARCHAR2(1);

l_PURGE_ERROR_FLAG VARCHAR2(1);

l_string VARCHAR2(50);

BEGIN

--Get parameters:

l_parameter_list := WF_PARAMETER_LIST_T();

l_parameter_list := p_event.GetParameterList;

l_source_system :=
wf_event.GetValueForParameter('P_SOURCE_SYSTEM',l_parameter_list);

l_batch_id :=
wf_event.GetValueForParameter('P_BATCH_ID',l_parameter_list);

l_debug_msg_flag :=
wf_event.GetValueForParameter('P_DEBUG_MSG_FLAG',l_parameter_list);

l_PURGE_ERROR_FLAG :=
wf_event.GetValueForParameter('P_PURGE_ERROR_FLAG',l_parameter_list);

/*

<-- CUSTOM GOES HERE -->

l_string := substr('Post-
'||l_batch_id||'-'||l_source_system||'-'||l_debug_msg_flag
||'-'||l_PURGE_ERROR_FLAG,1,50);

commit;

```

```

<-- END CUSTOM CODE -->
*/

return 'SUCCESS';

END Post;

END aml_import_event;

/

```

Custom User Hook

You can execute a custom user hook that will be called from the Import Sales Lead concurrent program before DQM processes each record. This user hook tries to find a `party_id` with an exact customer name match in the `HZ_PARTIES` table in TCA. Calling the user hook from the Import Sales Lead concurrent program is controlled by the `OS:Execute Custom Code from Lead Import` profile. If the value of the profile is `Yes`, then the program is invoked.

Purpose of a Custom User Hook

Consider the following scenario where the custom user hook is used. A batch has two records with the same customer name, and the customer is not recorded in the `HZ_PARTIES` table in TCA. The custom user hook processes the first record in the batch and tries to find an exact string match with `party_name` in the `HZ_PARTIES` table. Because this is a new party, the query will not return any records. Next, DQM is invoked to find a party match. This would also fail because it is a new party. Finally, the Import Sales Lead concurrent program creates a new customer in TCA, and creates a lead against it.

While processing the second record, before calling DQM, the user hook returns the `party_id` of the newly created customer. Hence, DQM processing for the record will be bypassed.

If the user hook is not available, DQM will fail because the synchronization is not done between the DQM Staging Schema and the TCA before the second row is processed. See *Limitation of the Import Sales Lead Concurrent Program*, page 3-21.

On the other hand, if the user hook returns more than one record for the exact customer match, it would mean that the customer was created before this import batch was loaded, and should have been present in the DQM staging area. Therefore, the program should call DQM to find the one right match among the multiple matches.

- The user hook party match program must be properly maintained for it to be effective.

- The user hook party match program can find a matching party only if the customer names in the batch are exactly the same.

Performance Impact

Running the user hook may affect performance because the program queries the HZ_PARTIES table for a party_id with the exact party_name.

Seeded Code for the User Hook

The following is the seeded PL/SQL package that is seeded for the user hook. You can customize it to suit your setup.

Procedure Specification

```
PACKAGE aml_find_party_match_pvt

PROCEDURE main (imp              IN OUT NOCOPY as_imp_int%ROWTYPE,

X_return_status OUT NOCOPY varchar2)

l_party_id  number;

Begin

X_return_status := FND_API.G_RET_STS_SUCCESS;

SELECT party_id

INTO l_party_id

FROM    hz_parties hzp

WHERE   hzp.party_name = imp.customer_name;

Imp.party_id := l_party_id;

UPDATE as_import_interface

SET party_id = l_party_id

WHERE import_interface_id = imp.import_interface_id;

Exception

When NO_DATA_FOUND Then
```

```

l_party_id := NULL;

When TOO_MANY_ROWS Then

l_party_id := NULL;

End main;

```

Purging Staged Lead Records

The AS_IMPORT_INTERFACE table is a temporary location where the imported records are stored before unique records are moved to the AS_SALES_LEAD table. After the records are checked for duplicates, the remaining records in the AS_IMPORT_INTERFACE may not be required. If they are retained in the AS_IMPORT_INTERFACE table, the number of records may affect the performance of the Import Sales Lead concurrent program.

You can delete the records from this table by running the Purge Leads Import Interface Table concurrent program.

Use the following details to run the concurrent program.

Responsibility: Oracle Sales Administrator

Parameters:

- From and To - This date range indicates the dates when the lead was created. When the program is run, all leads that were created in the specified range are deleted. For example, if the From and To Dates are 3-Nov-2003 and 15-Nov-2003, all leads created between the 3rd and 15th of November, 2003 are deleted from the table.
- Load status of records to be purged - Status of the leads that you want to purge
- Debug - Enter Yes if you want to see debug messages.
- SQL Trace - Enter Yes if you want to trace SQL.

Schedule: Once

For the steps to run the concurrent program, see Running Concurrent Programs., page C-1

Linking Interactions to Leads

This chapter covers the following topics:

- Interaction Matching Engine Overview
- Interaction Types and Interaction Scores
- Interaction Score Threshold
- Functioning of the Interaction Matching Engine
- The Mining Behavior of the Interaction Matching Engine
- Linking an Interaction with Existing Leads
- Linking if No Lead Exists for the Interaction
- Running the Concurrent Program
- Setting Up Interaction Types
- Interaction Selection Rule Sets
- Creating an Interaction Rule Set

Interaction Matching Engine Overview

An interaction is a single contact event between a customer or customer system and a resource of the business. An example of an Interaction with a potential customer would occur when marketing campaigns target lists of e-mail addresses for mass mailings or when a person calls a call center to get product information.

The Interaction Matching Engine provides the mechanism to mine and evaluate these customer interactions and responses for sales follow up. Interaction Matching rules can be set up to determine the marketing interactions that must be evaluated to generate sales leads or to enhance the quality of existing leads.

To understand the Interaction Matching Engine and associated concepts, see the following sections:

- Interaction Types and Interaction Scores
- Interaction Score Threshold

Interaction Types and Interaction Scores

Customer interactions are of various types. Examples of interaction type are Event Registration, Survey Completion, Web Visit, and so on.

Each interaction type is associated with a default interaction score that indicates the significance of the interaction. For example, an interaction where the customer tells the call center agent to have a sales representative call back is a very significant type of interaction. This event has an immediate sale potential, and merits a high score. Whereas, an interaction type such as Event Registration may indicate that the customer has some interest in the product associated with the event. However, it does not merit immediate sales attention, and hence is worthy of a low score.

The following table lists the seeded interaction types and the default interaction scores considered by the Interaction Matching Engine. The Seeded Interaction Rule uses these interaction scores. To set up additional interaction types, see Setting Up Interaction Types.

Seeded Interaction Types and Scores

Interaction Type	Description	Default Interaction Score
Survey Callback	Customer has requested a call back from the company.	10
Event Registration	Customer has registered for an event organized by the company.	3
Survey Completion	Customer has completed taking a survey posted by the company.	3
Web Collaboration	Customer has interacted with the company through the Web.	2
Inquired about an item from Customer	Customer has enquired about an item of the company.	3

Interaction Type	Description	Default Interaction Score
Web Advertisement Visits	Customer has clicked on the company's advertisement on the web for details.	1
Web Offer Visits	Customer has clicked on the company's offer on the web for details.	1
E-mail Clickthroughs	Customer has clicked on a URL in an e-mail sent by the company.	1

Note: The scores for the interaction types can be changed when you are setting up rule sets in the Interaction Matching Engine. For more information, see *Interaction Rule Sets*, page 4-7

Interaction Score Threshold

The Interaction Score Threshold (IST) is a set value against which the sum of all the interaction scores for a customer are matched. When the sum is equal to or greater than the IST, a lead is generated from the set of interactions. Each time an interaction is linked to a customer, the sum of interaction scores are evaluated against the IST.

For example, assume that the IST is set to 20. A customer interaction with a score of 5 is chosen by the Interaction Matching Engine. However, no lead is available for the corresponding customer. When another interaction for the same customer with a score of 10 is recorded, both the interaction scores are added and compared with the IST. In this case, the score does not exceed the IST yet. A lead is created from these interactions when the sum of all the interaction scores equal or exceed the IST.

The IST value is set in the `OS:InteractionScoreThreshold` profile. The default value for this profile is 20.

Functioning of the Interaction Matching Engine

The Interaction Matching Engine functions in two key ways:

- If no lead exists for a customer, new marketing interactions for the customer are mined and used to generate a lead for sales follow up.
- After a lead is generated, or if a lead already exists for a customer, new marketing

interactions for the customer are matched with the lead to improve its quality.

To understand the working of the Interaction Matching Engine, see the following sections:

- The Mining Behavior of the Interaction Matching Engine
- Linking an Interaction with Existing Leads
- Linking if No Lead Exists for the Interaction
- Running the Concurrent Program

The Mining Behavior of the Interaction Matching Engine

The Interaction Matching Engine is driven by a rule set that has an activation date range. A concurrent program (Run Interaction Matching Engine to Match or Create Leads) runs the Interaction Matching Engine at scheduled times. Based on the rule set, interactions are evaluated for their sales potential.

The Guards in the rule set support the Country and Campaign (Source Code) attributes. The attributes are ANDed during evaluation. You can also opt to capture all interactions without any attributes. The Rules in the rule set support the Interaction Type and the Interaction Score.

If the Interaction Matching Engine rules sets are modified, they are applicable only for the new interactions captured since the last time the engine ran. They are not used to match interactions that were processed before the rule sets changed.

All the matched interactions are tracked and displayed with the timestamp on the lead. You can also browse through all the responses that contributed to the lead. Review and periodically archive interactions that are not matched with leads or other interactions. You can view the interactions attached to a lead from the history details for the lead. For more information, see the *Oracle Field Sales User Guide*.

Note: Interactions with a Guest User's Party ID are not used for matching.

The mining behavior can be customized with business-specific logic that renders certain interactions in specific contexts valuable in determining sales value.

Mining by Source Code

The significance of an interaction may depend on the specific promotion associated with it. For example, an Event Registration for a strategic CEO level event is more significant than an Event Registration for a technical web seminar. Hence, the score of the interaction also depends on the promotion code associated with the interaction. For this type of mining, you can create specific rules that will leverage the source code as a guard.

In addition to the source code attribute, the country attribute can also be used as a guard to define interaction rules.

Linking an Interaction with Existing Leads

When an interaction is captured for linking, the Interaction Matching Engine first checks all the existing sales leads to match the interaction with a lead. The interaction is matched with leads based on the following criteria:

- **Creation Date Range** - This is the date range when the lead is created. For example, the interaction is compared with all leads created in the last 60 days.
- **Lead Status** - The status of the lead must be `Open`.
- **B2B and B2C** - The Interaction Matching Engine supports matching of Business to Business (B2B) as well as Business to Customer (B2C) leads. For B2B situations, the Customer (Organization) ID recorded in the interaction must match the Customer ID recorded in the lead. Optionally, the contact recorded in the interaction (if any) must match one of the contacts recorded on the lead. The profile OS: Interaction Matching Engine Matches Contact for B2B lead determines whether the primary contact should also be matched for B2B leads.

If more than one lead matches a given interaction, then a lead is chosen by its progress in the sales pipeline.

- Based on the rating of the lead, the lead with the better rating is chosen.
- Based on the creation date of the lead, the lead created most recently is chosen.

Lead Reevaluation

When an interaction is linked to a lead, the lead is reevaluated only if both these conditions are satisfied:

- One of the attributes in the Qualifying, Rating, or Channel Selection engine rule sets is Interaction Score.
- The OS: Auto Run Lead Engines While Update profile is set to `Yes`.

The Interaction Score can be leveraged in the Rating Engine to upgrade the lead value, along with any other relevant attributes. When new (additional) interactions are linked, the score of the new interaction will augment the rating of the lead.

When an unqualified lead is appended with an interaction, and the lead's Interaction Score goes up, the lead is processed by the qualification, rating, and channel selection engines again.

Linking if No Lead Exists for the Interaction

If a matching lead is not found for an interaction, the interaction is evaluated, and if the interaction score is equal to or greater than the IST score, then a lead is generated.

If the interaction score does not meet the IST, the interaction is linked with other unique interactions for the same party. All matching interactions for the prospect within the allowed time frame are scored and evaluated as an aggregate score. This aggregate score is constantly reevaluated against the IST, and a lead is created when appropriate. All interactions used to generate the lead are appended to the lead.

Lead Generation

When a lead is generated from a set of interactions, the lead is assigned to a sales channel with the following details:

- The name of the lead. This is taken from the interactions from which the lead was generated. The name could be the Contact first name or last name; or the name of the Organization.
- The customer and contacts are populated with the parties in the interactions. The party details can also be linked for contact point information.
- The source code is either the source code of the most recent interaction, or the source code of the interaction with the highest score. All the other source codes are recorded as lead lines along with the associated products.
- If present, all products and related offers associated with the interaction or source code are recorded as lead lines.

Running the Concurrent Program

The Run Interaction Matching Engine to Match or Create Leads concurrent program runs the Interaction Matching Engine.

Use the following details to run the Run Interaction Matching Engine to Match or Create Leads concurrent program.

Prerequisite: Interaction Matching Rules must be set up.

Responsibility: Oracle Sales Administrator

Parameters:

- Debug - Enter Yes if you want to see debug messages.
- SQL Trace - Enter Yes if you want to trace SQL.

Schedule - Once

For the steps to run the concurrent program, see *Running Concurrent Programs.*, page C-1

Setting Up Interaction Types

All customer interactions can be associated with a specific type. Examples of interaction types are Event Registration, Survey Completion, and Web Visit. You can set up interaction types that you want associated with the interaction rules.

The first few interaction types are seeded, and cannot be selected. You can enable or disable a seeded interaction type, and also change the description and meaning, if required.

Notes

- **Enabled:** An interaction type that is not enabled is not visible while creating rules for the Interaction Matching Engine rule sets.
- **Meaning:** The term entered in the Meaning column will appear as the Interaction Type when you create an interaction rule set.

Interaction Selection Rule Sets

The interaction rules sets identify the interactions that are appropriate and relevant to be mined for follow up purposes.

The Linking rules can be customized to:

- Find the particular business interactions that merit linking such as completed surveys, event registration, web collaboration, web visits, or inbound calls.
- Identify worthy business interactions that merit lead generation using promotion source codes.

If more than one rule set with equal precedence win, the rule set with the highest score is selected.

You can view the interactions attached to a lead from the history details for the lead. For more information, see the *Oracle Field Sales User Guide*.

You can track the effectiveness of the interaction rule sets by viewing the Lead Linking Report. For more details, see Lead Linking Report.

Creating an Interaction Rule Set

An interaction rule set contains a guard and rules. The Country and Campaign attributes make up the guard options. Rules can be created using Interaction Types and Scores. You can view the details of the seeded interaction rule set called Seeded Interaction Rule.

When an interaction matches with the rule set conditions, a score is assigned to the interaction based on the type of the interaction. When the score reaches a threshold, a lead is created from the interaction(s).

While creating a rule set, it is in the Draft status by default. Use the following procedure to create an interaction rule set.

Navigation: Log in as an administrator and navigate to Administration > Leads > Processing Rules > Interaction Selection.

Notes

- **Start and End Dates:** Dates between which the rule set is valid.
- **Precedence:** Each rule set can have a different precedence, to define the order of importance for evaluation (where 100 is higher than 1).
- **Match All Interactions:** If you do not want to filter any interactions, select Match all Interactions. To filter interactions based on their attributes, select Match Interactions based on Guard.

Guards define the domain to which the rule set applies such as campaign-specific or country-specific.

- **Match Interactions Based on Guard:** To filter interactions based on their attributes, select Match Interactions based on Guard. Guards define the domain to which the rule set applies such as campaign-specific or country-specific.
- **Add Attributes:** Add attributes to the rule set.
- **Rule:** Add rules to the rule set.
- **Interaction Type:** From this drop-down list, select a type and enter a value for the score.

Processing Leads

This chapter covers the following topics:

- Leads Processing Engine Overview
- The Qualification Engine
- The Rating Engine
- Setting Up Ratings
- The Channel Selection Engine
- Setting Up Channels
- Best Practices
- Rule Flows
- Purging Unqualified Leads
- Setting Up Lead Assignments
- Setting Up Immature Lead Assignment
- Routing Leads Using a User Hook
- Setting Up Lead Status
- Using Custom Attributes
- Setting Up Time Frames
- Customizing Time Frames

Leads Processing Engine Overview

The Leads Processing engine comprises 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 The Qualification

Engine, page 5-3.

- 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 activities for effective cost and resource management. See The Rating Engine, page 5-4.
- The Channel Selection Engine channels the leads 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 within the company and indirect when routed to partners or value added distributors. See The Channel Selection Engine, page 5-5.

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

Rule Set Details

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

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.

After the rule sets are bucketed into different groups, the Precedence of each rule set is used to determine the order of evaluation. For example, if the attributes of a lead are matched with a Country-specific and a Campaign-specific rule set, by assigning the Country-specific rule set a higher precedence, this rule set is evaluated before the Campaign-specific rule set.

For precedence, 100 is higher than 1.

Reports

The Leads Processing History and Rule Performance Reports help you analyze the effectiveness of the rule sets in an engine. To troubleshoot rule set issues, see the Rule Diagnostics Report, page 7-2.

Rule

Using the guards, after 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 rated A, or channelized to Direct Sales.

Rules are evaluated in precedences from 1-n, where 1 is evaluated first. 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 the highest ranked result, wins.

Attributes

Add attributes to the rule set.

The Qualification Engine

When a lead is run through the Leads Processing Engine, it is processed by the Qualification Engine (QE) first. In this release, the QE has two primary functions:

- **Qualifying a Lead:** A lead can be qualified when the attributes of the lead indicate interest in the purchase of a product. Some of these attributes can be: the lead has attended a product event, or the budget has been approved, or the purchase time frame is less than a year. Such details suggest that it is worthwhile for a sales team to follow up with the lead, and possibly arrive at a sale.
- **Disqualifying a Lead:** A lead can be disqualified when the attributes indicate that the lead is not genuinely interested in the purchase of a product. If the lead belongs to the Student category, or has an incomplete or incorrect e-mail address, or belongs to a competitor company, such leads can be filtered. They need not be pursued to a sales closure.

At the end of the qualification process, a lead may be qualified or disqualified. All qualified leads are routed to the Rating engine. All disqualified leads are routed to the Channel Selection engine.

How Does the Qualification Engine Work?

Each rule set is a grouping of rules. The rule set is defined by its Guard. The rules with a guard define the criteria and outcome.

The QE identifies the rule sets that can evaluate the lead by comparing the guard values in the rule set with the attributes of the lead being processed. For example, if the lead has Campaign A as an attribute, the QE searches for a rule set with Campaign A as a guard value.

After 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.

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

If no rule sets win, the lead is set to the value specified in the OS: Default Qualified Flag for Lead Qualification Engine profile. The default value for this profile is No.

In the case where two rule sets win, and one rule set qualifies the lead, and the other disqualifies, the lead is qualified by the QE.

The Rating Engine

After a lead is qualified by the Qualification Engine, it is processed by the Rating Engine (RE). The RE prioritizes the leads based on their attributes, and assigns them ratings. The rating helps the sales representative decide the importance of a lead, and accordingly follow up with the lead.

How Does the Rating Engine Work?

When a lead is run through the RE, 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, if the lead has Campaign A as an attribute, the RE looks for rule sets with Campaign A as a guard value.

After a matching Rating rule set is identified, the engine starts evaluating the rules for each rule set, starting with the rule set of the highest precedence. The RE evaluates the rules in the order of evaluation assigned and stops when it finds a rule that matches the lead.

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

If more than one rule set with equal precedence win, the highest rating is selected.

If no rule sets win, the rating set in the OS: Default Rating for Lead Rating Engine profile is used. The default value for this profile is Cold Lead. This profile value must not be set to blank.

Setting Up Ratings

Ratings must be set up before rating rule sets are defined. Use this procedure to set up ratings.

Navigation: Log in as an administrator, and navigate to Administration > Leads > Setup > Rating.

Notes

- **Grade:** Enter a grade for the lead. The lead is rated with this grade.
- **Precedence:** The value you enter sets the precedence for the grade. The precedence is used when 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.

- **Enabled:** Select the Enabled check box to include this grade to rate leads.

Note: When you create a rule set, the grades that you just set up may not appear in the drop-down lists. You must restart the apache to reload data from the database.

The Channel Selection Engine

The Channel Selection Engine (CSE) is responsible for distributing leads to the appropriate teams for further follow up and action. Based on channel selection rules and the lead attributes, a lead is assigned to a channel.

Examples of channels are Inside Sales, Direct Sales, Indirect Sales, and Partner.

How Does the Channel Selection Engine Work?

The CSE is similar to the Qualification and Rating engines. When a lead is run through the CSE, the engine first identifies the correct Channel Selection 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, if the lead has United States as its Country attribute, the CSE looks for rule sets with United States as a guard value for Country.

After the matching channel selection rule sets are identified, the engine starts evaluating the rules, starting with the rule with the 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 assigned to the selected channel and the winning rule set is logged into a history table for analysis.

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

Setting Up Channels

Channels must be set up before channel selection rule sets are defined. Use this procedure to set up channels.

Navigation: Log in as an administrator, and navigate to Administration > Leads > Setup > Channel.

Notes

- **Indirect:** Select this check box if this is an indirect channel.
- **Precedence:** 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.

Note: When you create a rule set, the channels that you just set up may not appear in the drop-down lists. You must restart the apache to reload data from the database.

Best Practices

Some best practices that you can use when you are working with the Leads Processing Engines are listed below:

- The Qualification Engine must not be used to both qualify and disqualify leads.
- If you use the Qualification Engine only to disqualify leads, it is recommended that you migrate your existing qualification rule sets to the Rating Engine. This ensures that the disqualified leads do not appear in sales representative queues.
- Create your channel selection rule sets in such a manner that all the disqualified leads are isolated and treated differently. A sales representative may work on them and if found suitable, change the Qualified flag to Yes.
- Create rule sets using the Campaign attribute. By doing so, leads generated from a specific campaign are processed by the specific rule sets. This helps you identify leads generated for a campaign, and you can track their progress in the sales pipeline.

For example, you want to track leads for campaign VisionVideos. When you create your rule sets, use Campaign as a guard and select this campaign. Hence, all leads that result from the VisionVideos campaign are processed by this rule set.

Rule Flows

This Rule Flows report allows you to query across the Qualification, Rating, and Channel Selection engines for rule sets based on certain guard values.

This report supports multiple rule set groupings in the Rules Engine setup to track rule sets across processing flows. The rule sets are grouped based on the engine type. For example, Qualification rule sets are displayed first, then Rating, followed by the Channel Selection rule sets.

Search By

- Date Range

- Guard values - Country, Campaign, Custom Setup, and Product Category
- Processing Stage
- Rule Set Status

Purging Unqualified Leads

You can remove unqualified leads from the `AS_SALES_LEADS` table by running a concurrent program. Based on the following conditions, leads are deleted:

- The lead is unqualified, and it is not converted to opportunity
- The status of the lead is the same as the value in the OS: Default Status for Leads profile.

Unqualified leads that have been converted to opportunity will not be deleted by this program.

Note: After leads are purged from the system, any Trend reports set up in Oracle Daily Business Intelligence (DBI) or a custom application will be affected.

Use the following details to run the Purge Unqualified Sales Leads concurrent program.

Responsibility: Oracle Sales Administrator

Parameters:

- From and To Date - Enter a date range indicating the leads created during this period.
- Debug - Enter `Yes` if you want to see debug messages.
- SQL Trace - Enter `Yes` if you want to trace SQL

Schedule: Once

See Section Running Concurrent Programs, page C-1 for the steps to run the concurrent program.

Setting Up Lead Assignments

Various system profiles can be set up to assign resources to leads based on the Leads Processing Engine results. Depending on the requirements in your organization, you can also define custom functions to route leads to appropriate resources.

Setting Up Automatic Lead Assignment

You can set up the application to automatically assign resources to a lead whenever an agent or salesperson creates or updates the lead. This is achieved by assigning values to selected profiles.

- **OS: Assign New Lead**

To enable automatic lead assignment, set the value of this profile to **N**. This is the default 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.

- **OS: Auto Convert Lead to Opportunity**

If this profile is set to **Y**, an opportunity is created for all qualified indirect leads, and the partner matching workflow is launched.

- **OS: Lead Incubation Channel**

All immature leads are assigned to a particular channel. The channel is decided by the value in this profile. The lead owner of an immature lead is determined by immature lead assignment.

- **OS: Default Resource ID Used for Sales Lead Assignment**

Set this profile to a resource who will handle any lead that is not matched with any territory. If this profile is not set, the lead is assigned 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.

- **OS: Calendar Setting Required for Assignment**

Set this profile to **Yes** if the territories in your organization use agent availability as one of the criteria for assigning agents. This enables the automatic assignment of lead owners based on availability. This profile is set to **No** by default.

Apart from setting this profile, you must also make sure that each resource has a calendar set up for them.

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

See Setting System Profile Options, page A-1 to set values to these profiles.

Setting Up Immature Lead Assignment

An immature lead is a lead that is not yet ready for a sales representative to spend time on. It is a low quality and low grade lead that needs to be matured by the marketing team before it is assigned to a sales team.

The Channel Selection engine assigns all immature leads to a specific channel, such as an Immature channel. This channel is decided by the value in the OS: Incubation Channel profile. When a lead is assigned to the immature channel, the owner is decided by the value in the OS: Default Lead Marketing Owner profile.

If the OS: Default Lead Marketing Owner profile is not set, the Territory Assignment program assigns all immature leads to the resources identified to act on immature leads.

The Maturation Assignment page in the Administration > Leads tab can be used to provide information about the territory assignment setup.

Note: The Maturation Assignment page will be obsolete from the next release. Use the Sales Channel to group the immature leads, and the Territory Assignment program will assign them to an appropriate resource based on the Channel Qualifier.

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

Navigation: Log in as an administrator, and navigate to Administration > Leads > Processing Rules > Maturation Assignment.

Notes

- **Owner:** Select this check box to indicate that a sales person must be the owner of the lead assigned.

Routing Leads Using a User Hook

You can implement custom rules for lead assignment by implementing the Lead Routing Engine user hook.

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

Procedure Specification

```
PROCEDURE Get_Owner_Pre(  
  
    p_api_version_number    IN    NUMBER,  
  
    p_init_msg_list         IN    VARCHAR2 := FND_API.G_FALSE,  
  
    p_validation_level      IN    VARCHAR2 := FND_API.G_VALID_LEVEL_FULL,  
  
    p_commit                IN    VARCHAR2 := FND_API.G_FALSE,  
  
    p_resource_id_tbl       IN    AS_LEAD_ROUTING_WF.NUMBER_TABLE,  
  
    p_group_id_tbl          IN    AS_LEAD_ROUTING_WF.NUMBER_TABLE,  
  
    p_person_id_tbl         IN    AS_LEAD_ROUTING_WF.NUMBER_TABLE,  
  
    p_resource_flag_tbl     IN    AS_LEAD_ROUTING_WF.FLAG_TABLE,  
  
    p_sales_lead_rec        IN    AS_SALES_LEADS_PUB.SALES_LEAD_Rec_Type,  
  
    x_resource_id           OUT   NUMBER,  
  
    x_group_id              OUT   NUMBER,  
  
    x_person_id             OUT   NUMBER,  
  
    x_return_status         OUT   VARCHAR2,  
  
    x_msg_count             OUT   NUMBER,  
  
    x_msg_data              OUT   VARCHAR2  
  
)  
  
IS  
  
    l_resource_count        NUMBER;
```

```

BEGIN

-- Standard Start of API savepoint

SAVEPOINT GET_OWNER_PRE_PVT;

-- Standard call to check for call compatibility.

IF NOT FND_API.Compatible_API_Call ( l_api_version_number,

p_api_version_number,

l_api_name,

G_PKG_NAME)

THEN

RAISE FND_API.G_EXC_UNEXPECTED_ERROR;

END IF;

-- Initialize message list IF p_init_msg_list is set to TRUE.

IF FND_API.to_Boolean( p_init_msg_list )

THEN

FND_MSG_PUB.initialize;

END IF;

-- Initialize API return status to SUCCESS

x_return_status := FND_API.G_RET_STS_SUCCESS;

-- Api body

l_resource_count := p_resource_id_tbl.count;

IF l_resource_count > 0

THEN

x_resource_id := p_resource_id_tbl(1);

```

```

x_group_id := p_group_id_tbl(1);

x_person_id := p_person_id_tbl(1);

ELSE

x_resource_id := NULL;

END IF;

-- END of API body

-- Standard check for p_commit

IF FND_API.to_Boolean( p_commit )

THEN

COMMIT WORK;

END IF;

-- Standard call to get message count and IF count is 1,get message info

FND_MSG_PUB.Count_And_Get

( p_count          => x_msg_count,

p_data            => x_msg_data );

END Get_Owner_Pre;

END AS_LEAD_ROUTING_WF_CUHK;

```

In Parameters

The following table lists the standard input parameters.

Standard Input parameters

Parameter	Description
p_api_version_number	For the Oracle Sales 12 application, this is set to 2.0.

Parameter	Description
p_init_msg_list	Should the message stack be initialized? By default, this is set to FND_API.G_FALSE.
p_validation_level	The validation level of the pass-in value. By default, this is set to FND_API.G_VALID_LEVEL_FULL.
p_commit	Should a commit be issued for the whole API at the end? By default, this is set to FND_API.G_FALSE.

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

The following table lists other input parameters.

Other Input Parameters

Parameter	Description
p_resource_flag_tbl	<p>This parameter specifies the source of the resource:</p> <ul style="list-style-type: none">• D: Default resource from the OS: Default Resource ID used for Sales Lead Assignment profile.• L: Login user.• T: Territory definition. <p>If the sales lead matches any territory, the above parameters will include all the resources returned from the territory engine and the p_resource_flag_tbl will be all T.</p> <ul style="list-style-type: none">• If the sales lead does not match any territory, and the OS: Default Resource ID used for Sales Lead Assignment profile 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)=L.• p_resource_flag_tbl(2)=L• If the sales lead does not match any territory, and the OS: Default Resource ID used for Sales Lead Assignment profile is not set:• p_resource_id_tbl(1), p_group_id_tbl(1)• p_person_id_tbl(1)=L

Parameter	Description
	<ul style="list-style-type: none"> 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 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, they set the sales lead owner.

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

For example, 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 will be assigned as the owner of the sales lead.

The following table lists the standard output parameters.

Standard Output Parameters

Parameter	Definition
x_return_status	<p>The return status.</p> <p>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.</p>
x_msg_count	<p>The message count.</p> <p>Call FND_MSG_PUB.Count_And_Get to get the message count and messages.</p>

Parameter	Definition
x_msg_data	<p>The messages.</p> <p>Call FND_MSG_PUB.Count_And_Get to get the message count and messages.</p>

Setting Up Lead Status

Some lead statuses are seeded in the application. They are New, In Progress, Converted to Opportunity, Dead Lead, and Loss.

Use the following procedure to define alternate statuses.

Navigation: Log in as an administrator, and navigate to Administration > Sales > Opportunity > Status Code.

Notes

- **Status Code:** You cannot view the status code. It is for internal use only.
- **Meaning:** You can view this in the drop-down list when you select a lead status.
- **Win Loss Indicator:** Select Neither in this region. The other options are reserved for opportunity statuses.

Using Custom Attributes

Apart from the seeded attributes, you can add custom attributes to meet your specific requirements. For example, you may want to set up rule sets based on a complex business logic, and the seeded attributes do not meet the requirements completely. For seeded attributes in Oracle Leads Management, see Seeded Attributes, page B-1.

To create custom attributes, log in as an administrator, and navigate to Administration > Leads > Setup > Custom Attributes.

Setting Up Time Frames

Time frames determine the expiration date of a lead. The expiration date of a lead is the maximum length of time frame relative to the creation date. You can assign number of days to a time frame.

Some seeded examples:

Within 1 week : 7 days

1-3 months : 90 days

Customizing Time Frames

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

To create custom time frames, log in as an administrator, and navigate to Administration > Leads > Setup > Timeframe.

Notes:

- When you enter the new period and click Update, the time frame is added with the new period. There may be multiple periods defined for the same time frame.
- Similar time frames cannot be enabled at the same time. Deselect any, if enabled.
- Seeded time frames cannot be removed.

Monitoring Leads

This chapter covers the following topics:

- Monitoring Engine Overview
- Running the Workflow Background Process Concurrent Program
- Creating a Monitor Rule
- Monitoring Engine Conditions
- Viewing the Monitor Log

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:

- When a lead is created (See Creation Date Monitors)
- When a lead is assigned to a sales team (See Assignment State Monitors)

When a lead satisfies one of the above conditions, a monitor is selected for the lead based on its scope. Based on the conditions specified, notification(s) or reminder(s) are sent to the lead owner and/or the owner's manager. If the lead is still not acted upon, it may be timed-out and rerouted.

A monitor is active on a lead till the lead reaches the To condition set for that monitor. When the monitors are evaluated again after changes in some lead attributes, the monitors are either stopped or continue to run the course. They are stopped if the lead's attributes no longer match the monitor's scope. If the new state of the lead satisfies another monitor, it is reassigned.

Creation Date Monitors

Every time a new lead is created, the monitoring engine is called. If the lead matches a

Creation Date (Creation From State) monitor, the lead is attached to this monitor. For this lead, the monitors defined for Assignment From State are never used. The creation date monitors are picked up only once for a single lead.

The Creation Date monitors are also called Absolute Monitors.

Assignment State Monitors

If the lead does not match any of the Creation Date monitors, then a matched Assignment State (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 State monitors that may be active on the lead are terminated, and the workflow for the new monitor is launched. If a new Assignment State monitor is not available for the new state, then no action is taken.

The Assignment State monitors are also called Relative Monitors because they are evaluated every time some lead attributes such as Lead Owner and Lead Rating change.

Behavior of the Monitoring Engine

The following table elucidates the behavior of the monitoring engine depending on the change in the lead attributes.

Monitoring Engine Behavior

Event	Creation Date	Assignment State
Change in Lead Rank	Monitors are reevaluated. If the current monitor is no longer valid, it is stopped. No new Creation Date monitors are assigned. This lead is never picked up by an Assignment State monitor.	Monitors are reevaluated. If the current monitor is no longer valid, it is stopped. A new Assignment State monitor that satisfies the lead attributes is assigned.
Change in Lead Owner	Creation Date monitors are not evaluated.	Monitors are reevaluated. If the current monitor is no longer valid, it is stopped. A new Assignment State monitor that satisfies the lead attributes is assigned.

Scope of a Monitor

The scope of a monitor is defined by the Country and Rating attributes. A monitor can have either one or both these attributes defined for it. A lead satisfying 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.

When no monitors satisfy the lead attributes, the lead is not monitored.

Note: There is an implicit AND between the Country and Rating attributes, if both are selected.

Smart Time Frames and Expiration Dates

Smart time frames determine the expiration date of a lead. For more information about smart time frames, see *Setting Up Time Frames*, page 5-16.

The Monitoring Engine uses smart time frames through the Relative to Expiration Date check box. If the check box is selected, and the expiration date is set for the lead, the owner, or owner's manager, or both, are notified *n* days before the expiration date of the lead. If the check box is selected, and the expiration date is not set for the lead, the monitor condition will never be satisfied by the lead.

Timeouts in the Monitoring Engine

The concept of a timeout in the monitoring engine is the mechanism to make sure that leads are followed up within a certain period of time. If a timeout in days is mentioned in the monitor rule set, then the lead is reassigned to a new owner from the sales team after the elapse of the timeout days unless the rule set condition is reached.

Consider the following scenarios to understand timeouts in the monitoring engine.

Monitor 1

Monitor 1 has been defined in the application with the following conditions:

Monitor 1 Rule Conditions

Condition	Value Set
From	Creation
To	Accepted
Timelag	1 day
Relative to Expiration	N
Reminders	To Lead Owner

Condition	Value Set
Timeout	2 days

A new lead NewTech Inc. is created that satisfies Monitor 1. The monitor is started as and when the lead is created. The lead owner has one day to accept the lead. After one day, the first notification goes to the lead owner stating that the lead must be accepted. Two days after the monitor was started i.e. one day after the first notification was sent, the monitor times out and the lead gets reassigned to the new owner.

Monitor 2

Monitor 2 has been defined in the application with the following conditions:

Monitor 2 Rule Conditions

Condition	Value Set
From	Creation
To	Accepted
Timelag	1 day
Relative to Expiration	Y
Reminders	None
Timeout	2 days

Note: In this example, the Relative to Expiration condition is set to Y. When a lead is created, it is assigned an expiration date based on the Timeframe attribute.

A new lead HiBiz Corp. is created that satisfies Monitor 2. The monitor is started as and when the lead is created. The lead owner has one day to accept the lead. In this case, technically the first notification will be sent only one day prior to the expiration date of the lead. However, the timeout will still happen two days after the lead is created. This means that if the lead were due to expire in 2 weeks time, the first notification will be sent one day before the two weeks are over. But that would never happen because the monitor would always timeout in two days flat.

Timeouts are valid only for very important leads where an opportunity might be lost if

not followed up within a certain time period. For the HiBiz Corp. lead, the timeout is not needed because the objective is to follow up on the lead before it expires.

Running the Workflow Background Process Concurrent Program

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

Use the following details to run the Workflow Background Process concurrent program.

Prerequisite: None

Responsibility: System Administrator

Parameters:

- Item Type - Monitoring Engine Workflow
- Process Deferred - Y
- Process Timeout - Y
- Process Stuck - Y

Schedule: Once

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.

Navigation: Log in as an administrator and navigate to Administration > Leads > Processing Rules > Monitoring Rules.

Notes

- **Valid From and To Date:** Dates between which the rule is valid.
- **Status:** The rule will be effective only in the Active status.

Monitoring Engine Conditions

The following table lists the monitor conditions with their descriptions.

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, and Updated.
Time Lag Limit	Number of days when the first notification must be sent after the From State changes.
Relative to Expiration Date check box	Three possibilities exist: <ul style="list-style-type: none">• 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.

Viewing the Monitor Log

The monitor log keeps track of all 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.

You can search for a specific log by specifying the name of a monitor, a valid date range, the status of a monitor, and the name and ID details of a lead.

Navigation: Log in as an administrator and navigate to Administration > Leads > Setup > Monitoring Rules.

Notes

- **Reports:** From the Reports column, click the Reports icon for the monitor whose log you wish to view.
- **Valid Date:** The Valid Date range indicates dates between which the monitor should have been triggered.

Operational Reports

This chapter covers the following topics:

- Imports Records Manager Reports
- Lead Processing History Reports
- Rule Performance Reports
- Rule Diagnostics Reports
- Lead Linking Report

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 resulting from an import operation, see [Correcting Errors During Lead Import.](#), page 3-6

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.

Search By: Processing Stage (including all), Lead Name, Lead ID, Rule Set Name, Guard fields, Creation and Evaluation dates, and Result.

Group By: Unique Lead ID and Lead Name. All sorting is within a lead.

Note: If the value in the OS: Default Rating for Lead Rating Engine profile is set to BLANK, lead records that get this as their default rating

will not be displayed by this report.

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. Order and revenue tracking enables you to correlate the ranking predictability of leads in relation to actual performance.

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

Group By: Stage, Rule Set, and Result, in that sequence.

Summary Reports

- Utilization - Total number of leads processed (matching criteria)
- Total Number of Orders
- Order amount
- Revenue in \$
- Effectiveness
 - Total number and % of total of Status
 - Total number and % total Upgraded or Downgraded
 - Total number and % of total Accepted.

For failed leads which use the default resource, the report displays the lead outcome (for example, the rating) with no rule applicable. You can drill down to view the winning sub-rule for each rule set and review the criteria.

Details: To view a detailed report, click the Details link after the Summary report displays. For the detailed report, you must enter a value for one the following fields: Lead Number, Lead Name, or Lead Status.

Rule Diagnostics Reports

The Rule Diagnostics 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.

Group By: Stage, then Rule Set.

Summary Reports

- 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.

Details: To view a detailed report, click the Details link after the Summary report displays.

Lead Linking Report

The Lead Linking report enables you to assess the effectiveness of the Interaction rule sets. The report displays the leads linked against standard guards used for interaction selections. The chief goal of these reports is to help you to optimize the logic of which interactions to capture and to set the thresholds for sales follow up.

You can view how many interactions are captured in a given time frame. The report also displays the type of interactions that are captured and how many leads are linked to those interactions. The percentage of interactions linked is also displayed.

Note: Only customer-related interactions are considered while generating this report.

Search By: Date range, Country, and Source Code.

Group By: Interaction Type, Campaign

Summary Reports

- Number of interactions captured between the dates given by the filter parameters.
- Number of interactions linked.
- Percentage of interactions linked.
- Number of leads linked to interactions.

- Leads converted and percentage of leads linked

Prerequisites: Interaction rule sets must exist.

Navigation: Log in as an administrator, and navigate to Administration > Leads > Operational Reports > Lead Linking.

System Profile Options

This appendix covers the following topics:

- Setting System Profile Options
- System Profile Options in Oracle Leads Management

Setting System Profile Options

The procedure for setting up and changing system profile options is the same for all Oracle applications. For a detailed description of the procedures, refer to the *Oracle Applications System Administrator's Guide*.

Navigation: To change profile options, log into Oracle Forms as a System Administrator, and navigate to Profile > System.

Notes

- Site (S): This field displays the current value, if set, for all users at the installation site.
- Application (A): This field displays the current value, if set, for all users working under responsibilities owned by the application identified in the Find Profile Values block.
- Responsibility (R): This field displays the current value, if set, for all users working under the responsibility identified in the Find Profile Values block.
- User (U): This field displays the current value, if set, for the application user identified in the Find Profile Values block.

System Profile Options in Oracle Leads Management

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

- OS: Oracle Sales
- OSO: Oracle Field Sales
- PV: Oracle Partner Management
- OTS: Oracle TeleSales

The following tables list the system profiles used by Oracle Leads Management by category name and in alphabetical order . The tables include the following columns from left to right:

- **Req?** (Required): "Y" in this column indicates the profile is required for the applications or functions.
- **New?** - "Y" in this column indicates that the profile is new for this release.
- **Profile Name:** Name of the profile.
- **Description:** Explains what the profile does.
- **Program:** Lists the functional areas affected by the profile.
- **Level:** Level at which this profile option can be set. A = Application, S = Site, R = Responsibility, U = User.
- **Default:** Lists the default value for the profile, if any.

The following table lists the default profiles in Oracle Leads Management.

Default Profiles

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Sales Lead Default Close Reason	<p>The Update Sales Lead API assigns statuses to leads based on the value in the OS: Lead Link Status profile.</p> <p>If the status in the profile is 'Closed', the API picks up the value in this profile.</p>	Leads	S	-
N	N	OS: Default Decision Timeframe for Leads	If the lead record does not have a timeframe value, the value in this profile is set. If this profile is not set, a NULL value is inserted.	Leads	S, R, U	Within 1 week
N	N	OS: Default Lead Contact Role	Used for defaulting the contact role on the contact page.	Leads	S,R,U, A	END_USER

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Default Status for Leads	Used as the default status for a lead in the UI and API. If not set, then a NULL value is inserted, and an error notification is displayed.	Leads	S, R, U	New
N	N	OS: Default Vehicle Response Code for Leads	Used as the default value in the UI and API. If not set, then a NULL value is inserted in the database.	Leads	S, R, U	User
Y	N	OSO: Default Country	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.	Leads	S, R, A, U	United States

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OSO: Default Lead Contact Title	Sets the title for the lead contact. For example, Mr., Ms., and so on.	Leads	S,A,R, U	Ms.

The following table lists the profiles related to Lead Import.

Profiles for Lead Import

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	HZ: Execute API Callouts	<p>Turn on/off business event when TCA entities are modified. Turn off this profile before you run the Import Sales Lead concurrent program. After running the program you must turn the profile on.</p> <p>Turn off this profile before you run the Import Sales Lead concurrent program. After running the program you must turn the profile on.</p> <p>For more information, see <i>Implementat ion in the Trading Community Architecture</i></p>	Import Sales Leads	A, S, R, U	-

Req.?	New?	Profile Name	Description	Program	Level	Default
			<i>Administration Guide and Business Events in the Trading Community Architecture Technical Implementation Guide.</i>			
Y	N	OS: Use DQM Rule code to match Party	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate by checking for a matching Party in the TCA database.	DQM	A, S, R, U	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Use DQM Rule code to match Party Site	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate by checking for a matching Party Site in the TCA database.	DQM	A, S, R, U	-
Y	N	OS: Use DQM Rule code to match Contact	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate by checking for a matching Contact in the TCA database.	DQM	A,S,R, U	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Use DQM Rule code to match Person	Used during Lead Import. The rule associated with this profile decides whether the imported record is a duplicate by checking for a matching Person in the TCA database.	DQM	A,S,R, U	-
Y	Y	OS: Activate inactive parties from lead import	Activate inactive parties returned from lead import	DQM	S	Yes
Y	Y	OS: Execute Custom Code from Lead Import	Controls the execution of the custom user hook from the Import Sales Lead concurrent program.	Lead	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OM: Item Validation Organization	Manufacturing organization items are validated. This profile is used to set the AS_SALES_LEAD_LI NES.organization_id.	Lead	S, R	-
Y	Y	OS: Minimum Number of Records for Parallel Processing in Lead Import	Determines the number of child processes to be spawned by the Import Sales Lead concurrent program during parallel lead import.	Lead	S	400

The following table lists the profiles related to the Interaction Matching Engine.

Profiles for the Leads Interaction Matching Engine

Req.?	New?	Profile Name	Description	Program	Level	Default
N	Y	OS: Default Source for Interaction Matching Engine if Interaction does not have Source Code	Source promotion is not mandatory for Interaction. The default value is used when no promotion is specified.	Interaction	S	-
Y	Y	OS: Look Back Period in Days for Interaction s when Interaction Matching Engine is run for the first time	The Interaction Matching Engine looks for interactions created for the past <i>n</i> days. The value in this profile is used when the engine is run for the very first time.	Interaction	S	30

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	Y	OS: Look Back Period in Days for Leads in Interaction Matching Engine	The Interaction Matching Engine looks for leads created during the past <i>n</i> days. If not specified, the number of days is used from this profile.	Interaction	S	30

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	Y	OS: Last Interaction ID processed in Interaction Matching Engine	This profile will be invisible and cannot be updated by users. The Interaction Matching Engine updates this profile to denote which interaction ID was the last interaction processed. If the value of this profile is 0, the interactions logged in the last <i>n</i> days as defined by the above profile are processed.	Interaction	S	0
Y	Y	OS: Default Interaction Score in Interaction Matching Engine	If the interaction does not have any business type matched in the rule, the default score from this profile is used.	Interaction	S	1

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	Y	OS: Interaction Score Threshold	When the Import Sales Lead concurrent program processes records in the as_import_interface table, the value from this profile is used as the threshold for creating lead. If the interaction score is less than this value, a lead is not created.	Interaction	S	20
Y	Y	OS: Interaction Matching Engine matches contact for B2B lead	If set to Y, the Interaction Matching Engine matches primary contacts for B2B leads. If set to N, primary contacts are ignored.	Interaction	S	Y

The following table lists the profiles related to the Leads Rule Engines.

Profiles for the Leads Rule Engines

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	Y	OS: Default Qualified Flag for Lead Qualification Engine	When none of the Qualification rule sets match the lead, the default qualification flag is based on this profile.	Leads	S	No
Y	Y	OS: Enable Rule Engine Log	If set to 'Y', details about every rule engine run are logged into the pv_entity_rules_applied table.	Leads	S	Y
Y	Y	PV: Run lead deduplication rule	This profile controls the execution of the Lead Deduplication rule. If set to 'No', then lead deduplication will not be performed during lead import.	Lead Import	S	Y

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	Y	OS: Execute custom code from lead import	This profile controls the invocation of custom user hook that can be executed for each record in a lead import batch. If the profile value is set to 'No', the custom code will not be executed.	Lead Import	S	Y
Y	Y	OS: Escalation Manager for Leads	Set this profile when there is one resource to act as an escalation manager. During the process of assigning a lead to a sales representative, if the maximum number of re-routes are reached, an e-mail is sent to the escalation manager.	Lead Assignment	S	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Run Lead Monitor Engine	If set to Y, the lead is processed by the monitoring engine after it is created/up dated.	Monitoring Engine	S	Y
Y	N	OS: Maximum Reroutes per leads	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 .	Leads	S	3
N	N	OS: Lead View Scorecard data	Set to Y to view the old Score Card Mapping and rank data.	Leads	A,S,R, U	N

Req.?	New?	Profile Name	Description	Program	Level	Default
	N	OS: Default Channel for Lead Channel Selection Engine	This sets the default channel for each lead if no channel selection rule sets evaluate to true.	Leads	S	Direct
Y	N	OS: Default Rank for Lead Rating Engine	This sets the default if no rating rule sets evaluate to true.	Leads	S	Cold Lead
N	N	OS: Partners Enabled for Leads	If set to Yes, the Preferred Partner LOV in the Lead Details page is displayed. This is useful when the Channel of the lead is Indirect and is routed to a Partner.	Leads	S,A,R, U	No

The following table lists the profiles related to the assignment of leads.

Profiles for Leads Assignment

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Assign New Lead	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 LOV. If the agent does not make an entry, then the	Leads	S	No

Req.?	New?	Profile Name	Description	Program	Level	Default
			<p>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.</p> <p>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.</p>			

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Calendar setting required for assignment	The calendar setting for assigning an owner to a lead. If set to Yes, then the routing APIs check the availability of the resource in JTF Calendar before assigning the lead owner.	Leads	S	N
Y	N	OS: Default Lead Marketing Owner	The immature lead owner assignment API will use the default marketing lead owner if a owner cannot be found in the owner table.	Leads	S	-
Y	N	OS: Sales Team Creator Keep Flag	Used for defaulting the keep flag in the lead sales team page.	Leads	S,A,R, U	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Default Resource ID Used for Sales Lead Assignment	<p>If the owner of a lead declines the lead, the owner will be the next available resource defined in territory. If all the resources are used up, the lead owner will be the one defined in this profile. The original lead owner, if not defined in territory, will be removed from the sales team.</p> <p>If this profile is not set, the ownership of any unassigned lead is automatically assigned to the user who is currently logged in</p>	Territories	S	-

Req.?	New?	Profile Name	Description	Program	Level	Default
			to the application.			
N	N	OS: Lead Assignment User Hook	Used by the Sales Online Territory Assignment concurrent program. The program uses the value in this profile to identify if the user hook is enabled.	Territories	S,A,R, U	No

The following table lists the profiles related to the follow-up of leads.

Profiles for Leads Follow-up

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Lead New State Transition	If set to Y, the new state transition logic controls the transition of a lead state from one status to another, based on common business logic. If you do not want any additional restrictions, set this profile to N.	Leads	S	Y

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Auto Run Lead Engines While Update	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, you can click Run Engines to process the lead via the lead engines.	Leads	S	Y

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Address Required for Person	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, you can enter contact information in empty rows.	Leads	S	'N

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Address Required for Sales Lead	Makes address entry mandatory for leads. If set to Y, then the address is required. The API gives users an error if the address is required but not present in the record.	Leads	S	No
N	N	OS: Default Budget Status for Leads	The value in this profile will be the default value in the UI and API. If not set, this profile inserts a NULL value in the database table.	Leads	S, R, U	Pending

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Source Code Mandatory for Leads	Used to make the Campaign Source Code entry mandatory for leads. A Yes setting causes an error to appear if a user tries to save a lead without a source code.	Leads	S	No
N	N	OS: Customer Address Required for Customer, Opportunity and Lead	Indicates whether customer address is required for a customer.	Leads	S	No
Y	N	OS: Auto-relate Lead note to Customer	While creating a note, this profile is used to automatically relate the note to the customer.	Leads	S,A,R, U	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Auto-relate Lead note to Primary Contact	While creating a note, this profile is used to automatically relate the note to the primary contact.	Leads	S,A,R, U	No
Y	N	OS: Auto-relate Lead task to Customer	While creating a task, this profile option is used to automatically relate the task to the customer.	Leads	S,A,R, U	Yes
Y	N	OS: Auto-relate Lead task to Primary Contact	While creating a task, this profile option is used to automatically relate the task to the primary contact.	Leads	S,A,R, U	No

The following table lists the profiles related to the conversion of leads.

Profiles for Leads Conversion

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Auto Convert Lead to Opportunity	If a channel is defined as INDIRECT in the Channel Setup page, and this profile is set to 'Yes', an opportunity is created for the lead, and the partner matching workflow is started.	Leads	S	Yes
	N	OS: Lead Incubation Channel	Immature lead owner assignment will be used if the sales channel in this profile matches the channel assigned by the Channel Selection Engine.	Leads	S	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Lead Link Status	The value in this profile is used to set the status of the lead after linking the lead to an opportunity.	Leads	S	Con-verted to Oppor-tunity

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Lead to Opportunity Move Sales Team	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 representatives who are included as per the Territory Manager settings. The Keep flag is	Leads	S	No

Req.?	New?	Profile Name	Description	Program	Level	Default
			copied as is, and the Owner flag is copied as No.			

The following table lists the profiles related to Lead Security.

Profiles for Leads Security

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Customer Access Privilege	This profile determines who can view or update customer data.	Leads	S,R,U	Full
Y	N	OS: Manager Update Access	If a manager has subordinates in the sales team, the manager's view or update privileges will be determined by this profile.	Leads	S,R,U	View data

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Opportunity Access Privilege	This profile determines who can view or update opportunity data.	Leads	S,R,U	Global Full
N	N	OS: Privilege to Change Lead Owner	If set to Yes, you can change the owner of a lead to which you have Update access. Users who do not have this privilege can change the 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.	Leads	S, R, U	No

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OS: Sales Admin Update Access	This profile determines whether an administrator can view or update data.	Leads	S,R,U	View data
Y	N	OS: Sales Lead Access Privilege	This profile determines who can view or update lead data.	Leads	S, R, U	Full
Y	N	OS: Create Organization Privilege	If set to Yes, the Create Organization button is displayed on the Create page.	Leads	S,R,U	Yes
Y	N	OS: Create Person Privilege	If set to Y, the Create Person button is displayed on the Create page.	Leads	S,R,U	Yes

The following table lists leads profiles in Oracle TeleSales (OTS).

Leads Profiles in Oracle TeleSales

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OTS: Default Country	Used for Lead Center in OTS.	TeleSales	S,A,R, U	US
Y	N	OTS: Default Party Type	Used to default the party type in Lead Center.	TeleSales	S,A,R, U	PARTY_RELATIONSHIP
Y	N	OTS: Default Tab in Sales Lead Center	Default tab in Lead Center.	TeleSales	S,A,R, U	AST_SL C_PUR
Y	N	OTS: Default Universal Search Tab	Default Universal Search tab.	TeleSales	S,A,R, U	QUICK SEARCH
Y	N	OTS: Default Universal Search Type	Default Universal Search type.	TeleSales	S,A,R, U	PARTY_RELATIONSHIP
Y	N	OTS: Interactions- Enable Automatic Start	Used to automatically start interactions .	TeleSales	S,A,R, U	N

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OTS: Show Campaign and Offer Names instead of Codes	If set to Y,, Campaign Name is shown in the UI instead of Source Name.	TeleSales	S,A,R, U	N
N	N	OTS: Default User Role	Used to decide what the user's default role must be when the user logs in.	TeleSales	S,A,R, U	-
N	N	OTS: Default User Sales Group	Used to decide what the user's default sales group must be when the user logs in.	TeleSales	S,A,R, U	-
N	N	OTS: Minimum Number of Characters for Lookup	Minimum characters for long LOV lookup.	TeleSales	S,A,R, U	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OTS: Validation Level for Sales Leads	<p>If the value in the OS: Lead New State Transition is Yes, then the status of a lead that is converted to opportunity should not be updateable .</p> <p>In order to achieve this in the Universal Work Queue (UWQ), set this profile to 90. This ensures consistent behavior in the Lead Center, eBusiness Center, and the UWQ.</p>	TeleSales	S	90
Y	N	OTS: Default Tab in Sales Lead Center	<p>Determines which tab is active when the Oracle Telesales Lead Center is launched.</p>	TeleSales	S,A,R, U	AST_SLC_PUR

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OTS: Interactions-Generate Lead Activity	<p>Yes: Tracks activities related to leads.</p> <p>No: Disables tracking of activities related to leads.</p> <p>The default is set at the system level. Option can be set at system, application, responsibility, and user levels. If the default value for the profile option is removed, then the application functions as if the value is No.</p>	TeleSales	S,A,R, U	No
Y	N	OTS: Number of Days Prior to View	Sets the default value of days for leads and opportunities	TeleSales	S,A,R, U	-

Req.?	New?	Profile Name	Description	Program	Level	Default
Y	N	OTS: Use Primary Address to Create Lead in UWQ	Set to Yes to use the primary address when creating a lead.	TeleSales	S,A,R, U	-
Y	N	OTS: UWQ - auto launch lead and oppty center	Set to Yes to open Lead Center and Opportunity Center when a new lead or opportunity is created in UWQ.	TeleSales	S,A,R, U	-

The following table lists profiles related to leads that enable calculations in multiple currencies.

Lead Profiles Options for Multiple Currencies

Req.?	New	Profile Option	Description	Program	Level	Default
Y	N	OS: Forecast Calendar	The name of the calendar you are using to manage your forecasts. Used to get the period set name used in the as_perio d_rates table.	Calendar	S	-
Y	N	OS: Daily Conversion Type	Daily conversion type for currency conversion. Used in the amount calculation.	Currency	V	Corporate
Y	N	OS: Date Mapping Type	Used to get the pseudo period rate from the start or end of the period.	Currency	S	S

Req.?	New	Profile Option	Description	Program	Level	Default
Y	N	OS: Maximum Roll Days for Converting Amount	Maximum Roll Days for converting amount if a conversion rate does not exist for a day. Used to get maximum rollup days for currency conversion.	Currency	S	-
Y	Y	OS: Default Period Type for Currency Conversion	Default period type for currency conversion.	Currency	S	-

The following table lists the obsolete profiles.

Obsolete Profiles

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Auto assign from lead import	-	-	-	-
N	N	OS: Auto ranking from lead import	-	-	-	-

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Auto Qualify Lead	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'.	-	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Budget status required	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.	-	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Campaign code required	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.	-	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Contact phone required	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.	-	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Contact role required	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.	-	S	Yes
NOT USED	N	OS: Dead Lead Status	-	-	S	Dead Lead

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Decision time frame required	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.	-	S	Yes
Y	N	OS: Default Channel for Leads	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.	-	S, R, U	Direct

Req.?	New?	Profile Name	Description	Program	Level	Default
n/a	N	OS: Default Lead Scorecard	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.	-	SRA	1
Y	N	OS: Lead Routing Status	When the routing engine finds a sales lead owner, then the sales lead status is reset to this profile	-	S	New

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Project name required	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.	-	S	Yes
N	N	OS: Rank Lead	The Rating Engine is run when this is set to SYSTEM, if the Rank ID is null.	-	S	System

Req.?	New?	Profile Name	Description	Program	Level	Default
	N	OS: Run New Lead Qualification, Rating, Channel Selection Engines	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.	-	S	Y
N	N	OS: Sales channel required	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.	-	S	Yes

Req.?	New?	Profile Name	Description	Program	Level	Default
N	N	OS: Total budget amount required	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.	-	S	Yes

Seeded Data

This appendix covers the following topics:

- Seeded Attributes
- Seeded Lookups

Seeded Attributes

An attribute is a column in the database, and is represented as a field in the User Interface. Every object has its own set of seeded attributes.

Seeded Lead Attributes

The following table lists the seeded attributes for a lead. The Remarks column indicates other functions that use the same attributes.

Seeded Attributes for a Lead

Attribute Name	Return Type	Description	Remarks
Customer	STRING	Compare as_sales_leads.c ustomer_id	-
Customer Address	NULL_CHECK	Check whether as_sales_leads.a ddress_id is NULL.	Also used in the Deduplication Rule.
Customer Annual Revenue	CURRENCY	Compare customer revenue and currency amount.	-

Attribute Name	Return Type	Description	Remarks
Primary Contact	NUMBER	Compare as_sales_leads.p rimary_contact_p arty_id.	Also used in the Deduplication Rule.
Primary Contact Specified	NULL_CHECK	Check whether as_sales_leads.p rimary_contact_p arty_id is NULL.	-
Primary Contact Role	STRING	Compare as_sales_lead_co ntacts.contact_r ole_code.	-
Purchase Timeframe	STRING	Compare as_sales_leads.d ecision_timefram e_code.	Also used in the Deduplication Rule.
Budget Status	STRING	Compare as_sales_leads.b udget_status_cod e.	Also used in the Deduplication Rule.
Lead Score	STRING	Compare as_sales_leads.m arketing_score.	-
Lead Status	STRING	Compare as_sales_leads.s tatus_code.	Also used in the Guards.
Total Budget	CURRENCY	Compare as_sales_leads.b udget_amount, currency amount comparison.	Also used in the Deduplication Rule.
Product Category	STRING	Compare as_sales_lead_li nes.category_id.	Also used in the Deduplication Rule, and the Guards.

Attribute Name	Return Type	Description	Remarks
Purchase Quantity - Product	NUMBER	Compare as_sales_lead_lines.category_id and as_sales_lead_lines.quantity.	-
Purchase Amount - Product	CURRENCY	Compare as_sales_lead_lines.category_id and as_sales_lead_lines.budget_amount, currency amount comparison	-
Response Channel	STRING	Compare as_sales_leads.vehicle_response_code.	Also used in the Deduplication Rule.
Project	STRING	Compare as_sales_leads.parent_project.	Also used in the Deduplication Rule.
Country	STRING	Compare lead customer location country	Also used in the Deduplication Rule, the Guards, and the Monitoring Engine.
Campaign	STRING	Compare as_sales_leads.source_promotion_id.	Also used in the Deduplication Rule, and the Guards.
Qualify Flag	STRING	Compare as_sales_leads.qualified_flag.	-
Lead Rating	STRING	Compare as_sales_leads.lead_rank_id.	Also used in the Monitoring Engine.
Sales Channel	STRING	Compare as_sales_leads.channel_code.	-

Attribute Name	Return Type	Description	Remarks
Creation Date	DATE	Compare as_sales_leads.c reation_date.	Also used in the Guards.
Total Purchase Amount - Product	CURRENCY	Compare sum of as_sales_lead_li nes.budget_ammoun t, currency amount comparison.	Also used in the Deduplication Rule.
Customer/Account_T ype	STRING	Compare lead customer account type.	-
Phone Number Specified	NULL_CHECK	Check whether as_sales_lead_co ntacts.phone_id is NULL.	-
State	STRING	Compare lead customer location state.	Also used in the Guards.
Area Code	STRING	Compare lead customer or primary contact area code.	Also used in the Guards.
County	STRING	Compare lead customer location county.	Also used in the Guards.
Province	STRING	Compare lead customer location province.	Also used in the Guards.
City	STRING	Compare lead customer location city.	Also used in the Guards.
Postal Code	STRING	Compare lead customer location postal code.	Also used in the Guards.

Attribute Name	Return Type	Description	Remarks
Email Address Specified	NULL_CHECK	Check whether customer or primary contact e-mail address is NULL.	-
Email Address or Phone Specified	NULL_CHECK	Check whether customer or primary contact e-mail address or phone is NULL.	Also used in the Deduplication Rule.
Customer Category	STRING	Compare lead customer category.	Also used in the Guards.
Customer_Name	STRING	Compare lead customer name.	Also used in the Deduplication Rule.
Primary Contact Name	STRING	Compare primary contact name.	-
Primary Contact Job Title	STRING	Compare primary contact job title.	-
Email Domain	STRING	Compare customer or primary contact e-mail domain.	-
Email Address	STRING	Compare customer or primary contact e-mail address.	-
All	STRING	Matches everything.	Only used in the Guards.
Lead Note/Type	STRING	Compare lead note.	Also used in the Deduplication Rule.
Created Within	NUMBER	Compare the days when the lead was created.	Only used in the Deduplication Rule.

Attribute Name	Return Type	Description	Remarks
Interaction Score	NUMBER	Compare as_sales_leads.i nteraction_score .	-
Custom Setup	STRING	Compare custom setup of as_sales_leads.s ource_promotion_ id.	Only used in the Guards.

Seeded Interaction Matching Engine Attributes

The following table lists the seeded attributes for the Interaction Matching Engine.

Seeded Attributes for the Interaction Matching Engine

Attribute Name	Return Type	Description
All	STRING	Matches everything.
Interaction Type	STRING	The combination of action and action item.
Campaign	STRING	Compare jtf_ih_interactions.so urce_code.
Country	STRING	Compare interaction customer location country.

Seeded Lookups

Lookups appear as drop-down values in the User Interface. This section discusses the seeded lookup values in Oracle Leads Management.

Time Frame

Target Table/View: AML_SALES_LEAD_TIMEFRAMES

The following table gives the seeded lookup values for Timeframe.

Timeframe Seeded Lookup 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

Lead Rank

Target Table/View: AS_SALES_LEAD_RANKS_VL.

The following table gives the seeded lookup values for Lead Rank.

Lead Rank Seeded Lookup Values

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

Lead Status

Target Table/View: AS_STATUSES_VL.

The following table gives the seeded lookup values for Lead status.

Lead Status Seeded Lookup 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

Concurrent Programs

This appendix covers the following topics:

- Running Concurrent Programs
- Concurrent Programs in Oracle Leads Management

Running Concurrent Programs

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

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

Concurrent Programs in Oracle Leads Management

The following table lists by name and in alphabetical order the concurrent programs used by the Oracle Sales Family of eBusiness Suite applications. The table includes the following columns from left to right:

- **Concurrent Program Name:** Name of the concurrent program.
- **Description:** Explains what the concurrent program does.

List of all the concurrent programs used by the Oracle Sales Family

Mandatory	Concurrent Program Name	Description
No	Assign Territory Accesses	<p>This concurrent program assigns new territory access to sales force employees. The program prepares databases for parallel processing. It requires setting three profile options:</p> <p>OS: Territory Minimum Number of Records for Parallel Processing</p> <p>OS: Territory Number of Child Processes</p> <p>OS: Territory Records to Open for Processing Changed Accounts</p> <p>Run this program after completing the setups in Setting Up Territory Management, and after the JTF Concurrent program Generate Territory Package has run.</p> <p>Parameters:</p> <p>Run Mode (New/Restart/Total)</p> <p>Lead Status: (All/Open/Closed)</p> <p>Previous Request ID for restart mode only.</p>

Mandatory	Concurrent Program Name	Description
No	Autocreate Opportunity from Sales lead	<p>Creates opportunities from existing sales leads. This program should be run after the Import Sales Leads concurrent program.</p> <p>Parameters:</p> <p>Debug Mode</p> <p>Trace Mode</p>
No	Generate Access Records	<p>This is a child program of Assign Territory Accesses and does not need to be run separately.</p>
No	Generate Territory Packages	<p>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.</p>

Mandatory	Concurrent Program Name	Description
Yes	Initial Build of Opportunity and Lead Bins	<p>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 parameters and indexes created. Possible values for the parameter are: Small (1M), Medium (5M), Large (10M)</p> <p>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.</p>
Yes	Initial Load for Lead Reports	<p>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.</p>

Mandatory	Concurrent Program Name	Description
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_CONTA CTS. 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.

Mandatory	Concurrent Program Name	Description
No	Refresh of Opportunity Bins Data	<p>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.</p> <p>Note: Users will not see opportunities in bins created after the last refresh of Refresh Sales Credit.</p>
No	Setup Checking for Oracle Sales application	<p>This program validates Sales Setups and produces an error log that the system administrator can use to diagnose invalid setups.</p> <p>Parameters:</p> <p>Upgrade - Yes/No</p>
Y	Import Sales Leads	<p>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.</p>
Y	Workflow Background Process	<p>This program sends notifications and reminders from triggered monitors. The system administrator must schedule it to run everyday, or twice a day, if required.</p>

Mandatory	Concurrent Program Name	Description
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.
Yes	DQM Staging Program	The DQM Staging Program must be run when DQM is set up for the first time. It creates the staging schema and is vital for existence checking.
No	Purge Lead Import Interface Table	The AS_IMPORT_INTERFACE table is a temporary location where the imported records are stored before unique records are moved to the AS_SALES_LEAD table. Run the Purge Lead Import Interface Table concurrent program to delete the records from this table.

Mandatory	Concurrent Program Name	Description
Yes	Run Interaction Matching Engine to Match or Create Leads	The Run Interaction Matching Engine to Match or Create Leads concurrent program runs the Interaction Matching Engine. The Interaction Matching Engine provides the mechanism to mine and evaluate customer interactions and responses for sales follow up. The Interaction Matching Engine is driven by a rule with an activation date range.
No	Purge Unqualified Sales Leads	<p>This program removes unqualified leads from the AS_SALES_LEADS table. Unqualified leads are those leads whose Qualified attribute is not selected in the application. Unqualified leads satisfying the following conditions are removed:</p> <ul style="list-style-type: none"> <li data-bbox="1062 1171 1354 1264">• The lead status has not changed since the creation date. <li data-bbox="1062 1306 1365 1465">• The lead's original status is the same as that defined in the OS: Default Status for Leads profile.
Yes	Product Catalog Migration for Leads	This program migrates the interest_type_id and interest_code_id details to a new product hierarchy.

Oracle Leads Management API Reference

This appendix covers the following topics:

- Oracle Leads Management Procedures
- Parameter Specifications
- Standard IN Parameters
- Standard OUT Parameters
- Type Declarations
- Create Sales Lead
- Procedure Specification
- Parameter Descriptions
- Update Sales Lead
- Procedure Specification
- Parameter Descriptions
- Update Sales Lead Lines
- Procedure Specification
- Parameter Descriptions
- Delete Sales Lead Lines
- Procedure Specification
- Lead Process After Create
- Procedure Specification
- Parameter Descriptions
- Lead Process After Update
- Procedure Specification
- Parameter Descriptions

- Run Lead Engines
- Procedure Specification
- Parameter Descriptions
- Build Lead Sales Team
- Procedure Specification
- Parameter Descriptions
- Rebuild Lead Sales Team
- Procedure Specification
- Parameter Descriptions
- Update Sales Lead Contacts
- Procedure Specification
- Parameter Descriptions
- Delete Sales Lead Contacts
- Procedure Specification
- Parameter Descriptions

Oracle Leads Management Procedures

The following table lists the procedures which make up the Leads Public APIs.

Leads Public API Procedures

Procedure Name	Description
Create Sales Lead	<p>Creates a new sales lead with the specified parameters. A unique sales lead ID will be created.</p> <p>This API calls the Create Sales Lead Lines and Create Sales Lead Contacts APIs internally if the appropriate parameters are passed.</p>

Procedure Name	Description
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.
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 col-umn 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.
Lead Process After Create	This is a wrapper of the Run Lead Engines and the Build Lead Sales Team APIs. It calls the above API based on specific logic to keep lead integrity, and launches workflow process if users want to monitor the lead.
Lead Process After Update	This is a wrapper of the Run Lead Engines and the Rebuild Lead Sales Team APIs. It calls the above APIs based on specific logic to keep lead integrity.
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.

Procedure Name	Description
Rebuild Lead Sales Team	Rebuilds lead sales team to reflect the latest lead information.
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.

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.

Standard IN Parameters

The following table describes standard IN parameters which are common to all APIs

provided by Oracle Leads Management.

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	<p>Default = FND_API.G_FALSE</p> <p>If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack.</p> <p>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>
p_commit	VARCHAR2	No	<p>Default = FND_API.G_FALSE</p> <p>If set to true, the API commits before returning to the calling program.</p> <p>If set to false, then it is the calling program's responsibility to commit the transaction.</p>

Parameter	Data Type	Required	Description
p_validation_level	NUMBER	No	<p>Level of validation required.</p> <p>If set to NONE, no validation will be done in the API.</p> <p>If set to FULL, all validations (item level and record level) will be performed.</p>
P_check_access_flag	VARCHAR2	No	<p>If set to Y, access security check is per-formed.</p> <p>If set to N, access security check is not performed.</p>
P_Admin_Flag	VARCHAR2	No	<p>If set to Y, the current user has adminis-trator privileges.</p> <p>If set to N, the current user does not have administrator privileges.</p>
P_Admin_Group_Id	NUMBER	No	<p>If the current user has administrator priv-ileges, the user's SalesgroupID.</p>
P_Identity_salesforce_Id	NUMBER	No	<p>Resource ID of the current user.</p>

Standard OUT Parameters

The following table describes standard OUT parameters, which are common to all public APIs provided by Oracle Leads Management.

Note: All standard OUT parameters are required.

Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	<p>Indicates the return status of the API. The values returned are one of the following:</p> <p>FND_API.G_RET_STS_SUCCESS which indicates the API call was successful.</p> <p>FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error.</p> <p>FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.</p>
x_return_status	VARCHAR2	<p>Default = FND_API.FALSE</p> <p>If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack.</p> <p>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>
x_msg_count	NUMBER	<p>Holds the number of messages in the message list.If the error message returned is one, then the message count will be zero.</p>

Parameter	Data Type	Description
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.

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

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

CATEGORY_SET_ID           NUMBER        := FND_API.G_MISS_NUM

CATEGORY_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

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_Typ 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

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

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)

);

```

Create Sales Lead

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

Note: The Create Sales Lead API is an atomic API in the AML_SALES_LEAD_V2_PUB package. It is not the same as the Create Sales Lead API in the AS_SALES_LEADS_PUB package.

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

:= AS_UTILITY_PUB.G_MISS_PROFILE_TBL,

P_SALES_LEAD_Rec         IN   AS_SALES_LEADS_PUB.SALES_LEAD_Rec_Type

:= AS_SALES_LEADS_PUB.G_MISS_SALES_LEAD_REC,

P_SALES_LEAD_LINE_Tbl    IN   AS_SALES_LEADS_PUB.SALES_LEAD_LINE_Tbl_type

:= AS_SALES_LEADS_PUB.G_MISS_SALES_LEAD_LINE_TBL,

P_SALES_LEAD_CONTACT_Tbl IN
AS_SALES_LEADS_PUB.SALES_LEAD_CONTACT_Tbl_Type

:= AS_SALES_LEADS_PUB.G_MISS_SALES_LEAD_CONTACT_TBL,

P_Lead_note              IN   VARCHAR2 DEFAULT NULL,

P_Note_type              IN   VARCHAR2 DEFAULT NULL,

X_SALES_LEAD_ID          OUT NOCOPY NUMBER,

X_SALES_LEAD_LINE_OUT_Tbl OUT NOCOPY
AS_SALES_LEADS_PUB.SALES_LEAD_LINE_OUT_Tbl_type,

X_SALES_LEAD_CNT_OUT_Tbl OUT NOCOPY
AS_SALES_LEADS_PUB.SALES_LEAD_CNT_OUT_Tbl_Type,

X_note_id                OUT NOCOPY NUMBER,

X_Return_Status          OUT NOCOPY VARCHAR2,

X_Msg_Count              OUT NOCOPY NUMBER,

X_Msg_Data               OUT NOCOPY VARCHAR2

)

```

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.

The following table describes the IN parameters for the Create Sales Lead procedure.

IN Parameters

Parameter	Data Type	Required	Description
P_Identity_Salesforce_Id	NUMBER	No	Salesforce Id of logged in user
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_RECORD_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

Parameter	Data Type	Required	Description
P_Lead_note	VARCHAR2	No	Note to be associated with the lead
P_Note_type	VARCHAR2	No	Type of lead note

The following table describes the OUT parameters for the Create Sales Lead procedure.

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.
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.
X_note_id	NUMBER	Generated lead note id for the sales lead

Update Sales Lead

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

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

);

```

Current Version

2.0

Parameter Descriptions

Notes

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

The following table describes the IN parameters for the Update Sales Lead procedure.

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_RECORD_TYPE	No	Sales lead record type for update

Note: This procedure does not have OUT parameters.

Update Sales Lead Lines

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

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

);

```

Current Version

2.0

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.

The following table describes the IN parameters for the Update Sales Lead Lines procedure.

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)

Parameter	Data Type	Required	Description
P_Sales_lead_line_tbl	AS_SALES_LEADS_P UB.SALES_LEAD_LI NE_TBL_TYPE	No	Table of sales lead line records

The following table describes the OUT parameters for the Update Sales Lead Lines procedure.

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.

Delete Sales Lead Lines

This procedure deletes one or more sales lead lines.

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

);

```

Current Version

2.0

Lead Process After Create

This API should be called after lead header, lead line, lead contact, lead notes are created. This is a wrapper of the Run Lead Engines and the Build Lead Sales Team APIs. It calls the above APIs based on specific logic to maintain lead integrity, and launches the workflow process if users want to monitor the lead.

If the Lead Process After Create API is used, please skip the Run Lead Engines and Build Lead Sales Team. API sections.

Procedure Specification

```

PROCEDURE Lead_Process_After_Create (

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_Salesgroup_id       IN  NUMBER         := FND_API.G_MISS_NUM,

P_Sales_Lead_Id       IN  NUMBER,

X_Return_Status       OUT NOCOPY VARCHAR2,

X_Msg_Count           OUT NOCOPY NUMBER,

X_Msg_Data            OUT NOCOPY VARCHAR2

)

```

Current Version

2.0

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'.

The following table describes the IN parameters for the Lead Process After Create API.

IN Parameters

Parameter	Data Type	Required	Description
P_sales_lead_id	NUMBER	Yes	Sales Lead Identifier.

Note: This procedure does not have OUT parameters.

Lead Process After Update

This API should be called after lead header, lead line, lead contact, lead notes are updated. This is a wrapper of the Run Lead Engines and the Rebuild Lead Sales Team APIs. It calls the above APIs based on specific logic to maintain lead integrity.

If the Lead Process After Update API is used, please skip the sections for the Run Lead Engines and the Rebuild Lead Sales Team APIs.

Procedure Specification

```
PROCEDURE Lead_Process_After_Update (

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_Salesgroup_id           IN  NUMBER        := FND_API.G_MISS_NUM,

P_Sales_Lead_Id           IN  NUMBER,

X_Return_Status           OUT NOCOPY VARCHAR2,

X_Msg_Count               OUT NOCOPY NUMBER,

X_Msg_Data                OUT NOCOPY VARCHAR2

)
```

Current Version

2.0

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'.

The following table describes the IN parameters for the Lead Process After Update API.

IN Parameters

Parameter	Data Type	Required	Description
P_sales_lead_id	NUMBER	Yes	Sales Lead Identifier.

Note: This procedure does not have OUT parameters.

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.

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

);

```

Current Version

2.0

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'.

The following table describes the IN parameters for the Run Lead Engines API.

IN Parameters

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

The following table describes the OUT parameters for the Run Lead Engines API.

OUT Parameters

Parameter	Data Type	Description
X_Lead_Engines_Out_Rec	LEAD_ENGINES_OUT_Rec_ Type	Result of qualification, rating, and channel selection engines.

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.

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  
  
);
```

Current Version

2.0

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'.

The following table describes the IN parameters for the Build Lead Sales Team API.

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.

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.

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

);

Current Version

2.0

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'.

The following table describes the IN parameters for the Rebuild Lead Sales Team API.

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.

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.

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

);

```

Current Version

1.0

Parameter Descriptions

Notes

Raise an exception if the object_version_number does not match.

The following table describes the IN parameters for the Update Sales Lead Contacts API.

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.

Parameter	Data Type	Required	Description
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.

The following table describes the OUT parameters for the Update Sales Lead Contacts API.

OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See Standard OUT Parameters
x_msg_count	Number	See Standard OUT Parameters
x_msg_data	VarChar2	See 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.

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.

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  
  
);
```

Current Version

1.0

Parameter Descriptions

Notes

Raise an exception if the object_version_number does not match.

The following table describes the IN parameters for the Delete Sales Lead Contacts API.

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	-

Parameter	Data Type	Required	Description
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.

The following table describes the OUT parameters for the Delete Sales Lead Contacts API.

OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See Standard OUT Parameters
x_msg_count	Number	See Standard OUT Parameters
x_msg_data	VarChar2	See 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.

Glossary

Acquisition

Acquisition is the part of the DQM matching process that matches input record attributes against the attributes in the staged schema to get a smaller group of records that form the work unit. This process narrows down the records that can be scored in the scoring part of the matching process.

Attribute

An attribute corresponds to a column in a TCA registry table, and the attribute value is the value that is stored in the column. For example, party name is an attribute and the actual values of party names are stored in a column in the HZ_ PARTIES table.

Control File

A control file is used to give instructions to SQL*Loader.

Concurrent Process

A task in the process of completing. Each time you submit a task, you create a new concurrent process. A concurrent process runs simultaneously with other concurrent processes (and other activities on your computer) to help you complete multiple tasks at once with no interruptions to your terminal.

CSV

Stands for Comma Separated Variable. A file format used to transfer basic data between databases and spreadsheets. Each line (up to the carriage return) is considered a record. Fields within each record are divided by a comma. Each line must have the same number of fields (commas). If a comma or leading and/or trailing blanks appear in any field value, the field must be enclosed by quotes (") to indicate the information is data and not a field divider.

Deduplication

Deduplication is identifying identical records from a list. For example, you will deduplicate the records in the AS_SALES_LEAD table to avoid multiple occurrences of a lead record.

DQM

Stands for Data Quality Management. DQM manages duplicate parties in TCA. Parties are entities, of type Person, Organization, or Relationship, that can enter into business relationships. Party information includes the party name, addresses, contacts, and contact points.

DTD

Stands for Document Type Definition. The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements.

Flat File

Flat file is a tilde (~) delimited text file with data to be imported as leads.

FTP

Stands for File Transfer Protocol. FTP is the protocol used on the Internet for exchanging files. FTP uses the Internet's TCP/IP protocols to enable data transfer.

Grading

See Rating.

Match Rule

A match rule is a set of rules that determine the records that are selected and displayed as matches for the input record. A match rule consists of acquisition attributes that are used for matching and can also include scoring attributes to score the matched records.

Organization Person

An organization person is one who acts on behalf of or in the context of an organization.

Party

A party is a person, organization, or collection of parties that can enter into relationships with other parties.

Ranking

See Rating

Rating

The rating assigned to a lead by the Rating Engine based on the lead attributes. Examples of rating are Hot Lead, Cold Lead, and so on.

TCA

Stands for Trading Community Architecture. TCA is the Oracle Customer Model where

all customer records are stored. It is a common repository which is accessed by the Oracle E-Business Suite and ERP applications.

URL

Stands for Uniform Resource Locator. It is the World Wide Web address of a site on the Internet.

User Hook

User hook is part of the source code of an application that is exposed for the purpose of customization.

XML

Stands for Extensible Markup Language. XML is a metalanguage --a language for describing other languages--which lets you design your own customized markup languages for limitless different types of documents.

Index

A

APIs

- build lead sales team, D-3
- create sales lead, D-2
- delete sales lead contacts, D-4
- delete sales lead lines, D-3
- lead process after create, D-3
- lead process after update, D-3
- rebuild lead sales team, D-4
- run lead engines, D-3
- update sales lead, D-3
- update sales lead contacts, D-4
- update sales lead lines, D-3

AS_IMPORT_INTERFACE

- delete, 3-56
- import leads, 3-4
- purge, 3-56

AS_LEAD_IMPORT_ERRORS

- errors, 3-5

assignment

- automatic, 5-8
- immature, 5-9
- profiles, A-19
- setting up, 5-7

attributes

- custom, 5-16
- monitoring engine, 6-2
- scope of monitor, 6-2

B

- best practices, 5-6

business event

- definition, 3-49

business flow, 2-1

- capturing, 2-1
- cleaning leads, 2-1
- processing leads, 2-3

C

channels

- setting up, 5-5

channel selection

- overview, 5-5
- setting up, 5-5

concurrent program

- delete leads, 3-56
- flat file, 3-7
- import sales lead, 3-18
- interaction matching engine, 4-6
- process imported leads, 3-10
- staging schema, 3-27
- workflow background process, 6-5

conditions

- monitoring engine, 6-5

creating

- monitor rule, 6-5
- conditions, 6-5
- rule sets, 2-7

currency

- codes, 3-20
- profiles, A-41

custom attributes

- setting up, 5-16

- custom code
 - business events, 3-49
 - execution control, 3-50
 - handling errors, 3-50
 - post, 3-49
 - pre, 3-49
 - routing leads, 5-9
 - sample, 3-51
 - user hook, 3-55

- customize
 - time frames, 5-17

D

- data quality, 3-22
 - customer, 3-22
 - DQM
 - duplicate contact points, 3-25
 - leads, 3-43
 - OSR, 3-23
- dedupe
 - custom deduplication, 3-44
 - leads, 3-43
- deduplication rule, 3-43
 - attributes, 3-43
 - customize, 3-47
 - flow, 3-47
 - seeded, 3-44
- definitions
 - business event, 3-49
 - event subscription, 3-50
 - guards, 5-2
 - precedence, 5-2
 - rules, 5-2
- delete
 - leads, 3-56
 - unqualified leads, 5-7
 - impact, 5-7
- derived attributes, 5-16
- disqualify leads, 5-3
 - delete, 5-7
- dqm
 - customer, 3-22
 - duplicate addresses, 3-25
 - duplicate contacts, 3-25
 - duplicate customers, 3-25
 - logic, 3-26

- matching rules, 3-28
 - procedure, 3-15
- prior user hook, 3-54
- setting staging schema, 3-27

E

- event description
 - definition, 3-50

F

- flexfields, 3-9
 - reference, 3-9
- followup
 - profiles, A-24
- functioning
 - channel selection, 5-5
 - interaction mining engine, 4-3
 - monitoring engine, 6-2
 - qualification engine, 5-3
 - rating engine, 5-4

G

- generating lead, 4-6
 - process, 4-6
- guards
 - definition, 5-2

I

- immature assignment
 - setting up, 5-9
- import
 - correct errors, 3-6
 - csv, txt, 3-3, 3-4
 - errors, 3-5
 - flat file, 3-3, 3-6
 - concurrent program, 3-7
 - run SQL loader, 3-7
 - sample file, 3-8
- HTML UI, 3-4
- loading tables, 3-18
- parallel, 3-20
 - child process, 3-20
 - parent process, 3-20
- performance, 3-20
- process, 3-3

- profiles, A-6
- tables, 3-9
- import errors
 - AS_LEAD_IMPORT_ERRORS, 3-5
- import lead
 - inactive parties, 3-13
- import sales lead, 3-10
 - creating lookups, 3-17
 - currency codes, 3-20
 - custom code, 3-49
 - flow, 3-11
 - leading interface tables, 3-18
 - limitation, 3-21
 - parameters, 3-19
 - performance, 3-20
 - prerequisites, 3-14
 - running, 3-18
 - setting profiles, 3-16
 - setting territories, 3-17
 - user hook, 3-54
- interaction matching engine
 - concurrent program, 4-6
 - functioning, 4-3
 - creating lead, 4-6
 - linking existing leads, 4-5
 - mining, 4-4
 - source code, 4-4
 - generating lead, 4-6
 - generating lead process, 4-6
 - interaction score threshold, 4-3
 - interaction types, 4-2
 - overview, 4-1
 - profiles, A-11
 - report, 7-3
 - scores, 4-2
 - setting interaction types, 4-7
- interaction types
 - setting, 4-7

L

- leads
 - assignment, 5-7
 - data quality, 3-43
 - dedupe, 3-43
 - delete, 3-56
- leads assignment

- profiles, A-19
- leads engines
 - best practices, 5-6
 - channel selection, 5-5
 - overview, 5-1
 - profiles, A-15
 - qualification engine, 5-3
 - rating engine, 5-4
 - reports
 - rule flow, 5-6
 - rule flow, 5-6
- lead source, 3-2
 - campaigns, 3-2
 - interactions, 3-2
 - Oracle iStore, 3-2
 - referrals, 3-3
 - scripting, 3-2
- lead status
 - setting up, 5-16
- log
 - monitor, 6-7

M

- matching rules
 - duplicate contacts, 3-38
 - duplicate parties, 3-28
 - duplicate persons, 3-33
 - samples, 3-28
- monitoring engine
 - absolute monitor, 6-1
 - attributes, 6-2
 - conditions, 6-1, 6-5
 - assignment state, 6-2
 - creation date, 6-1
 - creating a monitor rule, 6-5
 - expiration date, 6-3
 - functioning, 6-2
 - monitor log, 6-7
 - overview, 6-1
 - relative monitor, 6-2
 - scope, 6-2
 - time frame, 6-3
 - timeout, 6-3
 - workflow background process concurrent program, 6-5

O

obsolete

- profiles, A-42

Oracle Leads Management, 1-2

- features, 1-2

- profiles, A-1

OSR

- data quality, 3-23

- logic, 3-23

- recommendation, 3-25

overview

- channel selection, 5-5

- interaction matching engine, 4-1

- leads engine, 5-1

- monitoring engine, 6-1

- Oracle Leads Management, 1-1

- qualification engine, 5-3

- rating engine, 5-4

- role of lead, 1-1

- time frames, 5-16

P

performance

- import, 3-20

precedence

- definition, 5-2

profiles, A-1

- default, A-2

- interaction matching engine, A-11

- lead import, A-6

- leads assignment, A-19

- leads conversion, A-30

- leads engines, A-15

- leads followup, A-24

- leads security, A-33

- multiple currency, A-41

- obsolete, A-42

- setting, A-1

- telesales, A-36

purge

- AS_IMPORT_INTERFACE, 3-56

- staged leads, 3-56

Q

qualification engine

- best practices, 5-6

- disqualify leads, 5-3

- overview, 5-3

- qualify leads, 5-3

qualify leads, 5-3

R

rating

- setting up, 5-4

rating engine

- functioning, 5-4

- overview, 5-4

- ratings, 5-4

reports

- import records manager, 7-1

- lead linking, 7-3

- lead processing history, 7-1

- rule diagnostic manager, 7-2

- rule flows, 5-6

- rule performance, 7-2

routing leads

- user hook, 5-9

rules

- definition, 5-2

rule sets

- creating, 2-7

running

- import sales lead concurrent program, 3-18

S

sample

- custom code, 3-51

- matching rules, 3-28

- user hook, 3-55

scenarios, 2-4

- qualifying leads, 2-4

schema

- concurrent program, 3-27

- dqm, 3-27

scope

- monitoring engine, 6-2

security

- profiles, A-33

seeded

- lookups

- lead rank, B-7
 - lead status, B-7
- setting
 - profiles, A-1
- setting up
 - channels, 5-5
 - custom attributes, 5-16
 - leads assignment, 5-7
 - lead status, 5-16
 - ratings, 5-4
 - time frames, 5-16
- staged leads
 - delete, 3-56
- status
 - seeded, B-7

T

- telesales
 - profiles, A-36
- time frames
 - customizing, 5-17
 - expiration date, 6-3
 - monitoring engine, 6-3
 - overview, 5-16
 - setting up, 5-16
- timeout
 - monitoring engine, 6-3

U

- unqualified leads
 - purge, 5-7
- user hook, 3-54
 - dedupe, 3-44
 - performance, 3-55
 - prior to DQM, 3-54
 - purpose, 3-54
 - routing leads, 5-9
 - seeded code, 3-55

