# Oracle® TeleService

Implementation Guide

Release 11*i*

**Part No. B10653-01**

May, 2003

ORACLE®

Oracle TeleService Implementation Guide, Release 11*i*

Part No. B10653-01

# Contents

## 2 Implementing Customer Care

## E Lookup Codes

# Send Us Your Comments

**Oracle TeleService Implementation Guide, Release 11*i***

**Part No. B10653-01**

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

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

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available).

You can send comments to us by electronic mail to mfgdoccomments_us@us.oracle.com. Please include your product name in the subject line: Attn: Oracle TeleService.

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

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

# Preface

## Audience for This Guide

Welcome to Release 11*i* of the Oracle TeleService Implementation Guide.

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

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

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

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

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

## How To Use This Guide

This document contains the information you need to implement Oracle TeleService. The chapters in this guide organize the implementation steps by functional area:

- Chapter 1 describes the steps required to implement service request functionality.

- Chapter 2 provides the steps for implementing the Customer Care module.
- Chapter 3 provides the steps for implementing the Charges module.
- Appendix A includes information useful for integrating relationship plans in other Oracle E-Business Suite applications.
- Appendix B answers some frequently asked questions about relationship plans.
- Appendix C lists the seeded relationship plan data.
- Appendix D describes service-request-related APIs.

## Documentation Accessibility

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

### Accessibility of Code Examples in Documentation

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

## Other Information Sources

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

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

## Online Documentation

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

## Related Documentation

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

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

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

## Documents Related to All Products

### Oracle Applications User's Guide

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

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

## Documents Related to This Product

### About Oracle TeleService

This online document describes new features and changes to Oracle TeleService since the last major release. It is available on Oracle*MetaLink*.

### Oracle Common Application Components User's Guide

This guide explains how to use Assignment Manager and Resource Manager and other common application components.

**Oracle Common Application Components Implementation Guide**

This guide explains how to implement Assignment Manager and Resource Manager and other common application components.

**Oracle Universal Work Queue Implementation Guide**

This guide explains how to implement the work queue agents use to get their work and telephony integration for Oracle TeleService.

**Oracle Knowledge Management Implementation Guide**

This guide explains how to implement the knowledge base both agents can use to resolve customer problems.

**Oracle Territory Manager User Guide**

This guide explains how to set up territories using the Oracle Territory Manager module.

**Oracle Territory Manager Implementation Guide**

This guide explains how to implement the Oracle Territory Manager module.

**Oracle Contracts Core Implementation Guide**

This guide explains in more detail how to set up events which are used by Oracle TeleService relationship plans.

**Oracle Install Base Implementation Guide**

This guide describes how to setup the installed base for tracking customer products.

**Oracle iSupport Implementation Guide**

This guide describes how to implement the Web-portal customers can use to create service requests and monitor their status.

**Oracle Field Service Implementation Guide**

This describes steps to implement Oracle Field Service.

**Oracle Scheduler Implementation Guide**

This guide helps you implement the Oracle Advanced Scheduler agents can use to schedule field-service tasks.

### Oracle Scripting Implementation Guide 11*i*

This guide describes how to implement Oracle Scripting components and test the implementation appropriately.

## Installation and System Administration

### Oracle Applications Concepts

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

### Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle9 technology stack, and the Oracle9*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

### Oracle Applications Supplemental CRM Installation Steps

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

### Upgrading Oracle Applications

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

### Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to

run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

#### Oracle Applications System Administrator's Guide

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

#### Oracle Alert User's Guide

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

#### Oracle Applications Developer's Guide

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

#### Oracle Applications User Interface Standards for Forms-Based Products

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

### Other Implementation Documentation

#### Multiple Reporting Currencies in Oracle Applications

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

#### Multiple Organizations in Oracle Applications

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

### Oracle Workflow Guide

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

### Oracle Applications Flexfields Guide

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

### Oracle eTechnical Reference Manuals

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

### Oracle Manufacturing APIs and Open Interfaces Manual

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

### Oracle Order Management Suite APIs and Open Interfaces Manual

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

### Oracle Applications Message Reference Manual

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

### Oracle CRM Application Foundation Implementation Guide

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

## Training and Support

### Training

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

### Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle TeleService 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 Oracle9*i* server, and your hardware and software environment.

### Oracle*MetaLink*

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

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

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

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

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

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

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

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

## About Oracle

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

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

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

# 1

# Implementing Service Requests

This chapter describes steps required to successfully implement service-request-related functionality in Oracle TeleService.

It covers:

# 1.1 About Configuring Service Requests to Model Business Processes

This topic provides a conceptual overview of the service request setups you can perform to model the different ways your organization handles customer problems. It covers the following related features:

- Upgrading from Previous Releases
- Service Request Statuses
- Status Groups and Status Transitions
- Service Request Types
- Summary of Setup Outcomes

## 1.1.1 Upgrading from Previous Releases

The process for setting up service request types, statuses, and status transitions has changed from previous releases. Any statuses, status transitions, and service status types set up in previous releases are automatically upgraded during the installation process.

## 1.1.2 Service Request Status

Service request statuses provide a classification system which makes it possible for your organization to track the stage of a response to a customer problem from the initial customer contact to resolution.

The statuses seeded in Oracle TeleService are:

- Waiting
- Clear

- Closed
- Open

You can set up different statuses for different types of customer problems. For example, you may wish to include the status of "Invoice Corrected" for service requests about billing problems, but for equipment exchanges you may wish to include "Received", "Fixed", and "Shipped Back to Customer". In Oracle TeleService you specify all of the possible statuses first and then specify their uses later by grouping them in Status Groups.

You can specify what kinds of changes users can make to a service request with a specific status. For example, if your organization calculates the amount a customer is to be billed only after work has been completed, then you will want to permit agents working on service requests with the status of "Open" to update the information about the customer problem but not to update charges the customer has incurred. When the work has been completed and the service request has the status of "Closed", then you may want to bar agents from updating the problem information but permit them to modify the charges.

### 1.1.3 Service Request Status Groups and Status Transitions

For each service request type, you can specify which of the service request statuses you have created make sense for that particular type of customer problem. The list of permissible statuses is called a Status Group.

For example, "Invoice Corrected" makes sense as a status of a service request for a billing question and "Repaired" as a status for inhouse equipment repairs, but not vice versa.

Status groups make it possible for you to group your statuses according to use and to determine the initial status for a service request and the permissible status transition rules for each status in the group. The initial status is the default status of the service request when agents or customers create it. The transition rules make it possible for you to determine what statuses agents or customers can choose for a service request of a given status.

For example, you may want all new service requests to get the status of New, but you will not want to permit agents to set it back to New after it has already been worked on. Or you may not want to permit agents to turn a "Closed" service request back to "Open" or "New" because work has already been completed.

**Note:** The status transition rules only govern what statuses users can select, they do not determine whether the information in the service request can be updated or not. You determine whether users can modify information when you set up the statuses themselves.

### 1.1.4 Service Request Types

Agents use service request types to categorize service requests based on the type of question or problem. If you have implemented Oracle iSupport, then customers can use service request types to categorize their problems directly.

Examples of service request types include:

- Request for Information
- Customer Complaint
- Billing Issue
- Installation Request
- Preventive Maintenance Visit
- Return
- Depot Repair

You can channel different types of customer problems and inquiries to different groups in your organization by associating service request types with different responsibilities. For example, you may wish to restrict only accountants to viewing and updating Billing Issue type service requests and you may wish to restrict access to Installation Requests to the field service organization.

Each service request type can be linked to an Oracle Workflow process that can be automatically launched when agents create or update a service request of this type. You can select to have the workflow to launch automatically by setting the profile option Service Auto Launch Workflow to Y. Alternately, agents can launch the workflow manually from a menu in the Service Request window.

If you have implemented Oracle Contracts and wish to apply discounts from the contracts to any charges customers incur during the resolution of a service request then you must link service request types to business processes.

### 1.1.5 Summary of Setup Outcomes

Here is a summary of outcomes of different service request types, status groups, and status transition setups:

- If you map a service request type to a status group with transition rules, then agents can only change the status of service requests of this type to statuses permitted by the transition rules.
- If you map a service request type to a status group that does not have any transition rules, then agents can set service requests to any status listed in that status group.
- If you do not map a service request type to a status group, then agents can set service requests of that type to any status.

## 1.2 Configuring Service Requests to Model Business Processes

Use this general procedure as a guideline for setting up service request statuses, status groups, and service request types to model your business processes.

**To set up service requests to model your business processes:**

1. Set up service request statuses for all types of service requests. See Setting Up Service Request Statuses on page 1-6.
2. Group the statuses into the groups that will be used by different service request types and create permissible status transitions for each group. See Creating and Updating Status Groups on page 1-9.
3. Set up service request types and specify what status groups they use. Indicate any workflows to be launched when different types of requests are created or updated. See Setting Up Service Request Types on page 1-15.
4. If you wish to limit access to service request types by responsibility, then:
   a. Set the system profile option Service: Use SR Type -- Responsibility Mapping to Y. This setting bars all responsibilities from viewing or modifying service requests of any service request type. (A setting of N, the default, grants all responsibilities access to all service request types.)
   b. Grant access to service request types by responsibility by creating mappings between responsibilities and service request types. See Granting Access to Service Request Types by Responsibility on page 1-19.

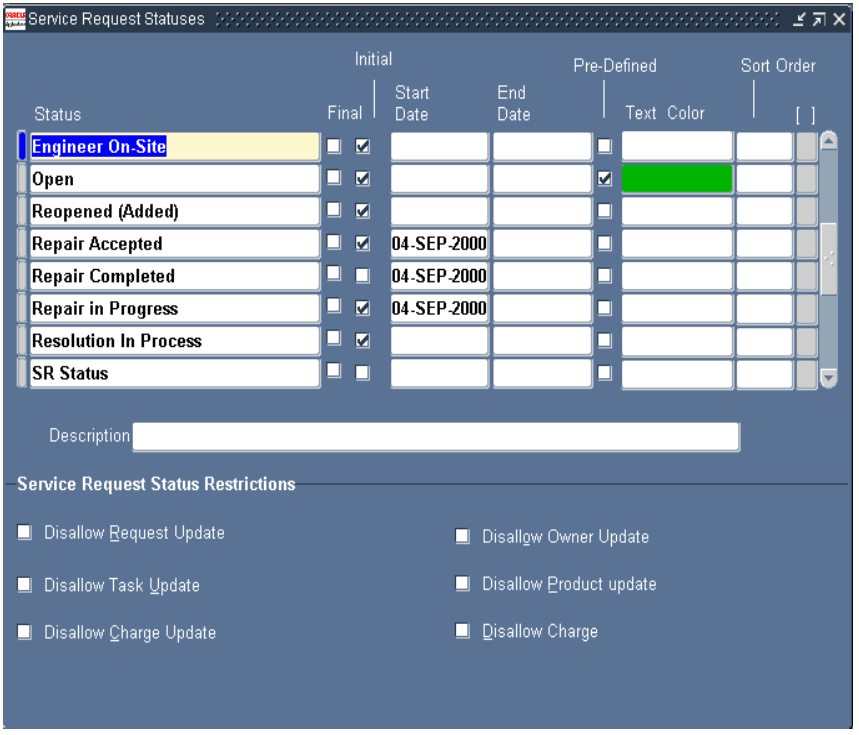
# 1.3 Setting Up Service Request Statuses

Use this procedure to set up service request statuses. The status of a service request tracks the status of a customer problem from the initial customer contact to resolution and controls what kind of information in the service request can be updated.

### To set up service request statuses:

1. Navigate to Setup > Service Requests > Service Request Status.

   The Service Request Statuses window appears.




2. Enter a status name in the Status field. The text you enter here is what is displayed in the Status List of Values (LOV) in the Service Request window.
3. Select the Final check box if you want the application to:
   - Enter the date in the Closed field of a service request set to this status.

The Closed field appears in the Service Request window header.

- Disable automatic assignments of service request ownership for this status.

  Autoassignment ignores all service requests set to a status flagged as Final. (See Setting Up Service Request Autoassignment and Load Balancing on page 1-47 for more information on automatic service request ownership assignment.)

- Terminate any instances of the workflow you have associated with the service request of that type in the Service Request Type window.

  You can set up a service request to automatically launch a workflow whenever an agent creates or updates it. Because each update launches a new instance of the workflow, each service request can have multiple instances of the workflow running at the same time. (See Setting Up Service Request Types on page 1-15 for details about associating workflows to service request types.)

  If a workflow is running and an agent sets the service request to a status you flag as Final, then the application takes one of two possible actions:

  * Warns the agent with a message that an instance of the workflow is running and permits the agent to terminate the workflow by clicking a button in the message text.

  * Aborts any instances of the workflow automatically without warning the agent.

  If you wish to abort the workflows automatically without the warning messages, then you must select the Abort Workflow on Final Status without Warning check box in the Service Request Type window. If you leave this check box unselected, then agents always receive a warning.

4. If you wish to use the status you are creating as the initial status of any type of service request, then select the Initial check box. You may wish different service requests types to have a different initial statuses. For example, you may wish to designate a service request created by a customer using Oracle iSupport as "Opened by customer" and one opened by service agent in the Service Request window as "New."

---

**Note:** You must designate a status as initial here, if you want to use it as the initial status of service requests for a Status Group.

---

5. Optionally, enter Start and End dates for the status if you wish to restrict status use. By specifying a future start date you can delay the use of a new status. You can inactivate a status by setting an end date.
6. A check mark in the Predefined check box indicates the status was seeded by Oracle. You can modify the wording of a predefined status or remove it from use by entering an end date. See Modifying the Wording of Statuses or Disabling Them on page 1-9.
7. If you wish to assign a color to the text the support agent sees in the Status field of the Service Request window, then select the color using the Text Color field.
8. If you wish to restrict users from updating certain types of information, then select one or more of the Service Request Status Restriction check boxes:
   - Disallow Request Update: Prevents users from updating the service request.
   - Disallow Task Update: Prevents users from updating tasks related to the service request in the Service Request window. (Tasks can still be modified from within the Oracle Task user interface, however.)
   - Disallow Charge Update: Prevents users from updating charges.
   - Disallow Owner Update: Prevents users from updating service request ownership.
   - Disallow Product Update: Prevents users from updating product information.
   - Disallow Charge: Prevents users from entering charges information.
9. Save your changes.
10. Repeat the above procedure for all the statuses you need for all of your service request types.

You are now ready to group the statuses you have created into Status Groups specific to each service request type according to the procedure outlined in Creating and Updating Status Groups on page 1-9.

### References

For more information about statuses and how they relate to service request types and status groups see About Configuring Service Requests to Model Business Processes on page 1-2.

## 1.4 Modifying the Wording of Statuses or Disabling Them

You can modify the wording of or disable seeded statuses. Oracle seeds the following statuses:

- Waiting
- Clear
- Closed
- Open

You can modify the wording users see in the Status list of values or disable these statuses. The application uses the seeded statuses for automatic status updates for linked service requests, so you are not permitted to disable these statuses if you are using the linking rules.

### To modify the wording of seeded statuses or disabling them:

1. Navigate to Setup > Service Requests > Service Request Status.

   The Service Request Statuses window appears. Seeded statuses are designated by a selected Predefined check box.

2. If you wish to change the wording of a given status, then make your changes in the Status field.
3. If you wish to disable a status, then enter an end date.
4. Click **Save**.

## 1.5 Creating and Updating Status Groups

Use Status Groups to create lists of statuses to be used with different service request types. Status Groups also make it possible for you to determine the initial status for a service request and the permissible status transition rules for each status in the group. This topic covers:

- Viewing and Updating Status Groups
- Creating New Status Groups
- Adding a Status to a Status Group
- Creating Transition Rules for Statuses in a Status Group

For more information about status groups and how they relate to service request statuses and types see About Configuring Service Requests to Model Business Processes on page 1-2.

## 1.5.1 Viewing and Updating Status Groups

Use this procedure to view existing status groups on the Status Group Summary window. This window provides the launching point for updating, copying, and creating new status groups.

### To view and update status groups:

1. Navigate to Setup > Service Requests >Status Groups and Transitions.

   The application displays the Status Group Summary page in a new browser window. The page lists the existing status groups.


2. If you wish to sort the existing status groups by name, description, start, and end dates, then click on the appropriate column heading.
3. If you wish to edit an existing status group, then click **Update** (the pencil icon).

4. If you wish to make a copy of an existing status group so you can use it as a template for other status groups, then click **Duplicate** (the icon with the two sheets of paper.)
5. If you wish to create a new status group, then click **Create New Group** and follow the procedure outlined in Creating New Status Groups on page 1-11.

## 1.5.2 Creating a New Status Group

Use this procedure to create a new status group.

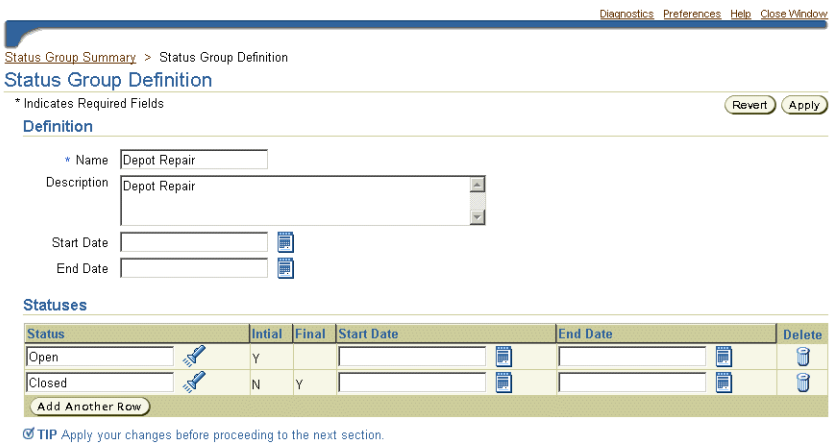
### Prerequisites

You must create statuses first following the procedure outlined in Setting Up Service Request Statuses on page 1-6.

### To create a new status group:

1. Navigate to Setup > Service Requests >Status Groups and Transitions.
2. Click **Create New Group** in the Status Group Summary page.

   The Status Group Definition page appears.




3. Enter a name for the group in the Name field.
4. Enter an optional description.

5. Enter start and end dates if you wish to restrict the availability of this status group.
6. Using the Add Another Row button, add the statuses for this group. (See Adding a Status to a Status Group on page 1-12 for details.)

   The Initial and Final flags and the rest of the status information is read only. It can only be updated while editing the statuses themselves.
7. After you have completed adding all of the statuses for your group click **Apply** to save.
8. Scroll down to the Default Initial Status region and select the default status for new service requests for this status group. You are limited to selecting statuses with the Initial flag set to Y.
9. Click **Apply**.

## 1.5.3 Adding a Status to a Status Group

Use this procedure to add a status to a Status Group you are creating or editing in the Status Group page.

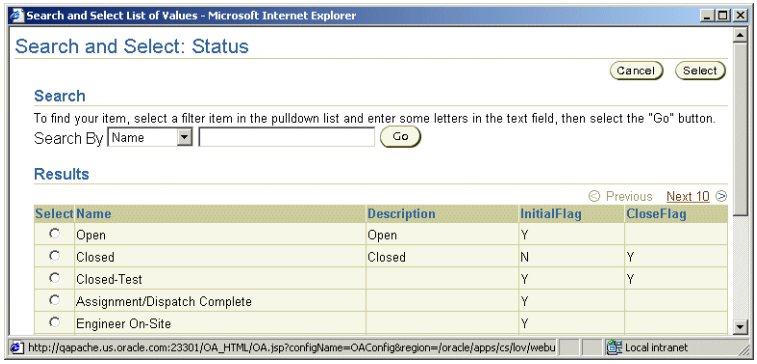
### Prerequisites

You must be in the process of creating or editing a status group in the Status Group page.

### To add a status to a status group you are editing or creating:

1. Click **Add Another Status**.
2. Click **Search for Status** (flashlight icon) to the right of the Status field.

   The Search and Select Status page appears.




3. Click **Go** to display all statuses or enter a partial name first for a short list matching your search criteria.
4. Select the radio button corresponding to the status you wish to add.'

> **Note:** You can only add one status at a time.

5. Click **Select**.

   The status you selected appears in the Status Group page.

## 1.5.4 Creating Transition Rules for Statuses in a Status Group

Use this procedure to create status transitions for a status group you are editing or creating.

**Note:** If you wish to use the automatic status update feature offered by service request relationships, then the status group must include the seeded statuses of "Waiting", "Cleared", and "Closed".

If you are setting up transition rules, then the status group must also permit all transitions to the status of "Closed". This is because an agent can indicate one service request is a duplicate of a second service request, an action that automatically sets the first service request to the status of "Closed." A transition rule you enter here takes precedence and can prevent this automatic resetting of statuses. See Setting Up Service Request Linking Through Relationships on page 1-53 for more information.

### Prerequisites

- You must be in the process of creating or editing a status group in the Status Group page.
- You must add a status to the group before you can create a transition rule for it.

### To add a transition rule for a status group you are editing:

1. Scroll down to the Transitions region of the Status Group page.
2. Click **Add Another Row**.
3. A new row opens.
4. To enter the status to transition from, click the flashlight icon to the right of the Status From field.
5. The Search and Select From page appears.
6. Click **Go** to display all statuses or enter a partial name first for a short list matching your search criteria.
7. Select the radio button corresponding to the status you wish to add.'

   **Note:** You can only add one status at a time.

8. Click **Select**.

   The status you selected appears in the Status From field in the Status Group page.

9. Enter the status you wish to transition to in the Status To field. You can do this by clicking on the flashlight icon to the right of the Status To field. The procedure is the same as for the Status From field.
10. Click **Apply**.
11. If you wish to delete a rule, then click **Delete** (the trash can icon) and click **Apply**.

## 1.6 Setting Up Service Request Types

Use this procedure to set up service request types which your agents will use to categorize customer problems. You can set up service request types to:

- Channel service requests to different groups in your organization by associating service request types with responsibilities.
- Trigger a workflow whenever a service request of that type is created.
- Limit agents to using a subset of service request statuses by linking a service request type to a Status Group.
- Enable contractual discounts by mapping a service request type to a business process.

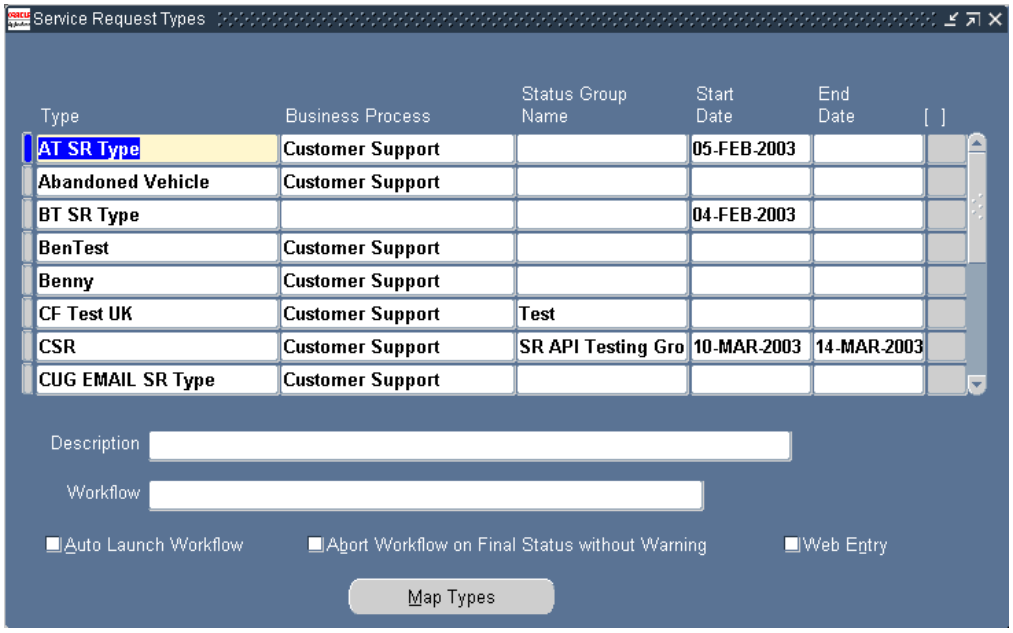
**Prerequisites:**

You must create the object you wish to link to the service request type first. For example, if you wish to link a service request type to a status group, then you must create the status group first.

### To set up service request types:

1. Navigate to Setup > Service Requests >Request Types.

   The application displays the Service Request Types window.




2. For each service request type you wish to create:

   a. Enter a name of the service request type in the Type field. This is the text agents see in the Type list of values in the Service Request window.

   b. If you have implemented Oracle Contracts, then select a business process from the Business Process list of values. Selecting the business processes links Charges to Oracle Contracts to determine the type of coverage and discounts for any billable work performed on behalf of the customer.

   c. Optionally, enter Start and End dates to control the use of service request types. By specifying a future start date you can delay the use of a new service request type. You can inactivate a type by setting an end date.

   d. Enter a description for the service request type.

   e. If you have set up a Status Group suitable for this service request type, then select it using the Status Group Name LOV. You can reuse the same status group for multiple service request types. If you leave this field blank, then users will be able to use all statuses you have defined.

**Note:** Any mapping between Service Request Types and Status Groups you make here in the Service Request Types window are overridden by entries you make in the Mapping Service Request Types window. If you wish to restrict the use of this service request type by responsibility, then perform the mapping of types to Status Groups in the Mapping Service Request Types window instead. (See Granting Access to Service Request Types by Responsibility on page 1-19.)

**f.** Optionally, you can associate a workflow with a service request type by selecting a workflow using the Workflow LOV.

Oracle TeleService includes two seeded workflows which are a part of the SRVEREQ item type and appear in the LOV:

* **Call Support Process**

  Use this workflow only under special circumstances described below.

* **Customer Support Process**

  Do not use this workflow. It is used by Oracle TeleService internally.

These two workflows are described in detail below:

* **Call Support Process**: This workflow notifies the individual service request owner each and every time a service request of this type is created or updated. The owner can set the status of the service request to closed by clicking the Resolved button on the notification.

  In this release, this workflow has been superseded by a more flexible notifications process which implementers can use to selectively notify both service request owners and customer contacts of service related events. (See Setting Up Notifications on page 1-62). You will want to associate this workflow to a type only if the resolution functionality is important to you.

  If you associate this workflow with a service request type and also use the new notifications feature, owners may receive two notices for the same event.

* **Customer Support Event Process**: This process notifies service request owners and contacts of different events related to service requests and

is used by the notifications feature of Oracle TeleService. See Setting Up Notifications on page 1-62.

---

**Note:** If you are implementing the Oracle Citizen Interaction Center, two additional seeded workflows are available to you provided that you have completed the post-installation steps: the Generic Workflow for the Citizen Interaction Center and the CIC e-mail Workflow.

---

**g.** If you have associated a workflow, then select one or more of the available check boxes:

* Auto Launch Workflow: Launches workflow automatically when a service request is created and updated.

* Abort Workflow on Final Status without Warning: Selecting this check box aborts without warning any instances of the workflow you have associated with this service request type whenever an agent sets that service request's status to a status that is flagged as Final in the Service Request Statuses window. (See Setting Up Service Request Statuses on page 1-6.)

  If you do not select this check box, then the agent always receives a warning if a workflow is about to be terminated and has a chance to cancel the abort.

**h.** If you wish to make this service request type available to customers who are creating service requests on the Web using Oracle iSupport, then select the Web Entry check box.

3. When you have completed creating your service request types, then click **Save** on the toolbar.

4. If you have selected the Auto Launch Workflow check box for any service request type, then you must set the service system profile option Service Auto Launch Workflow to Y.

You are now ready to link the service request types you have created to responsibilities and status groups according to the procedure outlined in Granting Access to Service Request Types by Responsibility on page 1-19.

## 1.7 Granting Access to Service Request Types by Responsibility

Use this procedure to grant access to service request types by responsibility. A single responsibility can be mapped to multiple service request types.

Use this procedure only if the profile option Service: Use SR Type -- Responsibility Mapping is set to Y. This setting bars all responsibilities from viewing or modifying service requests of any service request type unless you grant them access.

A setting of N, the default, grants all responsibilities access to all service request types. In this case, the mappings here are superfluous.

This procedure also permits you to map the service request type to a service request Status Group. Any mapping you make here overrides any setups you made in the Service Request Statuses window. If you do not make a mapping here, the application automatically uses any mappings you have entered in the Service Request Types window.

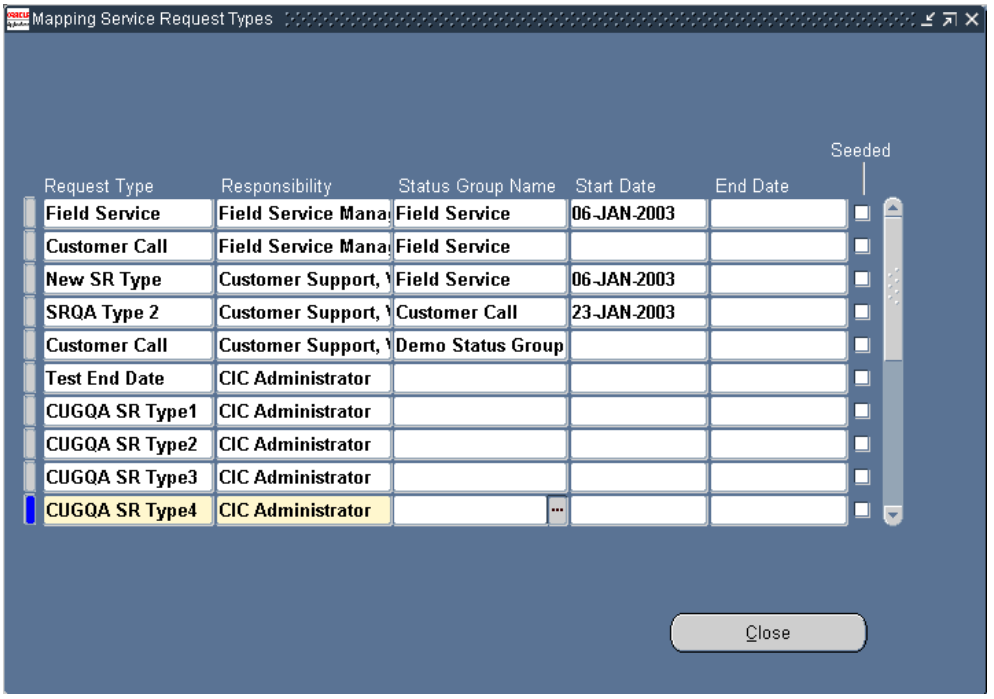
### To map service request types to responsibilities and status groups:

1. Navigate to Setup > Service Requests >Request Types.

   The application displays the Service Request Types window.

2. Click **Map Types**.

   The Mapping Service Request Types window appears.




3. Select the service request type you wish to map using the Request Type LOV.
4. Use the Responsibility LOV to enter the responsibility that will access this service request type.
5. Select the status group you wish to map to this responsibility using the Status Group Name LOV. Entries you make here override any mapping you made in the Service Request Types window.
6. If you wish to restrict the availability of this mapping, then enter a start date, end date, or both. If you do not make any entries in the date fields, then your mapping is available without restrictions.
7. Click **Save** on the toolbar to save your work. **Close** closes this window.

## 1.8 Setting Up Service Request Severities

You define service request severities to assist in setting service request priority. The service request severity reflects the support agent's perception of the reported service request. Service request severity is a mandatory field in the service request.

Following are some examples of service request severities:

- High
- Medium
- Low

### To set up a service request severity:

1. Navigate to Service Request Severities window (Setup > Service Requests > Service Severities).
2. Enter a severity name in the Severity field in the header of the Service Request window.
3. Enter a numerical value in the Importance Level field. The importance level indicates the importance of this particular severity with respect to other defined severities and must contain a unique value.

   **Note:** If you are implementing the workload autobalancing feature of service request autoassignment, then you must choose importance levels that make sense with the weights you choose for autoassignment's load balancing algorithm. See Setting Up Service Request Autoassignment and Load Balancing on page 1-47 for details.

4. Enter an optional description of the severity.
5. Optionally, enter Start and End dates to control the use of severities. By specifying a future start date you can delay the use of a new severity. You can inactivate a severity by setting an end date.
6. Optionally, select a text color in the Text Color field. The severity will be shown in this color on the service request window.
7. If a descriptive flexfield has been enabled for this form, select the appropriate values.
8. Save your service request severity.

# 1.9 Setting Up Service Request Urgencies

You define service request urgencies to provide an indicator of the customer's perception of the urgency of the service request. Urgency is an optional field in service requests.

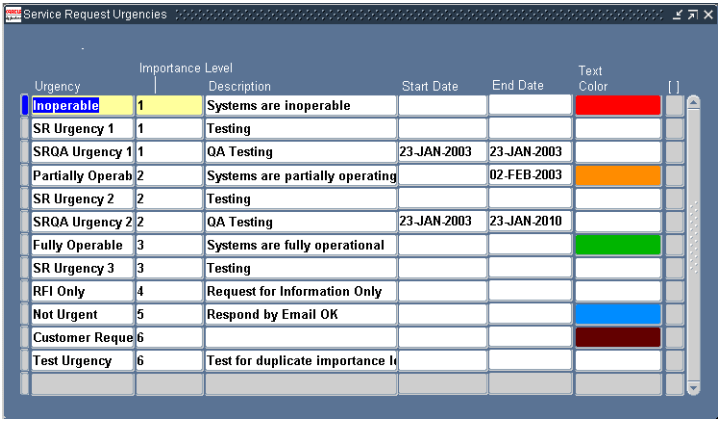
Following are some examples of service request urgencies:

- Inoperable
- Partially Operable
- Not Urgent

### To set up service request urgencies:

1. Navigate to Setup > Service Requests > Request Urgencies.

   The Service Request Urgencies window appears.




2. Enter an urgency name in the Urgency field. The urgency name appears in the Urgency list of values in the Service Request Workbench tab.
3. Enter a numerical value in the Importance Level field.The importance level indicates the importance of this particular urgency with respect to other defined urgencies and must contain a unique value.
4. Enter a description of the urgency.

5. Optionally, enter Start and End dates to control the use of urgencies. By specifying a future start date you can delay the use of a new urgency. You can inactivate an urgency by setting an end date.
6. Select a text color in the Text Color field. The urgency will be shown in this color on the service request window.
7. If a descriptive flexfield has been enabled for this form, select the appropriate values.
8. Save your service request urgency.

## 1.10 Setting Up Service Request Problem Codes

You can define service request problem codes to provide service agents with a standard way to classify customer requests. The Problem Code field is an optional field in a service request.

Problem codes are implemented using lookup codes that are standard to Oracle Applications. Lookup codes can be one of three types:

- User - No seeded values are supplied. Additional values can be added.
- Extensible - Seeded values are supplied. Additional values can be added.
- System - Seeded values are supplied. No additional values can be added.

Problem codes are of the type User.

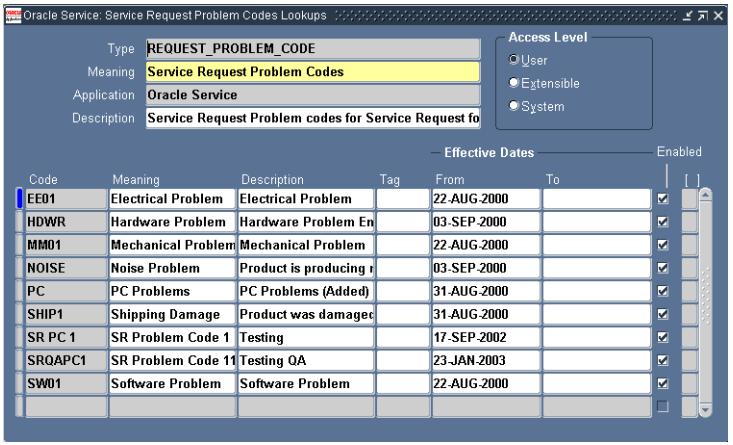
Following are some examples of service request problem codes:

- Hardware Problem
- Software Problem
- Shipping Damage

### To set up service request problem codes:

1. Navigate to Setup > Service Requests > Problem Codes.

   The Oracle Service: Service Request Problem Codes Lookups window appears. The Type and User Name fields default to REQUEST_PROBLEM_CODE and Service Request Problem codes respectively. The Application field defaults to Oracle Service. The Description defaults to Service Request Problem codes for Service Request form. All existing problem codes are displayed.




2. Enter a problem code in the Code field.
3. Enter a meaning. A meaning is a brief description of the code.
4. Enter a full description of the code in the Description field.
5. Optionally, enter Start and End dates to control the use of problem codes. By specifying a future start date you can delay the use of a new problem code. You can inactivate a problem code by setting an end date.
6. The Tag field is not used for problem codes.
7. Select the Enabled check box to make the code is available for use.
8. Optionally, enter the descriptive flexfield, if defined.
9. Save your problem code.
10. After you complete entering your problem codes, you may wish to restrict their usage by mapping them to service request types, inventory items or both. See Restricting the Use of Problem Codes on page 1-24.

## 1.11 Restricting the Use of Problem Codes

You can restrict the use of the problem codes agents use to describe customer problems by mapping them to service request types, individual inventory items, or both.

For example, if your customers call both about billing and computer hardware problems, then you may wish to map the available problem codes to the two service request types to prevent an agent from accidentally making the wrong entry. You do not want an agent to enter "Incorrect labor rate" in a service request about a faulty hard drive, for example, or "Replacement needed" for a billing question.

You may wish to restrict the use of problem codes further by inventory item to prevent an agent from using the wrong problem code for a particular product, for example, using "New battery required" for a service request that involves the replacement of the hard drive.

Here is an example of the type of user functionality one such mapping achieves. Suppose you map:

- Problem codes A, B, and C to service request type T1
- Problem codes D and E to T1 and inventory item P1

Here are the resulting user scenarios:

- If the agent selects type T1 and no item, then the available problem codes are: A, B, and C.
- If the user selects type T1 and item P1, then the available problem codes are: A,B,C, D, and E.

  A, B, and C are valid for service request type T1 and all inventory items.

  D and E are valid for the T1 and P1 combination.
- If the agent enters only item P1 but does not select a service request type, then all problem codes are available as there is no mapping defined for item P1 with a null type.

If you do not map any problem codes to a service request type, then agents can use all problem codes. Any mappings you create restrict the selection. For example, if you have created 20 different problem codes and map only the "Incorrect labor rate" problem code to the billing service request type, then agents are restricted to just that one problem code.

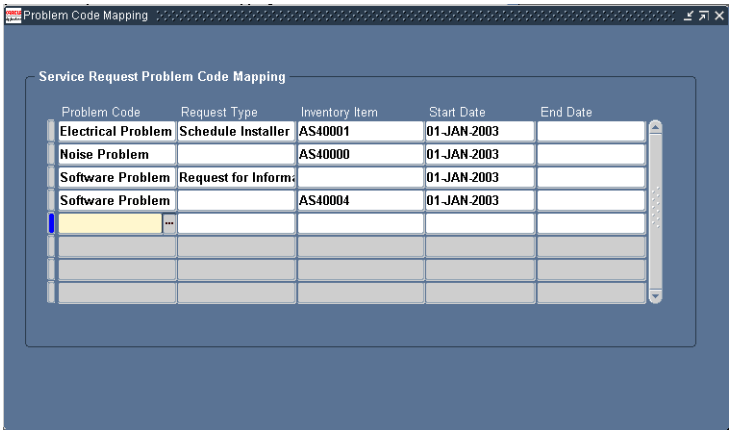
### Prerequisites

You must first set up service request types, problem codes and/or inventory items.

### To restrict use of problem codes:

1. Navigate to Setup > Service Requests > Problem Codes Mapping.

The Problem Code mapping window appears.




2. Select the problem code you wish to map.
3. Using the lists of values provided, specify a service request type, an inventory item, or both.
4. If you wish to restrict the availability of this mapping, then enter a start date, an end date, or both.
5. Repeat the above steps to specify all the permitted uses of a particular problem code. You can map a problem code to as many service request types and inventory items as you like.
6. Click **Save**.

## 1.12 Setting Up Service Request Resolution Codes

You define service request resolution codes to categorize how service requests are resolved. Resolution Code is an optional field in a service request.

Resolution codes are implemented in service requests by the usage of standard Oracle Applications lookup codes. Lookup codes can be one of three types:

- User - No seeded values are supplied. Additional values can be added.
- Extensible - Seeded values are supplied. Additional values can be added.
- System - Seeded values are supplied. No additional values can be added.

Resolution codes are of the type User.

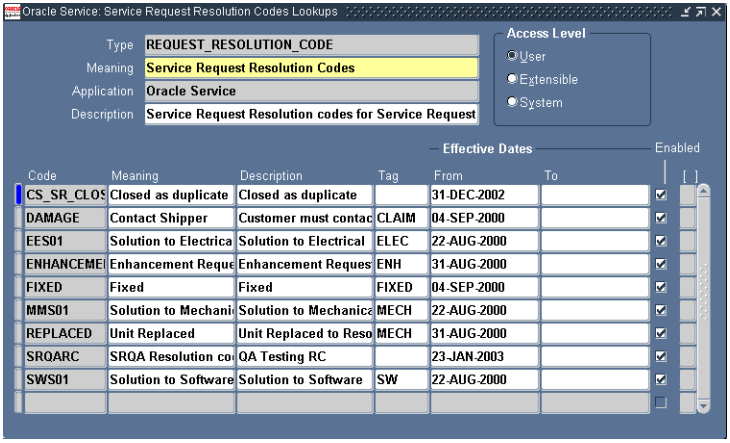
Following are some examples of service request resolution codes:

- Unit Replaced
- Patch Sent
- Documentation Sent

### To set up a service request resolution code:

1. Navigate to Setup > Service Requests > Resolution Codes.

   The Service Request Resolution Codes window appears. The type and user name fields default to REQUEST_RESOLUTION_CODE. The Description field defaults to Service Request resolution codes for Service Request form. All existing resolution codes are displayed.




2. Enter a resolution code in the Code field.
3. Enter a Meaning. A meaning is a brief description of the code.
4. Enter a full description of the code in the Description field.
5. Optionally, enter Start and End dates to control the use of resolution codes. By specifying a future start date you can delay the use of a new resolution code. You can inactivate a resolution code by setting an end date.
6. The Tag field is not used for resolution codes.

7. Select the Enabled check box to make the code available for use.
8. Optionally, enter the descriptive flexfield, if enabled.
9. Save your resolution code.

## 1.13 Setting Up Message Action Codes

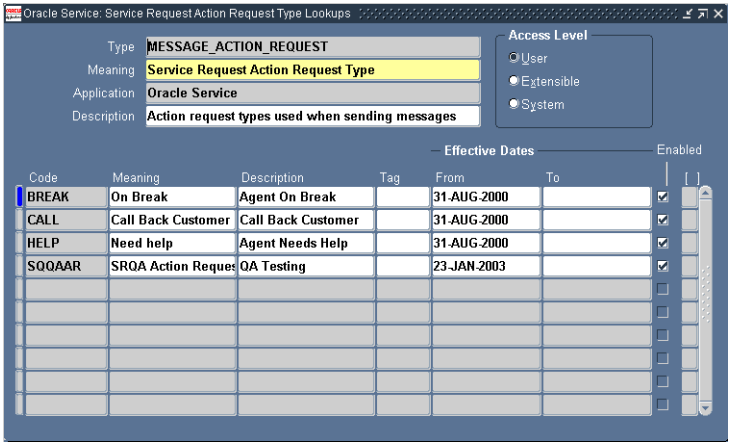
If your organization is using Oracle Workflow-based messaging feature of service requests, you can use message action codes to specify an action you want a message recipient to take.

Agents can send messages regarding a service request from the Service Request window by selecting Select Messages from the Tools menu.

### To setup a message action code:

1. Navigate to the Setup > Service Requests > Action Requests.

   The Oracle Service: Service Request Action Request Type Lookups window appears. The type field and description display MESSAGE_ACTION_ REQUEST and action request types used when sending messages.




2. Enter a message action code in the Code field.
3. Enter a brief description of the code in the Meaning field.
4. Enter a full description of the code in the Description field.

5. Enter the effective dates in the From and To fields.
6. Select the Enabled check box to make the code available for use.
7. Save the message action request code.

## 1.14 Setting Up Task Types and Priorities for Service Request Usage

This step is required to use task types and task priorities in the Service Request window's Task tab.

### To set up task types and priorities:

1. Navigate to the Task Type Setup window.

   Setup > Task Management > Task Types
2. All existing task types are displayed. For each task types which you want displayed in the Service request task tab, Click **Map Types**.
3. Associate the source Service Request to the task type.
4. Save your work.
5. Navigate to the Task Priority Setup window.
6. Setup > Task Management > Task Priority
7. All existing task priorities are displayed. For each task priority which you want to display in the Service Request Task tab, click **Map Priority.**
8. Associate the source Service Request to the task priority.
9. Save your work.

## 1.15 Setting Up the Display of Oracle Install Base Items

This step is required to display Oracle Install Base items in the Service Request window.

### To set up the display of Oracle Install Base items:

1. Navigate to the Customer Product Statuses setup window.

   Setup > Installed Base > Instance Status
2. Check the Request Allowed check box for the customer product statuses that you want to be displayed in the Service Request window.

## 1.16 Setting Up HTML Applications for Service Requests

This step is required to enable HTML applications from the Service Request. HTML applications include Oracle Knowledge Management and Oracle Install Base.

### To enable Oracle Knowledge Management:

1. Login as SYSADMIN to the JTF URL for the relevant environment.
2. Open the Users tab.
3. Click on "Users" on the "User Management" menu list.
4. Search for the required user, for example, "SUPPORT"
5. Click on the User Name, "SUPPORT" on the search result to view the user details.
6. Click **Roles** to display the Available and Assigned roles for the user.
7. Assign the role "CS_SYSTEM_ADMIN" to the user.
8. Click **Update**.

### To enable Oracle Install Base:

1. Perform steps 1 through 6 as described for Oracle Knowledge Management above.
2. Assign the role "CS_NORMAL_USER" to the user.

## 1.17 Setting Up Service Request Profile Options

System profile settings determine several default values that appear for service requests, control functionality and assist in the management of unique business processes. Profiles can be set at user, responsibility, application, and site levels.

Site-level profile settings apply to all users at an installation site. Application-level profile settings apply to all the users of the specified application and override those set at the site level. Responsibility-level profile settings apply to all users who use the responsibility to sign on to the application and override those set at site and application level. User-level profile options apply to individual users identified by their application user names and override all other profile options.

For a detailed explanation of system profile options and the procedures for setting them up, please refer to *Oracle Applications System Administrator's Guide*.

### To set up the service request profile options:

1. If you are using a Customer Support responsibility, the navigation path to update Profile Options is Other > System Profile Values.
2. You can also update profile option using a System Administrator responsibility. The navigation path is then Profiles > System.
3. Search for a specific profile and set the profile value at site, application, responsibility, or user levels. Most Support related profile options begin with the word Service, so you can query on Service% for these.

The following system profile options must be defined to use service request functionality in Oracle TeleService unless otherwise indicated.

### ASO Product Organization

This profile sets the product organization.

Valid values: Any valid Oracle Inventory organization

Default value: None

### Client Timezone

This profile option sets the default in the service request header for your user's desktop time zone.

Valid values: Any time zone

Default value: None

### Server Timezone

This profile option sets the default in the service request header for your server time zone.

Valid values: Any time zone

Default value: None

### Service: Auto Launch Workflow

This determines whether a workflow launches automatically when you save a service request

Valid values: Yes, No

Default value: None

### Service: Item Flexfield (Product)

This value determines the Oracle Inventory key flexfield structure you want to use when displaying products. The recommended value is MSTK, which is a seed value.This profile option is required even if not using Oracle Inventory.

Valid values: Any Oracle Inventory key flexfield structure.

Default value: none

### Service: Item Flexfield (Service)

This value determines the Oracle Inventory key flexfield structure you want to use when displaying support services. Recommended value is MSTK.This profile option is required even if not using Oracle Inventory.

Valid values: Any Oracle Inventory key flexfield structure.

Default value: none

### Service: Allow Charge Operating Unit Update

This profile is used to determine the operating unit that will be used for charges.

Valid values: Yes, No

Default value: No

### Service: Allow Knowledge Note Update

This profile option is used to indicate if the knowledge notes can be updated. This profile option is required if you are using Knowledge Management.

Valid values: Yes, No

Default value: No

### Service: Automatic Assignments - FORMS

This profile is used to determine if automatic assignment is done for service requests created through forms.

Valid values: Yes, No

Default value: No

### Service: Automatic Assignments - HTML

This profile is used to determine if automatic assignment is done for service requests created over the web.

Valid values: Yes, No

Default value: No

### Service: Day Unit of Measure

This profile option is used to set the default value for the unit of measure representing the day

Valid values: Any Time Unit of Measure Classes that have been defined.

Default value: Day

### Service: Default Assignee Type on the Service Request Tasks Tab

This profile option provides a default value for the Assignee Type field when creating a Service Request task.

Valid values: Employee Resource, Group Resource, Team Resource

Default value: None

### Service: Default Contact Type

This profile value sets the default value of the Contact Type.

Valid values: Person, Relationship, Employee

Default value: Relationship

### Service: Default Group Owner for Service Requests

This profile value sets the default value of the Service Request group owner.

Valid values: Any resource belonging to the Group Owner Type selected

Default value: None

### Service: Default Group Owner Type for Service Requests

This profile value sets the default value of the group owner type.

Valid values: Group, Team

Default value: Group

### Service: Default Installed Base

This profile value sets the default value of the Installed Base check box.

Valid values: Yes, No

Default value: No

### Service: Default Knowledge Base Solution Type

This value establishes the default solution type when new solutions are created. This profile option is required if you are using Knowledge Management.

Valid values: Any Knowledge Base Solution Type

Default value: SCA

### Service: Default New Note Type in Workbench Tab

This value sets the default value for the note type in a service request

Valid value: Any Note Type

Default value: None

### Service: Default Owner Type on the Service Request Tasks Tab

This profile option provides a default value for the Owner Type field when creating a Service Request task.

Valid values: Employee Resource, Group Resource, Team Resource

Default value: None

### Service: Default Party Type

This profile value sets the default value of the Customer Type.

Valid values: Person, Organization

Default value: Organization

### Service: Default Platform Category Set

This profile value sets the default value of the category set used to restrict the Platform in the Service Request.

Valid values: Any Valid values: Any category set in Oracle Inventory

Default value: None

### Service: Default Product Category Set

This profile value sets the default value of the category set which is used to restrict the items and products in the Service Request.

Valid values: Any category set in Oracle Inventory

Default value: None

### Service: Default Service Request Owner

This profile option provides a default value for the Service Request Owner field.

Valid values: Any defined resource

Default value: None

### Service: Default Service Request Owner Type

This profile option provides a default value for the Service Request Owner Type field.

Valid values: Employee Resource, Group Resource, Team Resource

Default value: None

### Service: Default Service Request Severity

This profile option provides a default value for the Service Request Severity field.

Valid values: Any Service Request Severity

Default value: None

### Service: Default Service Request Status

This profile option provides a default value for the Service Request Status field.

Valid values: Any Service Request Status

Default value: None

### Service: Default Service Request Type

This profile option provides a default value for the Service Request Type field.

Valid values: Any Service Request Type

Default value: None

### Service: Default Service Request Urgency

This profile option provides a default value for the Service Request Urgency field.

Valid values: Any Service Request Urgency

Default value: None

### Service: Default Task Assignee on the Service Request Tasks Tab

This profile option provides a default value for the Assignee field when creating a Service Request task.

Valid values: Any valid resource

Default value: None

### Service: Default Task Owner on the Service Request Tasks Tab

This profile option provides a default value for the Owner field when creating a Service Request task.

Valid values: Any valid resource

Default value: None

### Service: Group Owner Mandatory

This profile value indicates whether the Group Owner field is mandatory.

Valid values: Yes, No

Default value: No

### Service: Individual Owner Mandatory

This profile value indicates whether the individual Owner field is mandatory.

Valid values: Yes, No

Default value: No

### Service: Inventory Validation Organization

This profile is used to validate the item and product identified in a service request.

Valid values: All valid inventory organizations

Default value: None

### Service: Knowledge Base URL

This value identifies the URL for administering the Knowledge Base. This profile option is required if you are using Knowledge Management.

Valid values: The URL being used for the Knowledge Base.

Default value: None

### Service: Migration Organization for Employee Service Request

This profile is used for upgrading service requests created prior to release 11.5.6 with the customer of 'EMPLOYEE'.

Valid values: All internal customers from HZ_PARTIES

Default value: None

### Service: Navigate to Contact Center or Telesales

This profile determines which window displays when agents click the Customer button in the Service Request window.

Valid Values: Contact Center or eBusiness Center

Default value: Contact Center

### Service: Service Request Default Tab

This profile option is used to set the default tab for the Service Request window when an existing Service Request is queried. The recommended value is Workbench to expedite reviewing and updating existing Service Request activity.

Valid values: Product Coverage, Workbench, Log, Interactions, Contacts, Related Documents, Tasks

Default value: Workbench

### Service: Service Request First Tab

This profile option is used to set the tab that is first seen when a new service request is created. The recommended value is Product Coverage to allow for product entitlement handling while creating a Service Request.

Valid values: Product Coverage, Workbench, Log, Interactions, Contacts, Related Documents, Tasks

Default value: Product Coverage

### Service: System Generated Service Request Number

This profile value determines if the service request number is user entered or system generated.

Valid values: Yes, No

Default value: Yes

### Service: Visual Attribute for Setting Color for SR Number if SR is Escalated

This profile option assigns a color to the Service Request Number field when the Service Request is escalated.

Valid values: Dark Blue, Dark Green, Dark Red, Light Blue, Light Green, Light Red, Medium Blue, Medium Yellow, Medium Green, Medium Red

Default value: Light Red

### Customer Care: Default Outcome for Interactions

This value sets the default Outcome value for interactions while wrapping up a call.

Valid values: Any Outcome

Default value: None

### Customer Care: Default Reasons for Interactions

This value sets the default Reason value for interactions while wrapping up a call.

Valid values: Any Reason

Default value: None

### Customer Care: Default Results for Interactions

This value sets the default Results value for interactions while wrapping up a call.

Valid values: Any Result

Default value: None

### Notes: Default Note Status

This value sets the default status of new notes. Public means *i*Support users can view the note.

Valid values: Private, Public, Publish

Default value: None

### Task Manager: Default Task Owner

This value sets the default Task Owner value when creating a task.

Valid values: Any valid resource.

Default value: None

### Task Manager: Default Task Priority

This value sets the default Task Priority value when creating a task.

Valid values: Any Task Priority

Default value: None

### Task Manager: Default Task Status

This value sets the default Task Status value when creating a task.

Valid values: Any Task Status

Default value: None

### Task Manager: Default Task Type

This value sets the default Task Type value when creating a task.

Valid values: Any Task Type

Default value: None

## 1.18 Setting Up the Service Request Synchronize Index Concurrent Program

The Service Request Synchronize Index concurrent program is designed to enhance the text based search capability in the Service Request find window. When you use the text search keyword field in this find window, you enter some text and search for a Service Request that contains matching text in its summary field. By running this concurrent program on a periodic basis you can retrieve results that include more recently logged Service Requests.

This concurrent program is set up by the system administrator and is not a required setup. You can set up how often this program should run, every 15 minutes, 30 minutes, etc.

The following steps must be done to setup this a concurrent program:

1. Using System Administrator responsibility, navigate to:

   Security > Responsibility > Request

2. Query on the called Group "Support Reports".
3. Add the Service Request Synchronize Index concurrent program to the list.
4. Save your work.

## 1.19 Creating a Support Site

You can create different support sites for different locations or divisions or you organization. These can be used as one of the qualifiers in the Assignment Manager. An individual resource can be attached to a support site in the Resource Manager.

### To create a support site:

1. Create an Organization and Identifying it as 'Internal'

   Responsibility: Customer Support

   Navigation: Customer Management > Contact Center
2. Within the Contact Center screen, click **New** to open the Caller Information window.
3. Select a Caller Type of Organization and enter the remaining information.
4. Click **OK** to save the new Organization name and return to the Contact Center.
5. Open the Party Information tab to select the option of having this organization identified as Internal.
6. Save your work.
7. Query on the Organization you just created and open the Address Tab.
8. Enter the Status, Usage, and Address of this organization.
9. Assign the Support Site to a Resource

   Responsibility: Customer Support

   Navigation: Resource Manager > Maintain Resources > Resource
10. From the Find window, retrieve the name of the resource that you wish to link to the support site.
11. Once the resource has been obtained, go to the main resource screen. In the Roles tab, enter the various Roles and Role types that you wish to associate with this resource.

12. In the Service tab, select the Support Site that you wish to associate with this resource.
13. Confirm that Support Site is assigned to a Service Request

    Responsibility: Customer Support

    Navigation: Service Requests > Create Service Requests
14. In the Service Request window, create a service request and confirm that once an owner has been identified for the service request, the site defaults in the Support Site field.

## 1.20 Creating Owners and Supervisors

Follow these steps to create Owners and Supervisors.

1. Log-in as Sysadmin.
2. Select Responsibility: Human Resources.
3. From the "Enter and Maintain" form, and create two new Employees.
4. Make one of the Employees the Supervisor of the other, from the same form.

   This can be done using the 'Assignment' button on the same form.
5. Switch Responsibility to "System Administrator."
6. Create two new users from the form: Security-->Users-->Define.
7. Associate these users to the 2 new employees by using the LOV for the field "People" on the Define Users form.
8. Switch Responsibility to "Customer Support, Vision Enterprises."
9. Go to Resource Management-->Maintain Resources-->Import Resources.
10. Import the two new employees just created as resources, and save the resources.
11. Verify, from the Owner LOV on the Service Request form, that the two new resources exist.

## 1.21 Setting Up Service Request Autoassignment and Load Balancing

This topic explains how to set up the automatic assignment of service request ownership. It covers:

## 1.21.1 Overview

Oracle TeleService makes it possible for service organizations to automatically assign the ownership of service requests to different groups in your organization and then to individuals within those groups based on their qualifications, availability, and workload.

Service request owners usually coordinate the resolution of a customer service request and monitor the completion of the various tasks different individuals within the organization must perform in order to resolve it.

You can have the process automatically assign service request ownership in two ways:

- Every time an agent creates a service request in Oracle TeleService (either in the Contact Center or Service Request windows) and when a customer creates a service request on the Web using Oracle iSupport.
- In batches by periodically running a concurrent program.

The automatic assignment only assigns a service request if the agent does not assign ownership manually and permits the agent to override any automatic assignment by making an entry in the owner fields using the list of values.

If an agent enters a Group Owner for a service request and leaves the individual owner field blank, then the assignment engine chooses the best qualified resource from the group provided.

The autoassignment operates only on active service requests, those that have not been set to a status specified as final. (See Setting Up Service Request Statuses on page 1-6 for more details.) In this release, however, any service requests created initially with a final status are also assigned as if they were active service requests.

## 1.21.2 Autoassignment Process Components

The enable the autoassignment process you must set up:

- Load balancing formula weights

  The automatic assignment process uses a load balancing formula to calculate the best-qualified individual resource from the list of qualified resources. The

resource with the highest score is assigned ownership. You must set up the weights for this formula as described in the procedure outlined below. If the load balancing weights are not set up, the auto assignment process assigns ownership to the first resource on the list.

- The following E-Business Suite application and foundation components:
  - Resource Manager: for tracking resource skills.
  - Territory Manager: for setting up service territories.
  - Assignment Manager: to assign service requests to resources.
  - Oracle Service Contracts: to define preferred resources for contracts.
  - Oracle Install Base: to define preferred resources for product instances.

Detailed instructions on how to implement and use these components are covered in their respective documentation. For information on implementing and using Assignment Manager and Resource Manager, please refer to the *Oracle Common Application Components User's Guide* and the *Oracle Common Application Components Implementation Guide.*

For information on setting up and using the Territory Manager, please refer to the *Oracle Territory Manager User Guide* and the *Oracle Territory Manager Implementation Guide.*

Oracle Contracts and Oracle implementation guides describe how to setup preferred resources.

### 1.21.3 How the Application Assigns Ownership

If the service request is not already assigned to any group or individual resource, then the automatic assignment process:

1. Assigns group ownership

   The process searches for qualified groups among the preferred resources within contracts and the installed base, and by using service territories. If multiple groups qualify, the process always assigns ownership to the first qualified group it finds. See How the Application Assigns Ownership to Groups on page 1-44 for details.

2. Assigns ownership to individuals within the group

   The process selects the most qualified owner among the individuals in the assigned group based on availability, skills, and existing workload. See How the Application Assigns Ownership to Individuals on page 1-45 for details.

If an agent has assigned group ownership manually on the Service Request window, then the process performs only the second step, assigning ownership to individuals within the selected group.

If an agent assigns an individual owner, then the application does not assign a group owner.

### 1.21.3.1 Process for Assigning Group Ownership

If the agent has not assigned a group or individual owner to the service request, the autoassignment process:

1. Searches contracts and installed base for preferred group resources

   The process checks to see if there are any preferred groups listed in the contracts or the installed base for the specific item instance. You can have the process check either contracts or the installed base or both by setting the profile option JTFAM: Resource Search Order. If multiple preferred groups exist, it assigns ownership to the first group it finds.

   If the process finds a group owner in this step, it skips the next step. It does not search for group owners using service territories.

2. Searches for qualified resources using service territories

   If neither the contract or installed base specify group ownership, then autoassignment searches for group owners using the service territories set up with Oracle Territory Manager.

   If multiple groups qualify as potential owners, then the process assigns ownership to the first group on the list. If no groups qualify, then the user receives an error message.

   **Note:** In this release, there is no optimization between multiple groups. This means that the first group is always assigned ownership, regardless whether the resources belonging other groups may be more qualified or available to handle the service request. Workload and skills optimization works only for resources within the winning group.

### 1.21.3.2 How the Application Assigns Ownership to Individuals

If the service request is not already assigned by the agent to an individual owner, then the automatic assignment process searches for the optimal resource as follows. It:

1. Searches customer contracts and the installed base for preferred individual resources

   The autoassignment process first checks to see if there are any preferred individual resources listed in the contracts or the installed base for the specific item. You can have the process check either contracts or the installed base or both by setting the profile option JTFAM: Resource Search Order. All resources found are included in the list of potential owners. If the process finds preferred resources here, it does not check the service territories in the next step.

2. Searches for qualified individuals using service territories

   If neither the contract nor the installed base include preferred resources, then the autoassignment process checks to see if the service request attributes can be matched against any of the service territories set up with Oracle Territory Manager. If the service request can be assigned to a territory, then the individual resources listed in the territory become the potential owners.

3. Eliminates resources who are not members of the group assigned group ownership

   The autoassignment process eliminates all the individual resources that are not members of the group selected as the group owner.

4. Eliminates unavailable resources

   The autoassignment eliminates from the list of possible owners any resources who have indicated that they are not available. Agents can indicate they are unavailable in Oracle TeleService by deselecting the Available check box in the Resource Availability window (Tools1 > Web Availability).

   The autoassignment process eliminates all resources without a skill rating for either the product category or product. Resources with a rating for either one are kept on the list of potential owners.

   The problem skill rating is not used here as a filter, but it is used by the optimizer formula in the next step.

5. Picks the best resource using the optimizer formula

   The formula calculates a score for each resource and selects the one with the highest score.

   The calculation balances the level of skill for a resource (the sum of skill levels for the product, platform, product category, and the problem code) with the backlog of service requests of different severities (Severity 1 through Severity 4) and the difference in time zones between the customer and the agent.

You must assign weights for each of the factors in the formula for each service request type using the Load Balancing Weights window (Setup > Service Requests > Load Balancing Weights). You can set up different weights for each service request type and service request severity.

The optimizer formula calculates the score for each resource as follows:

```
product skill level (wt1)
+ platform skill level (wt2)
+ minutes since last SR assigned (wt3)
+ Severity 1 (wt4)
+ Severity 2 (wt5)
+ Severity 3 (wt6)
+ Severity 4 (wt7)
+ [Round (2.77 - abs(contact time zone - support site time zone in
hours/4)] (wt8)
+ problem code skill level (wt9)
+ product category skill level (wt10)
```

The weights you enter for the skill levels (wt1 to wt3, wt9, and wt10) must have a positive value. The weights for the backlog of service requests of different severities (wt 4 to 7) and for the distance (wt 8) must have a negative value.

There is no restriction on the numerical value of the weights. If you wish to remove a particular factor from the formula, you can assign it a weight of 0.

The contact time zone calculation is based on the best practices of Oracle's own support organization.

The severity level numbers come from the Importance Field set up in the Service Request Severities window (Setup > Service Requests > Severities). To enable the load balancing calculation for the backlog of service requests, you must set up the Importance Field with the values 1 through 4.

---

**Note:** If you do not set up the Load Balancing Weights formula, then the autoassignment process assigns the service request to the first resource listed in the list of resources in the contract, install base, or territory definition.

---

## 1.21.4 Setting Up Autoassignment

Use this procedure to set up the automatic assignment of service requests.

### Prerequisites

The skills-selection and load balancing portion of the autoassignment process takes into account the following skills:

- Product Category
- Product
- Problem Code

If you wish to take advantage of skills based assignment and autobalancing, then:

- You must rate the resources on these skills for them to be included in the selection process
- The service requests to be assigned must include at least one of these three factors.

### To enable service request autoassignment:

1. Specify whether you wish autoassignment to check for preferred resources in contracts, the installed base, or both by setting the profile option JTFAM: Resource Search Order.

   The available settings are:

   - Both Contracts and Install Base (this is the default value)

     Autoassignment returns all preferred resources found in contracts and installed base as the possible list of owners.

   - Contracts Preferred Resource

     Autoassignment uses only preferred resources found in contracts. If none are found, then and only then it searches the installed base.

   - Installed Base Preferred Resource.

     Autoassignment uses only preferred resources found in the installed base. If none are found, then and only then it searches contracts.

2. Determine whether you want to enable autoassignment each time a service agent creates a request using Oracle TeleService, by setting the profile option with the user profile name of Service: FORMS - Automatic Assignment of Owner.

   The available options are:

   - Yes

Set this profile option to "Yes" if you want automatic assignment each time a service request is created in Oracle TeleService (both the Customer Care and Service Request modules).

- No (the default setting)

  Set this profile option to "No" if you want to let the concurrent program Automatic Assignment of Resource for Service Request take care of the assignment instead. If you wish to assign ownership with the concurrent program, then you must set up the concurrent program to run periodically.

3. If you have implemented Oracle iSupport, then determine whether you want to enable autoassignment each time a customer creates a service request on the web, by setting the profile option Service: HTML - Automatic Assignment of Owner.

   The available options are:

   - Yes

     Set this profile option to "Yes" if you want automatic assignment each time a service request is created in iSupport.

   - No (the default setting)

     You will want to set these profile options to "No" if you want to let the concurrent program Automatic Assignment of Resource for Service Request take care of the assignment instead. If you wish to assign ownership with the concurrent program, then you must set up the concurrent program to run periodically.

4. Optionally, set up territories as described in *Oracle CRM Application Foundation User Guide* and *Oracle Territory Manager User Guide*. If you do not set up territories, then the autoassignment process skips this step.

   **Note:** If you do not set up territories and the autoassignment process does not find any preferred resources in contracts or the installed base, then autoassignment generates a user error message.

   When setting up territories, you must select Service Request as the Transaction Type. The Service Request and Task transaction type is not available for service applications at this time.

   The following table lists the transaction qualifiers in the order they appear in the list of values and explains their source.

You cannot use some of the qualifiers in the current release as indicated below.

| Territory Transaction Qualifier | Available for Use | UI Source / Comments |
|---|---|---|
| Account Code | No | Reserved for future use. |
| Area Code | Yes | Area code for a customer's phone number from the Phone field in the Customer region of the Service Request window. |
| City | Yes | City of incident address entered in the Service Request header. The application passes the address to the territory assignment program only if users select the address using the Address list of values (indicated by a selected check box next to the Address field). The qualifier ignores any addresses entered as free text. |
| Contact Preference | Yes | Customer contact preference entered in the Communication Preference field in the Service Request window. |
| Country | Yes | Country of incident address. The application passes the country to the territory assignment program only if users select the address using the Address list of values (indicated by a selected check box next to the Address field). The qualifier ignores any countries entered as free text. |
| County | No | Reserved for future use. |
| Customer Name | Yes | Customer name entered in the Name field in the Customer region of the Service Request window. |
| Customer Name Range | No | Reserved for future use. |
| Customer Site | No | Reserved for future use. |
| Group Owner | Yes | Group Owner field in the Service Request window. |
| Inventory Item | No | Do not use. Instead, use the Product qualifier for inventory items. |
| Platform | Yes | Platform field on the Product Coverage tab of the Service Request window. |

| Territory Transaction Qualifier | Available for Use | UI Source / Comments |
|---|---|---|
| Postal Code | Yes | Postal Code field of incident address. The application passes the postal code to the territory assignment program only if users select the address using the Address list of values (indicated by a selected check box next to the Address field). The qualifier ignores any countries entered as free text. |
| Problem Code | Yes | Problem Code field on the Workbench tab of the Service Request window. |
| Product | Yes | Item field in the Service Request window header. |
| Product Category | Yes | Category field on the header of Service Request window. |
| Province | Yes | Province of incident address. The application passes the province to the territory assignment program only if users select the address using the Address list of values (indicated by a selected check box next to the Address field). The qualifier ignores any province entered as free text. |
| Request Creation Channel | Yes | Channel field on the header of the Service Request Window. |
| Request Severity | Yes | Severity field on the header of the Service Request Window. |
| Request Status | Yes | Status field on the header of the Service Request Window. |
| Request Type | Yes | Type field on the header of the Service Request Window. |
| Request Urgency | Yes | Urgency field on the Workbench tab of the Service Request window. |
| Service Contract Coverage | Yes | Coverage field on the header of the Service Request Window. |
| Service Item | Yes | The Service field on the header of the Service Request window |
| Service Request Language | Yes | Language field in the header |

| Territory Transaction Qualifier | Available for Use | UI Source / Comments |
|---|---|---|
| State | Yes | State field of incident address. The application passes the state to the territory assignment program only if users select the address using the Address list of values (indicated by a selected check box next to the Address field). The qualifier ignores any states entered as free text. |
| Support Site | No | Reserved for future use. |
| VIP Customers | No | Reserved for future use. |

Because a service request is assigned both to group and individual owners, you must create separate service territories for each.

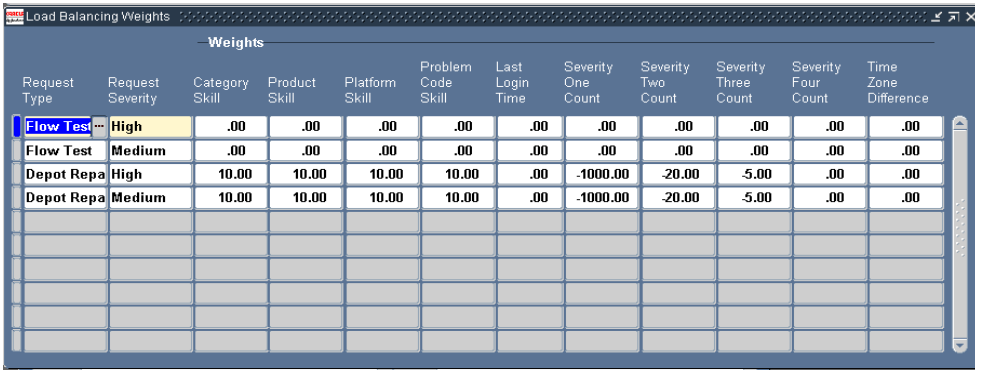
Although the Territory Management program makes it possible for you to create your territories based on product skills, Oracle recommends that you set them up on the basis of geography, or such factors as urgency and the service level agreement SLAs. The autoassignment process takes resource skills into consideration later during load balancing.

**Note:** Territory setup forces you to hardwire all possible resources into each territory by entering them on the Resources tab of the Territory Manager. This means that if you do choose to use territories for autoassignment, you must update the list of resources in your territories as employees are hired or change their skill sets.

5. Set up the load balancing weights for each service request type and severity:

   a. Navigate to **Setup > Service Requests > Load Balancing Weights.**

      The Load Balancing Weights window appears.




**b.** Enter the service request type and severity you wish to set up using the LOVs.

**c.** Enter the weights:

* If you do not wish to use a particular factor in the load balancing formula, then enter a 0.

* Enter positive numbers in Category Skill, Product Skill, Platform Skill, and Problem Code Skill.

* Enter negative numbers in Last Login Time (the time a service request has been inactive since it was assigned), Time Zone (the difference between agent and customer time zones) and for the Severity 1 through 4 Count fields. Each severity field represents the number of unassigned service requests of a particular severity.

**d.** Click **Save** on the toolbar.

**6.** If you wish to assign service requests via the concurrent program, then set up the concurrent program Automatic Assignment of Resource for Service Request to run periodically:

- If you wish to assignment all service request types of all severities and all group owners, then do not enter any parameters.
- You can restrict running the program by entering parameters for up to five service request types, severities, and group owners.

# 1.22 Setting Up Service Request Linking Through Relationships

Agents can link service requests to other service requests and to other E-Business Suite objects by creating a relationship on the Related Objects tab of the Service Request window. This topic explains this functionality and the permitted modifications. It covers:



## 1.22.1 Overview

By creating a relationship link to another service request or to another object, such as a maintenance plan for a piece of equipment, agents can make information available with the click of a mouse.

Relationships indicate, for example, that one service request is a duplicate of the other, that one is the cause of another, or that it contains relevant information for the resolution of another. An individual service request can be linked to multiple other objects.

Some types of the relationships are merely informational. Others propagate the change of status in one service request to the other service requests linked to it. For example, if agents specify that service requests A and B are caused by service request C, then closing C closes A and B as well.

---

**Note 1:** Status propagation is available only for linked service requests, not to other E-Business Suite objects.

---

The breakdown of a major system can sometimes cause multiple customer contacts to report the same problem. The status propagation permits the service organization to deal with all the duplicate service requests all at the same time.

Users have six available relationship types for linking service requests to other service requests:

- Caused by
- Root cause of
- Duplicate of
- Original for
- Reference for
- Refers to

Only the last two relationship types can be used to link service requests to other objects you set up:

- Reference for
- Refers to

Because relationships between two objects are bidirectional, they come in pairs. If a user specifies a relationship in one direction, the application automatically specifies a relationship in the other direction:

- Caused by /Root cause of
- Duplicate of /Original for
- Reference for / Refers to

The following two relationship pairs also propagate status changes between the linked service requests:

- Caused by/Root cause of
- Duplicate of/Original for

The following table explains the three most frequently used relationship types. The Action column of this table indicates both the status changes and reciprocal links created automatically by the application.

| Relationship | Action | Example |
|---|---|---|
| Caused by | Linking service request A to service request B using a "Caused by" relationship indicates that A is caused by B.<br>The application:<br>■ Sets the status of A to "Waiting".<br>■ Creates the reciprocal link "Root Cause of" for B.<br>When B is set to the status of "Closed", then the status of A is automatically set to "Clear".<br>If multiple service requests are "Caused by" B, then all are set to "Clear". | Problem:<br>A customer contact calls to say his e-mail is not working. The support agent discovers that the system administrator has already logged a service request to track this problem. The contact's e-mail is not working because the e-mail server is down.<br>Resolution:<br>The agent creates a "Caused by" link between the contact's service request and the service request already logged for the e-mail server problem. The application automatically sets the status of the contact's service request to "Waiting".<br>When the system administrator resolves the e-mail server failure and closes the server problem service request, then the application automatically updates the contact's service request status to "Clear". |
| Duplicate of | Linking service request A to service request B with the "Duplicate of" relationship indicates that A is a duplicate of B.<br>The application automatically:<br>■ Creates an "Original for" link from B to A.<br>■ Changes the status of A to "Closed" with the resolution code "Closed as Duplicate". | All call center agents have been notified of the e-mail outage. Subsequently, an employee logs a ticket to report the e-mail outage.<br>The agent who is assigned to work on the service request identifies it as a duplicate and enters the "Duplicate of" link.<br>The application automatically closes the employee's service request with the resolution code of "Closed as Duplicate" and a reference to the service request already logged for the e-mail outage. |

| Relationship | Action | Example |
|---|---|---|
| Refers to | If service request A "Refers to" service request or document B, this means that B has information relevant to A.<br><br>The application automatically creates the reciprocal "Reference for" relationship from B to A provided both documents are service requests.<br><br>Agents can use this type of relationship for linking a service request to another object, such as a maintenance plan. See Adding Reference Links to Objects Other Than Service Requests on page 1-58 for more details. | An agent working on a service request remembers that he created a similar service request for another customer. By creating a link to that old service request, he makes it possible for others to refer to it. |

**Note 1:** Agents are permitted to indicate service request A is a duplicate of service request B only if service request A has no existing relationship links of type Duplicate of, Original for, Root Cause of, and Caused by. The service request can have existing relationship links of type Reference for and Refers to. If A does contain links, and agents wish to indicate A is a duplicate of B, they must first remove the links from A.

Agents are restricted in this way for logical reasons:

- A service request can be a duplicate of only one original.
- Service requests that are originals for others cannot be chained.
- Service requests that are the cause of or caused by others will cause logical errors in the status propagation.

**Note 2:** If you are implementing status transitions as described in Status Groups and Status Transitions on page 1-3, then you must make sure that you permit the status transitions required by the automatic service request status update rules. The rules must allow the application to be able to update the status of duplicate service requests to "Closed", for example.

**Note 3:** The application automatically sends an Oracle Workflow notification to the service request owner if the automatic status update fails. If you wish, you can modify the text of the Oracle Workflow message "Notify: Service Request Update Failed Message. The item type for this message is Service Request (Internal Name: SERVEREQ).

## 1.22.2 Changes From Previous Releases

The service request relationship functionality obsoletes the need to enter the object usage of type SR_LINK in the Usage tab of the Task Setup: Object Types window.

You must still set up any objects you wish to link to service requests using this window, which is available by navigating to Task and Escalation Manager > Setup > Objects Meta-data under the CRM Administrator responsibility.

You then link the objects to service requests in the Define Relationships page as described in Setting Up Reference Links to Objects Other Than Service Requests on page 1-58.

Any objects you have set up in previous releases with usage type SR_LINK are automatically migrated to the new setup during installation.

## 1.22.3 Permissible Implementation Setups

Implementers are not permitted to add new relationship types. However, you can:

- Change relationship wording or remove a relationship type from use. (See Changing the Wording of a Relationship Type or Removing It from Use on page 1-57.)
- Add reference type links to objects other than service requests. (See Setting Up Reference Links to Objects Other Than Service Requests on page 1-58.)
- Turn off automatic service request status update rules. (See Turning Off Automatic Service Request Status Updates Rules on page 1-60.)
- Modify the wording of statuses or disable them. (See Modifying the Wording of Statuses or Disabling Them on page 1-9.)

## 1.22.4 Changing the Wording of a Relationship Type or Removing It from Use

Use this procedure to change the wording users see in the Relationship Type list of values or to remove a link type from use.

### To change the wording for a relationship type or remove it from use:

1. From the Navigator, navigate to Setup > Service Requests > Relationships and Valid Objects.

   The Define Relationships page appears in your browser window.

2. If you wish to modify the wording of a relationship agents see in the LOV, then make the modifications in the Relationship Name field.

3. If you wish to restrict the availability of a relationship, then enter a start date, an end date, or both. Entering a start date in the future postpones the use of that relationship type until that date. Entering an end date ends the availability of the relationship type at midnight of the date specified.

   **Note:** Because the relationships are reciprocal and come in pairs, restricting the availability of one relationship type automatically restricts the availability of its reciprocal.

4. Click **Apply** to save your work.

5. If you wish to permit users to link service requests to objects other than service requests using the "Reference for" and "Refers to" link pair, then click **Update Valid Objects** (the pencil icon) in the Reference for row and follow the procedure outlined in Adding Reference Links to Objects Other Than Service Requests on page 1-58.

## 1.22.5 Setting Up Reference Links to Objects Other Than Service Requests

Use this procedure to make it possible for users to link service requests to objects using the "Refers to" relationship link. The system automatically creates the reciprocal "Reference for" relationship. By default, this relationship type is already set up for service requests.

### Prerequisites

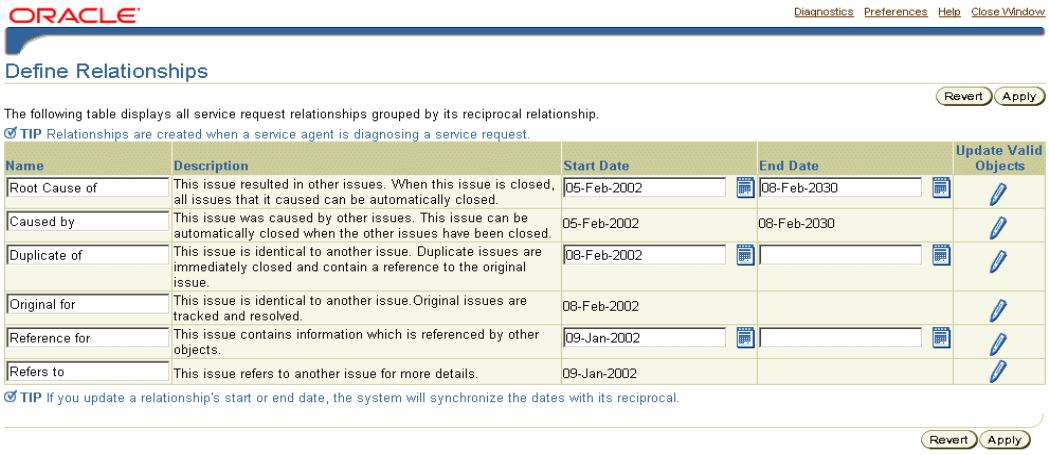
You must set up the objects you wish to link to service requests using the Task Setup: Object Types window which is available by navigating to Task and Escalation Manager > Setup > Objects Meta-data under the CRM Administrator responsibility.

See Setting Up Metadata Objects in *Oracle Common Application Components Implementation Guide* for details.

### To set up reference links to objects other than service requests:

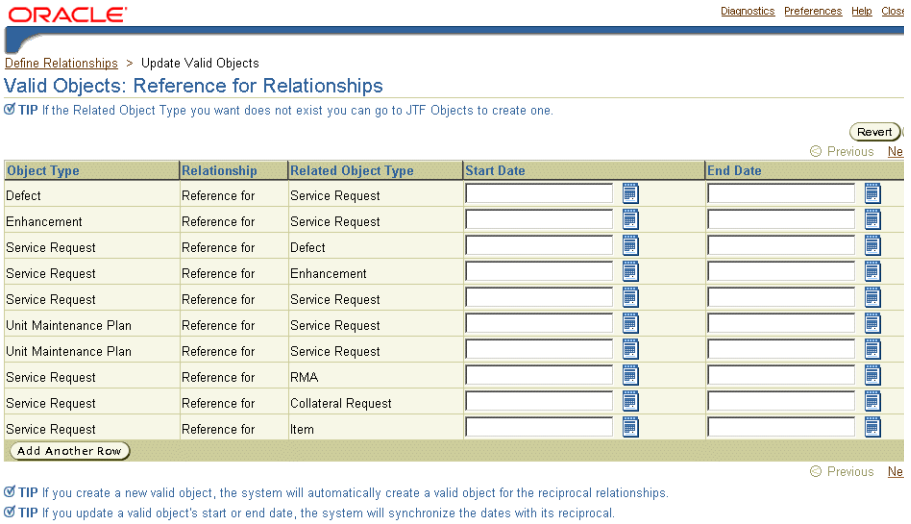
1. From the Navigator, navigate to Setup > Service Requests > Relationships and Valid Objects.

   The Define Relationships page appears in a browser window.




2. Click **Update Valid Objects** (the pencil icon) in the "Reference for" row.

   The Valid Objects page appears.




3. Click **Add Another Row**.
4. From Relationship Type drop-down list, select **Refers to.**
5. Click **Search for Related Object Type** (the searchlight icon in the Related Object Type column.)

   Search and Select: Related Object Type page appears.
6. Click **Go** to display all of the objects you have defined.
7. Select the radio button next to the name of the object.
8. Click **Select**.

   The object you selected appears in the Valid Objects: Reference for and Refers to Relationships page.
9. Click **Apply** to save your work.

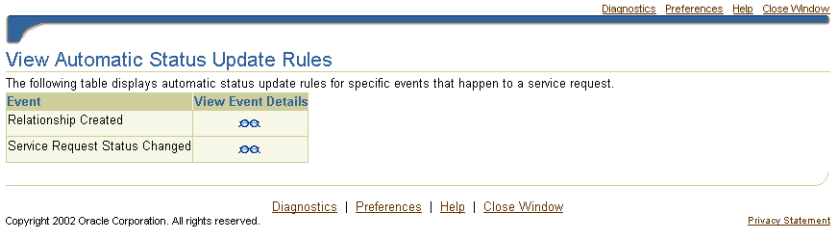
## 1.22.6 Turning Off Automatic Service Request Status Updates Rules

Use this procedure to turn off any of the seeded automatic service request status update rules. You cannot add rules or modify them.

### To turn off automatic service request status update rules:

1. From the Navigator, navigate to Setup > Service Requests > View Status Update Rules.

   The View Automatic Status Update Rules page appears in a browser window.




2. Click **View Event Details** (the glasses icon) to the right of the rule type you wish to disable.

   The View Event Details page appears. The page displays different details depending on the rule type you have chosen.


3. Enter an end date for any rule you wish to disable.
4. Click **Apply**.

# 1.23 Setting Up Notifications

This topic explains how to set up rules to trigger automatic notifications to agents and customers of different service-request-related events. It covers:



## 1.23.1 Overview

You can set up the application to automatically notify the service request owner and customer contacts associated with the service request whenever a certain event occurs, for example, whenever a service request is created, closed, or reassigned. Notifications can be sent via Oracle Workflow notifications or by e-mail.

Oracle TeleService includes a notification workflow and notification templates suitable for different types of events, but implementers must set up the rules that trigger the workflow.

The rules make it possible to selectively notify users only about events of interest to the organization. This notification functionality enhances and replaces notifications through the Call Support Process workflow which can be associated with service request types. The Call Support Process notifies users each and every time a service request is updated.

The following table describes the events that can trigger a notification message to be sent and the scope of the rule you can create.

| Event | Description | Who you can Notify | Explanation |
|---|---|---|---|
| Contact Added to Service Request | An agent or customer adds a contact to a service request. A contact can be either a customer contact or an employee. | ■ All contacts associated with the service request<br>■ The new contact<br>■ The service request owner<br>■ The primary contact | The application sends out notifications when an agent or a customer adds a contact to the service request. |
| Published Solution Added to Service Request | An agent or a customer has linked a published knowledge management solution to the service request. | ■ All contacts<br>■ Only the primary contact | The application sends out notifications whenever an agent or customer links a knowledge management solution to a service request. You may wish to use this message to e-mail suggested fixes to customer problems. |
| Service Request Status Changed | An agent or a customer changes the service request status. | ■ All contacts<br>■ The service request owner<br>■ Owners of all related service requests. | The application sends out notifications whenever a service request is changed from one status to another. You must specify the status transition that triggers the notification. For example, if you wish to notify all contacts whenever a service request is closed, then you could trigger a message whenever the status changes from "Working" to "Closed". |
| Service Request Created | An agent creates a service request. | ■ All contacts<br>■ Only the primary contact | The application sends out a notification whenever an agent creates a service request. |
| Service Request Owner Changed | An agent or concurrent program has assigned a new owner or removed an old owner. | ■ Old owner of the service request<br>■ New owner of the service request | You can notify new owner that a service request has been assigned to them and that the old owner is no longer assigned. |

| Event | Description | Who you can Notify | Explanation |
|---|---|---|---|
| Relationship Created | An agent creates a relationship between a service request and other related service requests, maintenance plans, or other E-Business Suite objects. | ■ Service request owner | Agents can specify different relationships between a service request and other service requests or E-Business Suite objects. For example, they can specify service request A is a duplicate of service request B or caused by the problem described in the service request B.<br>You can specify which relationship will trigger a notification to be sent to the service request owner. See Setting up Service Request Relationships on page 1-53 for more details. |
| Relationship Removed | An agent removes a relationship in a service request. | ■ Service request owner | You can notify the owner of the service request if another agent has removed a previously established relationship between the service request and another service request or object. You must specify the relationship you wish to trigger the notification. |
| Service Request Updated | An agent or customer updates the service request. | ■ Service request owner | You can notify the owner whenever someone else updates the service request. |

## 1.23.2 Notification Post-Install Steps

This procedure explains how to remove unnecessary lines in the body of e-mail notification messages.

By default all e-mail notification templates contain the following two unnecessary lines in the body of the message:

- `Oracle Workflow Notification (FYI)`
- `From`

### To remove unnecessary e-mail notification lines:

1. Open the "System: Mailer" item type
2. Open the "Workflow Open FYI Mail" (internal name:OPEN_MAIL_FYI) message
3. Select the Body tab.
4. Delete the first two lines and save the message.

### 1.23.3 Restrictions for the Notification Process

The following restrictions apply:

- You can only notify individual service request owners (not group owners).
- Service request owners are not notified if they themselves have taken the action that triggered the notification. For example, a service request owner who is being notified every time a service request is being updated receives notifications only when others update the service request.
- For e-mail notifications, the customer contact record must contain an e-mail address for the notifications to be sent.

### 1.23.4 Modifying The Notification Mode

You can send notifications via e-mail or Oracle Workflow notifications.

By default:

- Service request owners receive notifications via Oracle Workflow.
- All service request contacts, employee and customer, receive notifications by e-mail.

To change the notification modes, you can modify the Notification Mailer setup in Oracle Workflow as described in the *Oracle Workflow Guide*.

### 1.23.5 Modifying and Creating Notification Message Templates

The subject and text of the message users receive is stored in a Oracle Workflow message template. You can create your own notification message templates or you can modify those provided with Oracle TeleService by following the procedures in the Setting Up Oracle Workflow section of the *Oracle Workflow Guide*.

This table lists the templates provided with Oracle TeleService:

| Message Name | Description |
|---|---|
| Notify Contact: Service Request Created Message | Notifies a contact that they have been listed as a contact in a service request. |
| Notify Contact: Service Request Status Updated Message | Notifies recipient that a service request status has been updated. |
| Notify New Owner: Service Request Reassigned Message | Notifies an agent that they have been assigned as the new owner of a service request. |

| Message Name | Description |
|---|---|
| Notify Old Owner: Service Request Reassigned Message | Notifies an agent that ownership of the service request has been reassigned to someone else. |
| Notify Owner: Service Request Created Message | Notifies an agent that a service request was created with them as the owner. |
| Notify Owner: Service Request Status Updated Message | Notifies the recipient that the service request status was updated. |
| Notify Primary Contact: Service Request Created Message | Notifies the recipient that a service request has been created with them listed as a primary contact. |
| Notify: Published Solution Added to Service Request Message | Notifies recipient that a knowledge base solution has been added to the service request. |
| Notify: Related Service Request Created Message | Notifies recipient that an agent has created a relationship with another service request. For example, if an agent has indicated another service request is a duplicate of the service request. |
| Notify: Related Service Request Removed Message | Notifies recipient that a relationship between a service request and a second service request or object has been removed. For example, if an agent removes the link indicating a duplicate service request. |
| Notify: Related Service Request Updated Message | Notifies recipient that a linked service request has been updated. |
| Notify: Service Request Contact Added Message | Notifies recipient that a new contact has been added to the service request. |
| Notify: Service Request Updated Message | Notifies recipient that a service request has been updated. |

Here are the message requirements:

- All seeded messages are defined in the Oracle Workflow item type Service Request (Internal Name: SERVEREQ)
- The internal name for all new messages you create must start with: `CS_SR_EVT`
- The message definition can be up to 1376 characters in length.
- Although Oracle has seeded only text messages, you can set up HTML messages instead.
- You can automatically display service request details in your message by entering different tokens. Here is the list of available tokens:

| Display Name | Internal Name |
|---|---|
| Created By | CREATED_BY |
| Customer Product ID | CUSTOMER_PRODUCT_ID |
| Expected Resolution Date | EXPECTED_RESOLUTION_DATE |
| Initiator Role | INITIATOR_ROLE |
| Inventory Item ID | INVENTORY_ITEM_ID |
| Old Request Status | REQUEST_STATUS_OLD |
| Owner ID | OWNER_ID |
| Owner Name | OWNER_NAME |
| Owner Role | OWNER_ROLE |
| Previous Owner ID | PREV_OWNER_ID |
| Previous Owner Name | PREV_OWNER_NAME |
| Previous Owner Role | PREV_OWNER_ROLE |
| Problem Code | PROBLEM_CODE |
| Problem Description | PROBLEM_DESCRIPTION |
| Product Description | PRODUCT_DESCRIPTION |
| Request Customer | REQUEST_CUSTOMER |
| Request Date | REQUEST_DATE |
| Request ID | REQUEST_ID |
| Request Number | REQUEST_NUMBER |
| Request Severity | REQUEST_SEVERITY |
| Request Status | REQUEST_STATUS |
| Request Summary | REQUEST_SUMMARY |
| Request Type | REQUEST_TYPE |
| Request Urgency | REQUEST_URGENCY |
| Resp Appl ID | RESP_APPL_ID |
| Resp ID | RESP_ID |
| Response Deadline | RESPONSE_DEADLINE |

| Display Name | Internal Name |
|---|---|
| Response Time | RESPONSE_TIME |
| Solution Summary | SOLUTION_SUMMARY |
| Updated By | UPDATED_BY |
| User ID | USER_ID |
| Workflow Administrator Role | WF_ADMINISTRATOR |

The SERVEREQ item type includes additional tokens which are specific to internal application workflows.

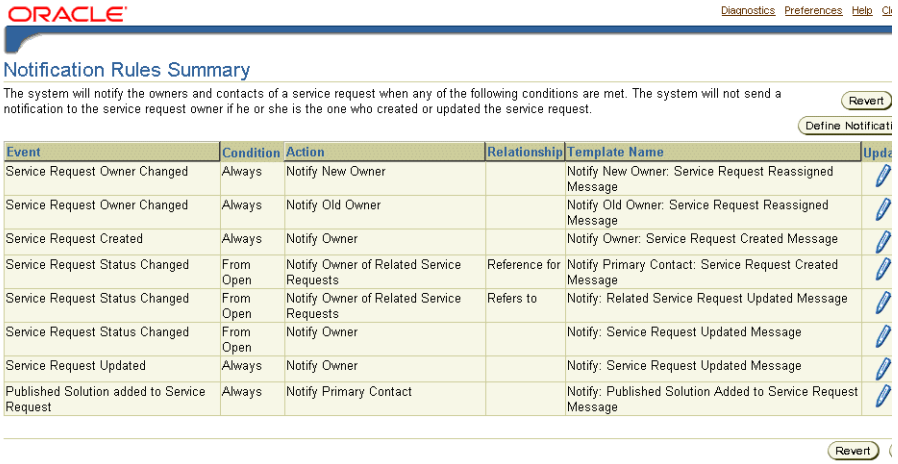
## 1.23.6 Displaying Existing Notification Rules

Use this procedure to display existing notification rules.

### To display existing notification rules:

1. Navigate to Setup > Service Request > View Notification Rules.

   The View Notification Rules page appears listing any rules you have created previously.




   You can:

- Delete a rule by clicking **Delete** (the trash can icon).
- Update a rule by clicking **Update** (the pencil icon). See Updating Existing Notification Rules on page 1-70.
- Create a new rule by clicking **Define Notification Rules.** Creating a New Notification Rule on page 1-69.

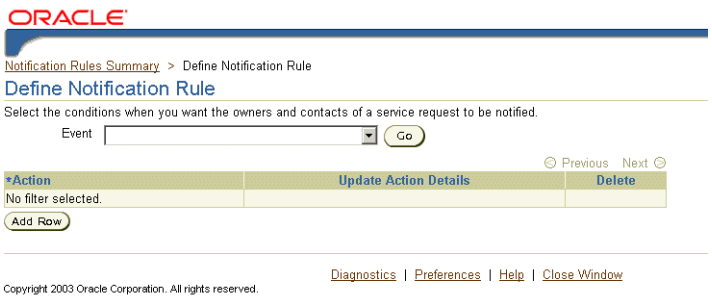
## 1.23.7 Creating a New Notification Rule

Use this procedure to create a new notification rule.

### To create a new notification rule:

1. Navigate to Setup > Service Request > View Notification Rules.
2. Click **Define Notification Rules**.

   The Define Notification Rule page appears.




3. Select the event for the rule using the Event drop-down box.
4. Click **Go**.

   The page displays any rules already created for this event.
5. Click **Add Row**.
6. In the new row, select the condition(s) you wish to use as the trigger the notification. The conditions are different for each type of event.

   For example, if you wish to send out a notification each time a service request changes a status, then select:

   - The status to trigger the notification.

- The condition

  Whether you want to send the notification each time a service request changes to the target status or changes from that status.

- Whom you want to notify

  You have three choices: you can notify all contacts on the service request, the service request owner, or all the owners of service requests that have been linked to the service request via a relationship.

7. Click **Update Action Details**.

   The Notification Details page appears.

8. Click **Search for Notification Template** (the searchlight icon).

   The Search and Select: Notification Template page appears.

9. If you wish to restrict your search, then enter a complete or partial name of a template in the Search by field.

10. Click **Go**.

    The page displays the available templates in the Results region.

11. Select the radio button next to the template you wish to use.

12. Click **Select**.

    The template name displays in the Notification Details page.

13. Click **Apply.**

14. If you wish to create or update another rule, then select the View Notifications Rule link at the top left of the page.

## 1.23.8 Updating Existing Notification Rules

Use this procedure to update existing notification rules. You can delete rules or update existing rule conditions and attach different notification message templates.

### To update an existing notification rule:

1. Display the existing notification rules by navigating to Setup > Service Request > View Notification Rules.

   The Define Notification Rule page appears.

2. Click **Update** (the pencil icon) for the rule you wish to update.

The Define Notification Rule page appears displaying the rules already associated with this event.

3. If you wish to delete a rule, then click **Delete** (the trash can icon).
4. If you wish to modify the conditions for this rule, then make your modifications.

   The conditions are different for each type of event. For example, if you wish to modify a rule which sends out a notification each time a service request changes to the status of "Closed" then you can modify:

   - The status you wish to trigger the notification.
   - The condition: whether you want to send the notification each time a service request changes to the status or changes from that status.
   - Who you want to notify. For status changes, you have three choices: you can notify all contacts on the service request, the service request owner, or all the owners of service requests that have been linked to the service request via a relationship.
5. If you wish to select a different notification message template for this rule, then click **Update Action Details**.

   The Notification Details page appears.
6. Click **Search for Notification Template** (the searchlight icon).

   The Search and Select: Notification Template page appears.
7. If you wish to restrict your search, then enter a complete or partial name of a template in the Search by field.
8. Click **Go**.

   The page displays the available templates in the Results region.
9. Select the radio button next to the template you wish to use.
10. Click **Select**.

    The template name displays in the Notification Details page.
11. Click **Apply.**
12. If you wish to update another rule, then select the View Notifications Rule link at the top left of the page.

# 1.24 Implementing Oracle eMail Center

Integration with Oracle eMail Center makes it possible for agents to communicate with customers via emails from both the Contact Center and the Service Request windows.

When Oracle eMail Center is implemented, agents can:

- Compose emails to customers and customer contacts by clicking on an icon. Agents can use email templates which supply the contextual information about the customer.
- View the text of emails sent via Interaction History.

## 1.24.1 Guidelines for Setting Up the Oracle eMail Center

Follow these general steps when implementing the Oracle eMail Center:

1. Set up Oracle eMail Center as described in the *Oracle eMail Center Implementation Guide*.
2. Set up email templates for use with the Contact Center and Service Request forms. You can use the mail-merge fields listed in Customer and Contact Mail Merge Fields on page 1-73 and Service Request Mail Merge Fields on page 1-75.
3. To enable email from the Contact Center, set the following system profile options:
   - Customer Care: Outbound Email Enabled

     This profile option sets enables the email buttons on the application's tool bar. A setting of Yes, the default, enables the icons and permits users to use the icons to send emails. A setting of No disables the icons and functionality.
   - Customer Care: Default eMail Template for Outbound Email from Contact Center Form

     Set this profile to the email template you wish agents to use as a default for creating emails. There is no seeded template available, so you must create your own.
   - Customer Care: Default eMail Category for Outbound Email from Contact Center Form

If you wish to restrict the use of eMail Center templates by Oracle Marketing Encyclopedia System (MES) product categories, then set this profile to any of the existing categories.

4. To enable email from the Service Request window, set the following system profile options:

   - Service: Enable Defaulting CC field in Email Header While Sending An Email In Service Request Form

     Setting this profile option to Yes automatically defaults in the CC address line the email addresses of all the service request contacts. If an agent is sending an email to the customer email address, then all of the service request contacts are copied on the email. If the agent is emailing one of the contacts all of the remaining contacts are copied. The application defaults the email addresses only when agents launch the compose window from the Service Request window.

     Setting this profile to No leaves the CC address line blank.

   - Service: Subject Prefix for Outbound Email in Service Request

     Setting this profile to Yes causes the application to populate the Subject line of the email to be populated with the problem summary from the service request. The application defaults the subject line only when agents launch the compose window from the Service Request window.

     Setting this profile to No leaves the subject line blank.

   - Service: Subject Prefix for Outbound Email in Service Request

     Setting this profile to Yes causes the application to populate the Subject line of the email with the problem summary from the service request. The application defaults the subject line only when agents launch the compose window from the Service Request window.

     Setting this profile to No leaves the subject line blank.

   - Service: Default Email Template

     Set this profile to a default email template you have created.

## 1.24.2 Customer and Contact Mail Merge Fields

For setting up email templates for use with the Contact Center, you can use the following customer, contact, and interaction parameters:

| Customer Parameter | Description |
|---|---|
| CUSTOMER_NAME | Customer's name |
| CP_FIRST_NAME | Customer's first name |
| CP_LAST_NAME | Customer's last name |
| CP_PARTY_NUMBER | Party Number |
| CP_PRIMARY_PHONE | Identifying phone number |
| CP_PRIMARY_EMAIL | Identifying email |
| CUST_URL | Identifying URL |
| CP_ADDRESS | Identifying address (the address displayed on Contact Center header) |
| CP_CITY | City for the identifying address |
| CP_STATE | State for the identifying address |
| CP_POSTAL_CODE | Postal Code for the identifying address |
| CP_COUNTRY | Country for the identifying address |
| ACCOUNT_NUMBER | Account number displayed on the header |

| Contact Parameter | Description |
|---|---|
| CONTACT_NAME | Party name |
| CON_FIRST_NAME | First name |
| CON_LAST_NAME | Last name |
| CON_PARTY_NUMBER | Party number |
| CON_PRIMARY_ PHONE | Primary phone |
| CON_PRIMARY_EMAIL | Primary email |
| CONT_URL | Primary URL |
| CON_ADDRESS | Identifying Address displayed on the contact center header |
| CON_CITY | City |
| CON_STATE | State |
| CON_POSTAL_CODE | Postal Code |

| Contact Parameter | Description |
|---|---|
| CON_COUNTRY | Country |
| RELATION | Relationship displayed on Contact Center header. |
| CRITICAL_FLAG | Critical Customer Flag |
| TIMEZONE | Contacts Time Zone |

| Interaction Parameter | Description |
|---|---|
| INTERACTION_ID | Interaction Id |
| DEFAULT_OUTCOME_FOR_INTERACTION | Default outcome for an interaction |
| DEFAULT_RESULT_FOR_INTERACTION | Default result for an interaction |
| DEFAULT_REASON_FOR_INTERACTION | Default reason for an interaction |
| DEFAULT_OUTCOME_FOR_ACTIVITY | Default outcome for an activity |
| DEFAULT_RESULT_FOR_ACTIVITY | Default result for an activity |
| DEFAULT_REASON_FOR_ACTIVITY | Default reason for an activity |

### 1.24.3 Service Request Mail Merge Fields

You can use the following service request mail merge fields in the templates for emails sent from the Service Request window:

| Service Request Parameter | Description |
|---|---|
| SR_CUSTOMER_NAME | Customer name |
| SR_CONTACT_NAME | Contact name |
| SR_ACCOUNT_NUMBER | Account number |
| SR_NUMBER | Service request number |
| SR_STATUS | Service request status |

| Service Request Parameter | Description |
|---|---|
| SR_TYPE | Service request type |
| SR_SEVERITY | Service request severity |
| SR_URGENCY | Service request urgency |
| SR_REPORTED_DATE | Service request incident date |
| SR_GROUP_OWNER | Service request group owner |
| SR_INDIVIDUAL_ OWNER | Service request individual owner |
| SR_ITEM | Item |
| SR_ITEM_DESC | Item Description |
| SR_INSTANCE | Instance |
| SR_SERIAL_NUMBER | Serial number |
| SR_REVISION | Revision |
| SR_TAG | Tag |
| SR_SYSTEM | System |
| SR_CONTRACT | Contract |
| SR_CONTRACT_LINE | Contract Line |
| SR_COMPONENT | Component |
| SR_ COMPONENT _ DESC | Component description |
| SR_ COMPONENT _ REV | Component revision |
| SR_SUBCOMPONENT | Subcomponent |
| SR_ SUBCOMPONENT _ DESC | Subcomponent description |
| SR_ SUBCOMPONENT _ REV | Subcomponent revision |
| SR_SUMMARY | Service request summary |
| SR_RESOLUTION_ SUMMARY | Resolution summary |
| SR_RESOLUTION_BY | Service request resolution by date |

| Service Request Parameter | Description |
|---|---|
| SR_RESPOND_BY | Service request respond by date |
| SR_SUPPORT_SITE | Support site |
| SR_ ESCALATION _ LEVEL | Escalation level |

## 1.25 Implementing Automatic Task Generation for Service Requests

Using descriptive flexfields and Oracle Workflow, you can set up the application to automatically create one or more tasks whenever an agent creates a service request of a specific type. For example, if a repair requires a special type of equipment, you can automatically have the application create a task for the shop to ready this equipment each time an agent creates a service request for a customer who requests that repair.

### Prerequisites

Familiarity with descriptive flexfield implementation. Knowledge of Oracle Workflow programming.

### To set up task generation for service requests:

1. Create a task template specifying all the tasks you wish to launch for a particular service request type. See the Working With Task Templates section of the *Oracle Common Application Components Implementation Guide* (formerly published under the title *Oracle CRM Application Foundation Implementation Guide*).
2. Create a descriptive flexfield as described in Creating a Descriptive Flexfield for Task Generation on page 1-78. This descriptive flexfield is used to associate a specific task template with a service request type.
3. Use the descriptive flexfield to associate a task template you have created with the service request type as described in Associating the Task Template with a Service Request Type on page 1-82.
4. Create an Oracle Workflow process to generate the task(s) using from the template. See guidelines described in Guidelines for Creating an Oracle Workflow Process to Trigger Task Generation on page 1-84.

5. Associate the workflow process you have created with the service request type. See Associating the Oracle Workflow Process Definition with the Service Request Type on page 1-85.

## 1.25.1 Creating a Descriptive Flexfield for Task Generation

Use this procedure to create a descriptive flexfield for task generation. For additional information, please refer to the Defining Descriptive Flexfields Structures section of the *Oracle Applications Flexfields Guide*.

### Prerequisites

Familiarity with descriptive flexfields.

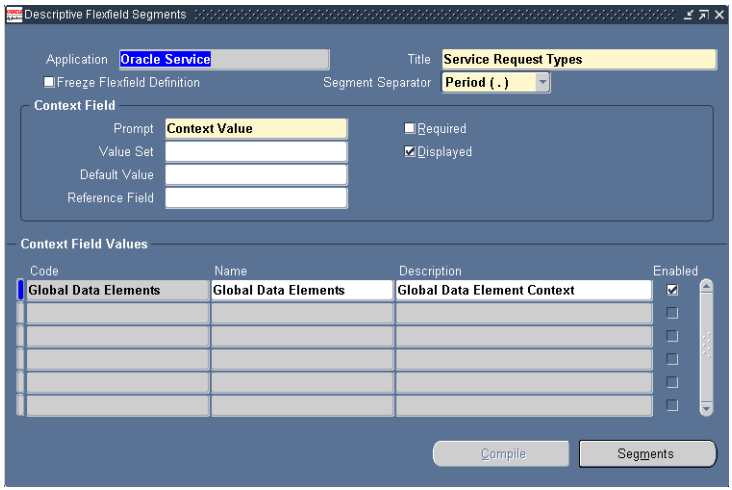
### To create a descriptive flexfield for task generation:

1. Log into Oracle applications under the Application Developer responsibility.
2. Navigate to Flexfield > Descriptive > Segments.
3. Click **Find** in the toolbar.

   The Find Descriptive Flexfield window appears.
4. Enter `%Oracle Service%` in the Find field.
5. Click **Find**.

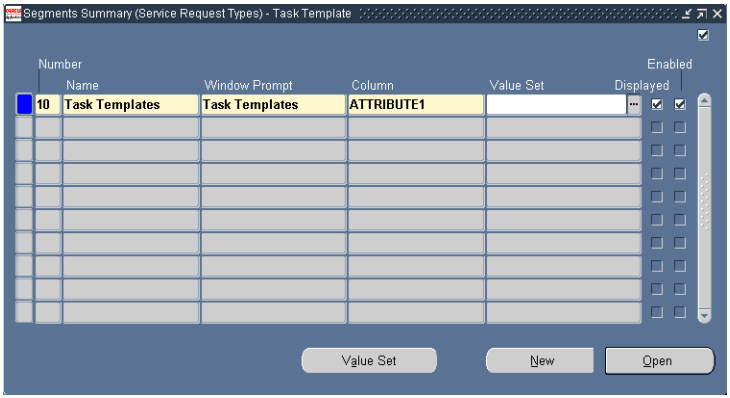
   The window displays the list of service descriptive flexfields.
6. Scroll down and select Service Request Types.

   The Descriptive Flexfield Segments window displays any previously defined segments.



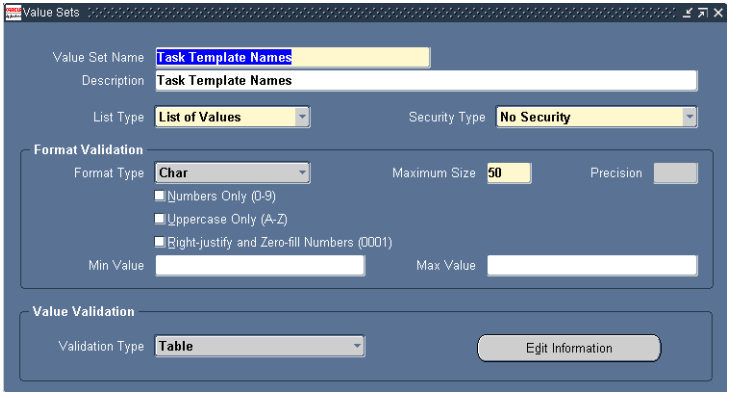

7. Click **New** in the toolbar.
8. Enter `Task Template` in the Code, Name, and description fields.
9. Click **Save**.
10. Using the Default Value List of Values (LOV) select Task Template.
11. Deselect the Displayed check box as you do not need to show the default value for this flexfield.
12. Click **Save**.
13. Click **Segments**.

    The Segments Summary window appears.




14. Enter a number such as 10 in the Number field. The value is unimportant as your flexfield has only one segment.
15. Enter `Task Templates` in the Name and Window Prompt fields. The Window Prompt is the value users see.
16. Using the Column LOV, select ATTRIBUTE 1. There are 15 possible attributes.
17. Click **Value Set**.

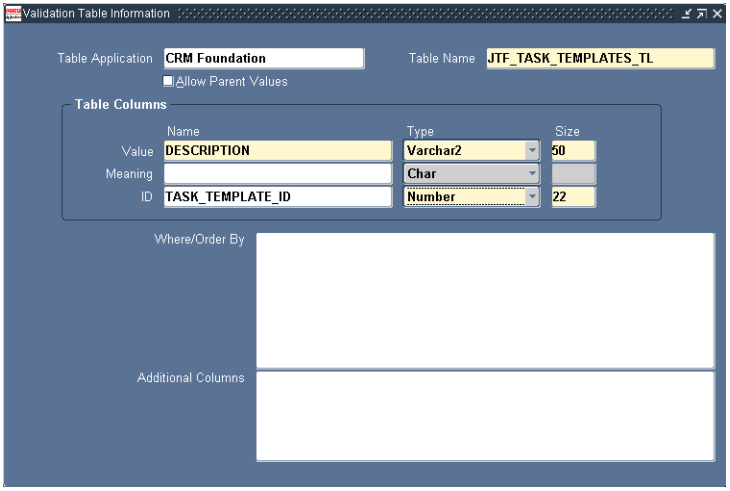
    The Value Sets window appears. This window controls the list of values of the task templates you have created.




18. Enter `Task Template Names` in the Value Set Name and Description.

19. Select List of Values from the List Type drop-down list.
20. Select No Security from the Security Type drop-down list.
21. Select Char from the Format Type drop-down list in the Format Validation region.
22. Enter 50 or some other number for the length of the field in the Maximum Size field.
23. Click **Edit Information**.

    The Validation Table Information window appears. Here is where you specify the task templates for the list of values that will populate the flex field.




24. Using the Table Application LOV, select CRM Foundation.
25. Using the Table Name LOV, select JTF_TASK_TEMPLATES_TL.
26. Using the Value LOV, select DESCRIPTION.
27. For Type, enter Varchar 2.
28. For Size, enter 50 or the same value you entered as the maximum size earlier.
29. Using the ID LOV, enter TASK_TEMPLATE_ID.
30. Using the Type LOV, enter Number.
31. Enter 22 in the Size field.

32. Click **Save**.
33. Navigate back to the Segments Summary window.
34. Using the Value Set LOV, select Task Template Names (the value set you have just defined).
35. Click **Save**.

## 1.25.2 Associating the Task Template with a Service Request Type

Use this procedure to associate a task template with a service request type using the descriptive flexfield you have created.

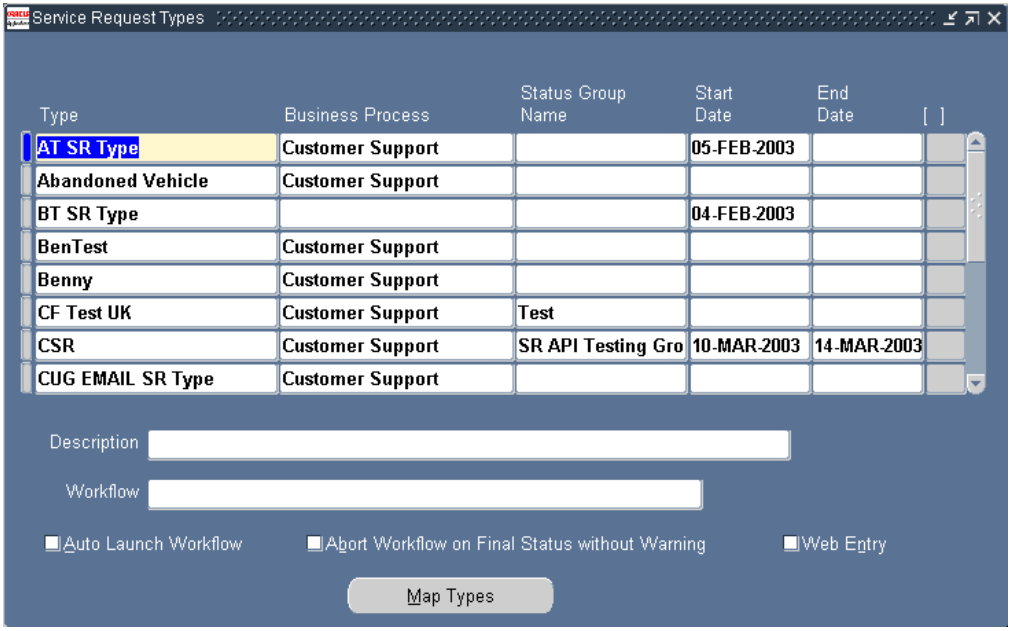
### Prerequisites

You must create the descriptive flexfield and service request types first.

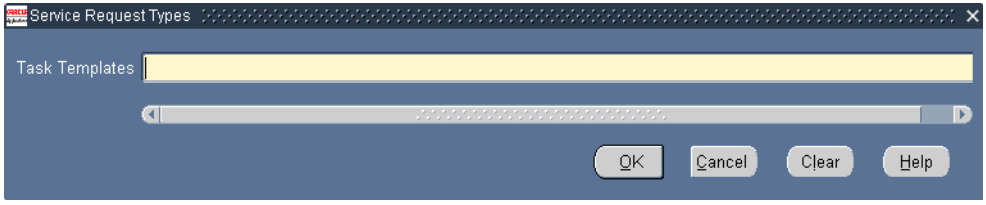
### To associate a task template with a service request type:

1. Log into Oracle applications under the Customer Support responsibility.
2. Navigate to Setup > Service Requests > Request Types.

   The application displays the Service Request Types window.




3. Click on the flexfield to the right of the End Date column for the service request type you wish to set up.
4. The Service Request Types flexfield window appears.




5. Use the list of values to select the task template you wish to use for this service request type.
6. Click **OK**.

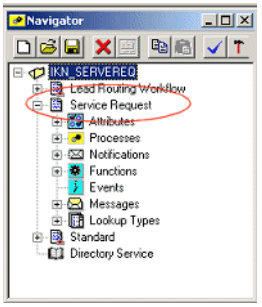
7. Click **Save**.

## 1.25.3 Guidelines for Creating an Oracle Workflow Process Definition for Task Generation

Use these guidelines for creating a workflow processes for generating tasks using the task templates you have defined. For detailed information on creating workflow processes, please see the *Oracle Workflow Guide*.

### Guidelines

- Create your process definition under the Service Request item type folder.




- Define a function using the Create Task from Template API (jtf_tasks_pub.create_task_from_template). This function must create a task as defined in the template associated with the service request type. Please refer to the *Oracle Common Application Components API Reference Guide* for further information on task APIs.
- If you wish to launch the workflow automatically whenever an agent creates a service request, then you should consider including a function that will delay the task creation for a period of time. This is because the application permits agents to change the service request type after the service request is created and does not check or abort any workflows when they do.

  Suppose, for example, you set up the service request type “Equipment Repair” to create a task instructing the shop to ready the required repair equipment. This task gets created even if a customer calls with a billing question and the agent selects this service request type by a mistake. The application does not abort the workflow or close the task if the agent changes the service request type to “Billing”.

## 1.25.4 Associating the Oracle Workflow Process with the Service Request Type

Use this procedure to associate the workflow process you have created for task creation with the service request type.

### Prerequisites

Before you undertake this step, you must:

- Create the descriptive flexfield, the service request type, and workflow process definition.
- Associate the descriptive flexfield with the service request type.

### To associate the Workflow process definition with the service request type:

1. Log into Oracle applications under the Customer Support responsibility.
2. Navigate to Setup > Service Requests > Request Types.

   The application displays the Service Request Types window.
3. Select the service request type.
4. Use the Workflow list of values to select the Workflow process.
5. If you wish the application to automatically launch the Workflow each time a service request of this type is created, then select the Autolaunch Workflow check box. Leaving this check box unselected, requires the agent to launch the workflow by making a selection from the Tools menu.
6. You can have the application abort any active workflow instance (but not any tasks already created) when the agent closes the service request by:
   - Setting up service request statuses with the Final flag. See Setting Up Service Request Statuses on page 1-6.)
   - Setting the Abort Workflow on Final Status without Warning check box.

     Selecting this check box causes the application to automatically abort the workflow whenever the agent sets the service request to a status flagged as Final. If you do not select this check box, then the agent always receives a warning if a workflow is about to be terminated and has a chance to cancel the abort.

# 1.26 Setting Up the List of Values for the Language Field

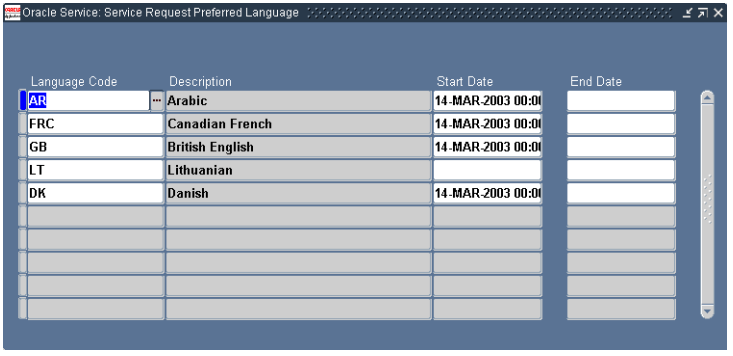
Use this procedure to set up the list of values for the Language field in the Service Request window header. Agents can use this field to indicate a customer's preferred language for a specific service request.

> **Note:** The language preference agents enter using this LOV remain specific only to individual service requests. The preference is not stored in the customer record itself. This means that selecting a language preference in a service request does not modify the customer's overall language preference which is entered in the Preferred Language field in the Party Information tab of the Contact Center.

### To set up the list of values for preferred languages:

1. Navigate to Setup > Service Requests > Preferred Language

   The Oracle Service: Service Request Preferred Language window appears.




2. Use the Language Code LOV to select the languages you wish to add to the list of values in the Language field.

   The language name the agent will see in the LOV appears in the Description field.

3. Optionally, enter dates if you wish to limit the availability of your entry.

4. Click **Save**.

# 2

# Implementing Customer Care

This chapter presents a detailed description of the setup and implementation steps required to successfully implement the Customer Care module of Oracle TeleService.

> **Note:** Please use the Customer Support responsibility for performing the setup procedures detailed in this chapter unless indicated.

This following topics are covered:

# 2.1 Setting Up Customer Profiles

Customer profiles make it possible for you provide agents with summaries of key customer information, such as the number of open service requests or expiring contracts.

Agents can view the summarized customer information in two places:

- The Dashboard tab

  Available in the Contact Center (Oracle TeleService) and E-Business Center (Oracle TeleSales).

- The Profile window

  Available in the Service Request and Override Customer Criticality windows in Oracle TeleService and the Repair Order window in Oracle Depot Repair.

Setting up customer profiles is also a prerequisite for implementing relationship plans. While the information on the Dashboard tabs and Customer Profile windows summarizes key information about customers, they require agents to check that information periodically. Relationship plans, described in a separate section of this guide, automatically display an alert message or a script when a customer meets certain conditions based on your customer profile setup.

---

**Note:** Customer profiles are not the same as system profile options.

---

For a brief overview of customer profile concepts, see About Customer Profiles on page 2-3

For an overview listing the major steps of the setup procedure, see Overview of Customer Profile Setup on page 2-6.

Details of individual steps are covered in the following topics:

- Define Profile Ratings on page 2-7
- Associating Ratings with Colors on page 2-9
- Defining Profile Check Categories on page 2-10
- Defining Profile Check Variables on page 2-11
- Defining Profile Checks on page 2-18
- Defining Drill-Downs for a Profile Check on page 2-13

## 2.1.1 About Customer Profiles

Customer profiles make it possible for you to provide agents with:

- Summaries of key customer information in a graphical format
- The ability to drill down to the details of the information the profiles highlight

You can specify access to customer profiles by application and by responsibility, so that agents working on different types of customer problems receive different customer views.

### 2.1.1.1 Summaries of Key Customer Information

Customer profiles make it possible for you to create summaries of key customer information, called profile checks.

Profile checks can be of two types:

- Variable
- Yes and No (Boolean)

#### Variable

The variable profile checks tell agents the level of a particular customer attribute, such as the number of open service requests or the number of booked orders. Each profile check displays a number (the number of open service requests or expiring contracts, for example), a rating (high, medium, low) and a color (green, yellow, and red, for example). The rating and color together comprise a rating label.

#### Yes and No (Boolean)

Boolean profiles compare the level of a profile check variable against a constant or an expression with one or more variables and returns either a Yes or No. For example: "Does the customer have more than one open escalation?", "Does the customer have a credit hold?".

### Graphical Format

Agents can view the summarized customer information in two places:

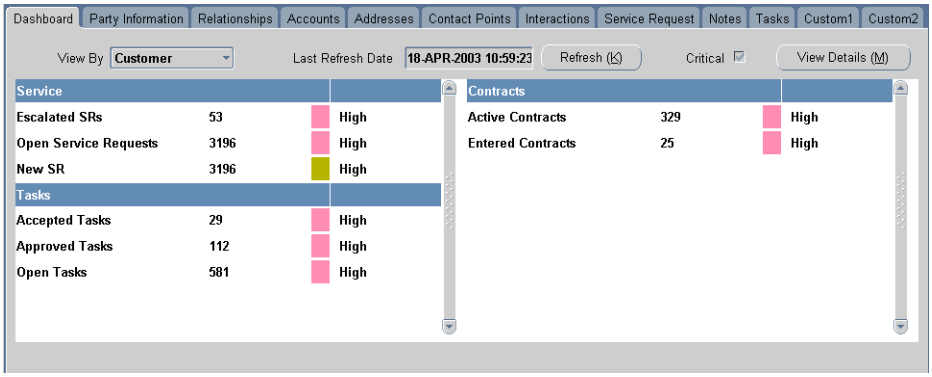
- The Dashboard tab

  The Dashboard tab displays the summarized customer information organized in distinct regions called dashboard groups. These regions are separated by headings called categories.

- The Profile window

  The Profile window displays a list of select profile checks in the order you determine. Profile windows do not use category headings.

Here is a sample Dashboard on the Contact Center showing key customer information organized in three dashboard categories: Service, Tasks, and Contracts.




Agents can filter their views of profile checks by account or by customer by making a selection from the View By drop-down list.

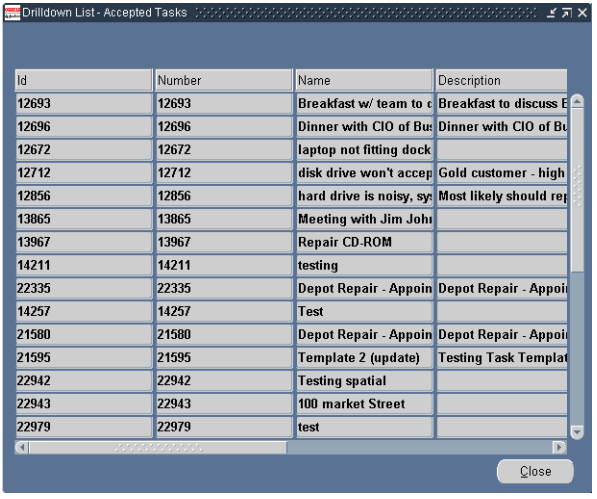
Profile windows display only select individual profile checks. Here is a sample Profile window:


### 2.1.1.2 The Ability to Drill Down to Details

You can set up customer profiles to make it possible for agents to drill down from a variable profile check in a Dashboard tab or customer Profile window to a detailed list of all the items in the Drill-Down List window. This is called a first-level drill-down.

During implementation you can select which fields agents see in this window. Boolean (Yes or No) profile checks do not permit drill-downs.

Here is a sample first-level drill-down window:




Agents can drill down to the individual item in this window by clicking on one of the displayed fields.

The item displays in an application window. This is called the second-level drill-down.

## 2.1.2 Overview of Customer Profile Setup

Use this overview of major setup steps to guide you through the customer profile setup.

### To set up customer profiles:

1. Set up the rating labels (the wording and colors) you will use with your profile checks and the categories you wish to use to organize the profile checks on the Dashboard tab:
   a. Navigate to Setup > Customer Care Lookups.
   b. Define the general labels you wish to use to classify customer information for agents. For example, you may wish to classify customers into three categories, Low, Medium, and High. See Define Profile Ratings on page 2-7 details.
   c. Navigate to Setup > Customer Management > Customer Profiles.
   d. On the Preferences tab, associate the profile ratings you have defined with color codes to create rating labels. For example, you may wish to display red next to the word High. See Associating Ratings with Colors on page 2-9.
   e. On the same tab, define the headings (categories) which organize the profile checks on the Dashboard. See Defining Profile Check Categories on page 2-10.
2. Navigate to Setup > Customer Management > Customer Profiles.
3. On the Profile Variable tab, define the variables you wish to use in your profile checks. This is where you enter the SQL queries that retrieve and summarize your customer data. You can use the same variable in multiple profile checks.

   Agents can filter customer information displayed on the Dashboard by account or by customer. If you are displaying the profile check in the Dashboard and wish to take advantage of this feature, then you can determine the appropriate view by selecting the Customer or Account radio buttons.

   For details see Defining Profile Check Variables on page 2-11.
4. On the Profile Checks tab, define all of the profile checks you want to use to summarize customer information in the Profile windows and Dashboard tabs.

You get to select which agents see them in a subsequent step. See Defining Profile Checks on page 2-18.

5. On the Drilldown tab, set up the two levels of drill-downs your agents will use to view the details of the customer information summarized by the profile check. The first level provides a window with a list of individual records; the second level displays the individual record itself. See Defining Drill-Downs for a Profile Check on page 2-13.
6. On the Profile Groups tab, define the groupings of profile checks for display in the customer Profile window. By matching these profile groups to application modules and responsibilities in a later step, you can create different customer overviews for different groups within your organization. See Grouping Profile Checks for Display in Customer Profile Windows on page 2-22.
7. On the Dashboard Groups tab, define the groupings of profile checks for display in the Dashboard tabs of the Contact Center and E-Business Center windows. By matching these profile groups to application modules and responsibilities in a later step, you can create different customer overviews for different groups within your organization. See Grouping Profile Checks for Display in Dashboard Tabs on page 2-24.
8. On the Preferences tab, specify which application modules and which application responsibilities can view the profile groups and dashboard groups you have created. See Determining Access by Module and Responsibility on page 2-26.
9. Set up the concurrent program that will automatically refresh the Dashboard tab and Profile views of customer information. See Running the Customer Profile Engine on page 2-28.
10. You have the application refresh the customer information on the dashboard each time an agent displays a customer by setting the profile Customer Care: To Allow Auto Refresh Facility for Dashboard to Yes. A setting of No requires the agent to click the Refresh button to update the display. See Setting Automatic Refresh of Profiles on the Dashboard on page 2-29 for more details.

## 2.1.3 Defining Profile Ratings

Use this procedure to modify or create the labels used to describe the ranges of values of your profile checks. The application provides three seeded values: High, Medium or Low. These labels are called profile ratings.

Profile ratings are implemented in Customer Care using standard Oracle Applications lookup codes. Profile rating codes are implemented with the User

access level. This means that you can create new labels and modify the seeded values.

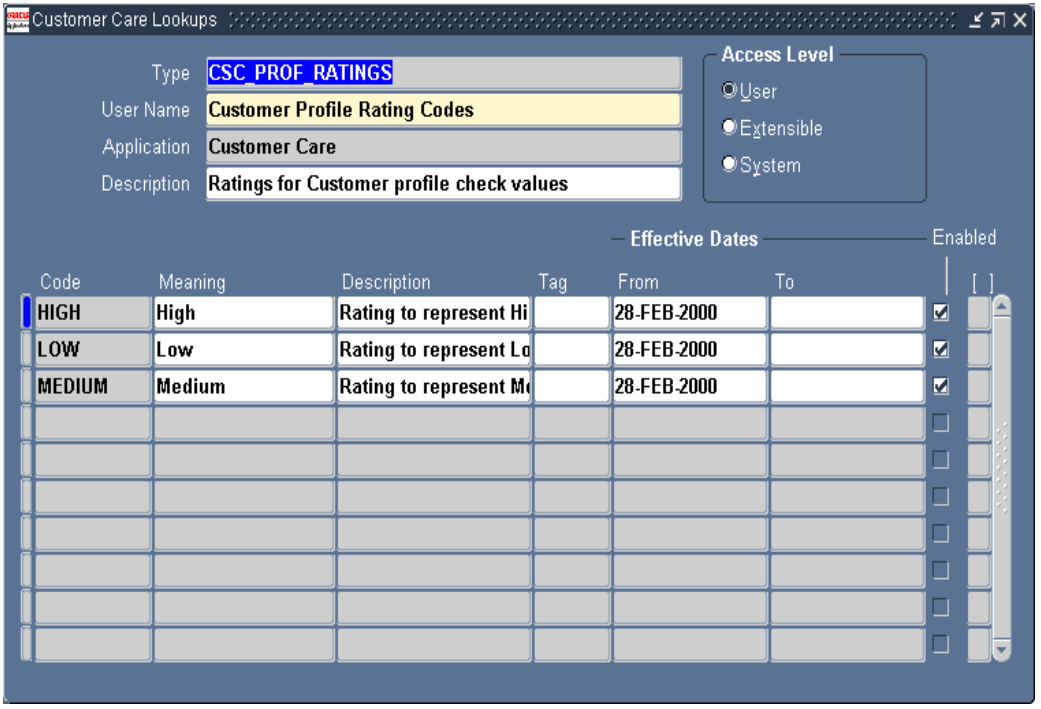
### Prerequisites

None

### To define profile ratings:

1. Navigate to **Setup > Customer Care Lookups**.

   The Customer Care Lookups window appears.




2. In the Type field perform a search using the Query by Example method on `CSC_PROF_RATINGS`.

   The window displays the three predefined ratings, High, Medium, and Low, and any others your organization has defined.

3. To create a new rating code:

a. Place your cursor in a row and click **New** in the toolbar.

b. Enter a name in the Code field. You will be associating this name while building the profile checks.

c. Enter a name in the Meaning field. This is the wording agents will see on the Dashboard tab and Customer Profile window.

d. Optionally enter a description in the Description field.

   The Tag field is not used.

e. If you wish to control the dates this rating code is to be effective, then enter dates in the Effective Date fields.

4. Verify that the Enabled check box is selected.
5. Save your work.

### See Also:

- Defining Profile Variables
- Defining Profile Checks

## 2.1.4 Associating Ratings with Colors

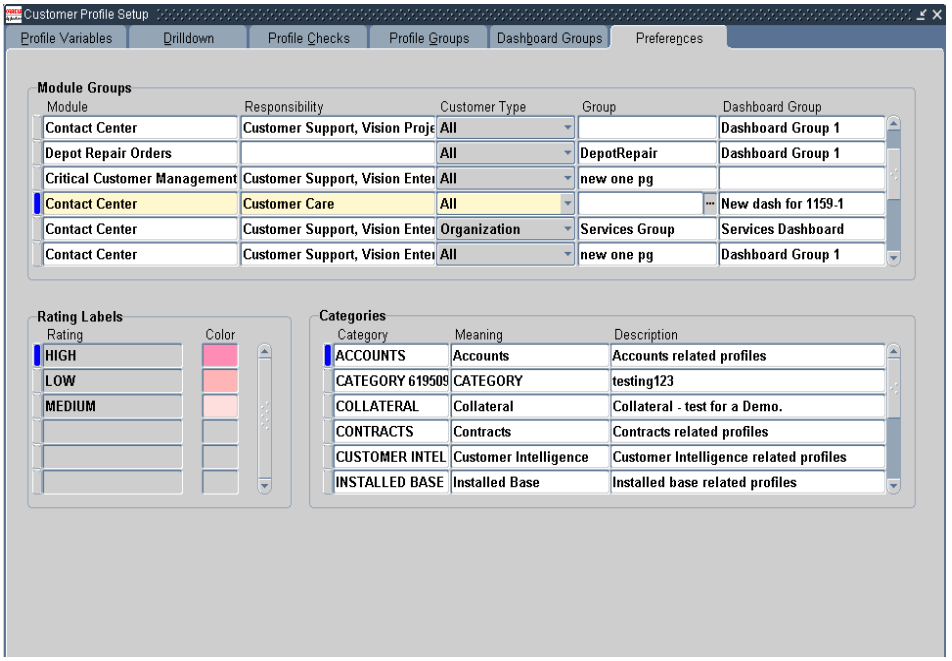
Use this procedure to associate colors with the ratings you defined to create a rating label. It is here that you specify, for example, that the label High will always display in the Dashboard and Customer Profile windows together with the color red.

### Prerequisites

You must define profile rating before you can associate them with a color.

### To associate ratings with colors:

1. Navigate to **Setup > Customer Management > Customer Profiles**.
2. Select the Preferences tab.




3. In the Rating Labels region, select the rating you wish to associate with a color from the Rating list of values. The Rating field displays the Code value from the lookup, not the value users will see.
4. Select a color to associate to the rating from the Color list of values.
5. Save your work.

## 2.1.5 Defining Profile Check Categories

Use this procedure to define the categories used in the Dashboard tab to group profile checks. For example, open, escalated, and accepted tasks can all be grouped under the Tasks category.

### Prerequisites

None

### To define categories:

1. Navigate to **Setup > Customer Management > Customer Profiles**.
2. Select the Preferences tab.
3. In the Categories region of the tab, for each category you wish to define:
   a. Enter a name for the category in the Category field. This internal name will not display on the Dashboard tab.
   b. Enter the wording that will display on the Dashboard tab in the Meaning field.
   c. Optionally enter a description for the category in the Description field.
4. Save your work.

## 2.1.6 Defining Profile Check Variables

Use this procedure to define the variables you wish to use in your profile checks. The variables contain the SQL statement that is executed to retrieve the customer profile check value. You can use the same variables in multiple profile checks.

Your SQL statements can contain two bind variables:

- :party_id - Use this variable if the SQL statement refers to a party
- :cust_account_id - Use this variable if the SQL statement refers to a customer account

No other bind variables are supported.

Your SQL statement must be constructed so that only a single row is returned by the query.

For a detailed description of the database objects you can include in your SQL statement, please refer to the Electronic Technical Reference Manual (eTRM) for Customer Care that is available from http://metalink.oracle.com. Navigate to Top Tech Docs and use the link titled Applications Electronic Technical Reference Manuals. Customer Care database objects are stored in the folder named CSC.

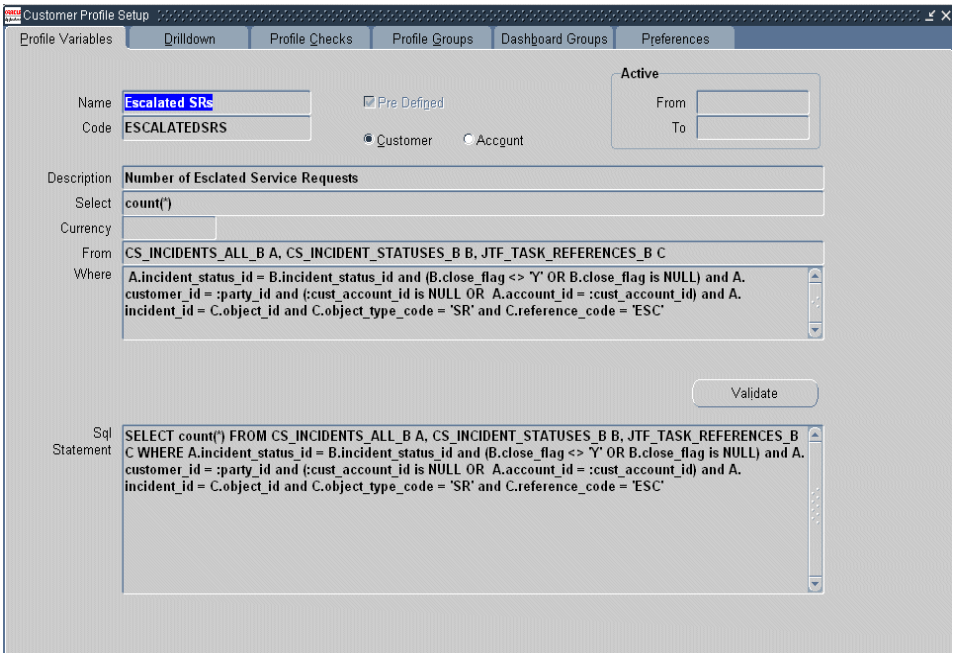
### Prerequisites

None

### To define a profile check variable:

1. Navigate to Setup > Customer Management > Customer Profiles.

The Customer Profile Setup window appears.




2. Select the Profile Variables tab.
3. Click **New** on the toolbar.
4. Enter a name in the Name field.
5. Enter a unique name in the Code field.
6. If you are displaying the information from this variable in the Dashboard tab, then select either the Customer and Account radio buttons to determine the agent view in which the information will appear.

   Agents can select filter their Dashboard views by accounts or customers. Your setup determines which view displays the information.
7. If you wish to restrict the use of this variable, then enter dates in the Active: From and Active: To fields.
8. Optionally enter a description of the profile variable in the Description field.
9. Enter the appropriate components of the SQL statement in the following fields:

- Select (mandatory field).
- Currency (optional field).
- From (mandatory field).
- Where (mandatory field).
- Other (optional, used for including clauses such as Group By).

10. To validate the SQL statement, click **Validate**.
11. Save the variable by clicking **Save** in the toolbar.

    Your SQL statement is validated automatically when you save. You cannot save an invalid SQL statement.

### See Also:

- Setting up Customer Profile
- Defining Profile Ratings
- Defining Profile Checks
- Defining Profile Groups

## 2.1.7 Defining Drill-Downs for a Profile Check

Use this procedure to define the drill-down windows for a profile check. There are two levels of drill-down windows. The first level lists the items summarized by the profile check. The second-level drill-down window is the application window which displays the details of an individual record. An agent displays an item in the application window by double clicking on one of the fields in the summary list.

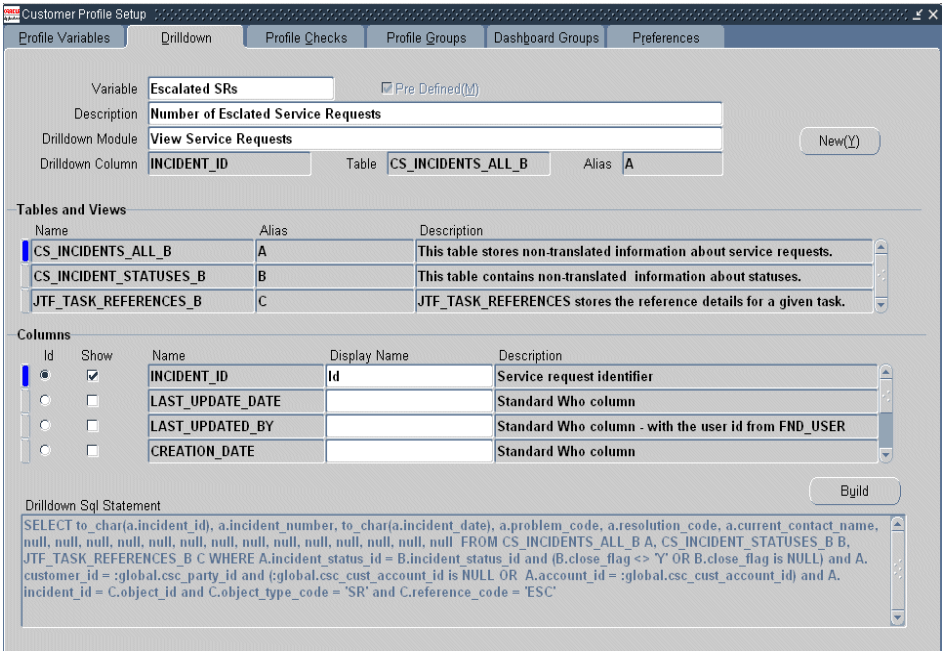
### Prerequisites

You must create the variables for the profile check before you can define drill-down windows.

### To define drill-down windows:

1. Navigate to Setup > Customer Management > Customer Profiles.

   The Customer Profiles window opens.
2. Select the Drilldown tab.




3. From the Variable field, select the profile variable for which you want to define the drill-down.

   The Description field as well as the Tables and Views region display information about the selected profile variable.

4. Using the Drilldown Module list of values, select the application module you wish to use for the second-level drill down. If the module you wish to use is not listed, then you can add it to the list by following the steps outlined in Enabling Application Windows for Drill-Down on page 2-15.

5. Select up to 20 columns you wish to display in the first-level drill-down list:

   a. Select one of the tables in the Table and Views region.

      The Columns region displays the available columns for that table.

**Note:** Only columns from registered tables and views are displayed. Column names are retrieved from FND_COLUMNS and FND_VIEWS. If the columns you wish to use are not visible, check to see if the table or view is registered.

**b.** Select the Show check box for the columns that you wish to display in the first-level drill-down.

**c.** For the columns you wish to display, enter a user-friendly name in the Display Name field. The text you enter here is what agents see in the first-level drill-down window.

6. Select one of the columns in the first-level drill-down as the column that agents will use to drill-down to the application window (the second-level drill-down). For example, if you select the service request number, an agent can view the details of the service request in the Service Request window by double-clicking on the service request number in the summary list.

   **a.** Select the table in the Table and Views region.

   The Columns region displays the available columns for that table.

   **b.** Select the ID radio button for the column that you wish use for the drill-down.

   **Note:** You can select only one ID column for a profile variable.

   **c.** Click Save in the toolbar.

   The column name of the ID you have selected, displays in the Drilldown Column field at the top of the tab.

7. Optionally, click **Build** to generate and display the SQL for the drill-down.

   The SQL statement is validated when you save. You cannot save an invalid SQL statement.

## 2.1.8 Enabling Application Windows for Drill-Down

Use this procedure to enable application windows for drill-down.

### Prerequisites

You must define the form object first in the Form Functions window. You can access this window by logging in under the Systems Administrator responsibility and navigating to Application > Function.

### To enable application windows for drill-down:

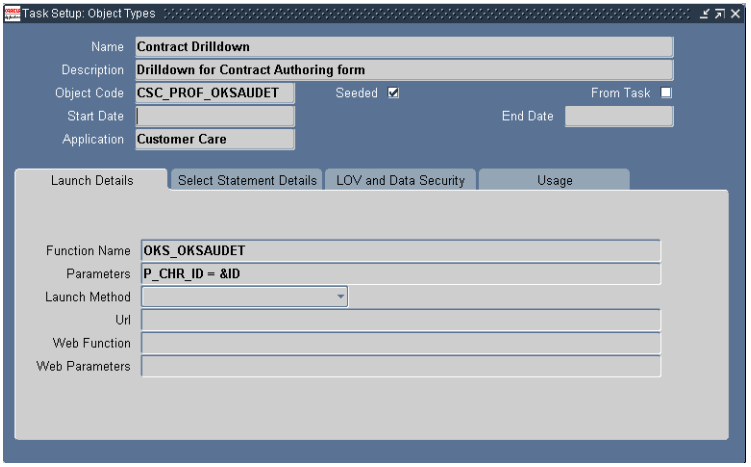
1. Navigate to Setup > Customer Management > Customer Profiles.

   The Customer Profiles window opens.

2. Select the Drilldown tab.

3. Click **New**.

   The Task Setup: Object Types window opens.




4. Enter a name for the drill-down module object type you are creating in this procedure in the Name field.

   **Note:** The name you enter here is for your reference only. It does not display in the Drilldown Module list of values.

5. Enter a description for the drill-down module in the Description field.

6. Enter a value in the Object Code field that begins with `CSC_PROF`, for example: `CSC_PROF_OKSAUDET`.
7. You must know both the function name and the window name displayed for the user. The function name, for example, `OKS_OKSAUDET, is` what you enter in the Function Name field in this window. You need to know the associated User Function Name: `Author Service Contracts` because this is what gets displayed in the Drilldown Module list of values on the Drilldown tab. Use the following procedure to find either name:
   a. Switch responsibility to Systems Administrator.
   b. Navigate to Application > Function.

      The Forms Functions window appears.
   c. Using the Query by Example method, display the function either using the Function or User Function Name fields.
   d. Note down the names of both fields.
8. Enter the function name in the Function Name field. For example, `OKS_OKSAUDET`.

   The Drill Down Module list of values displays the User Function Name associated with this form function name.
9. Enter the parameters.
10. Select Customer Care from the Application Name list of values.
11. Entry in the Usage region is optional. You can select CUSTOMER_CARE from the list of values, which also populates the Object User field. The Seeded check box indicates whether the module is predefined.
12. Save your changes.
13. Close the form to return to the Drilldown tab.

    The new application window is available in the Drilldown Module field list of values.

### See Also:

- Defining Profile Variables
- Defining Profile Checks
- Defining Profile Groups

## 2.1.9 Defining Profile Checks

Use this procedure to define profile checks.

You can use the predefined profile checks as a guide. See Predefined Profile Checks on page 2-21.

### Prerequisites

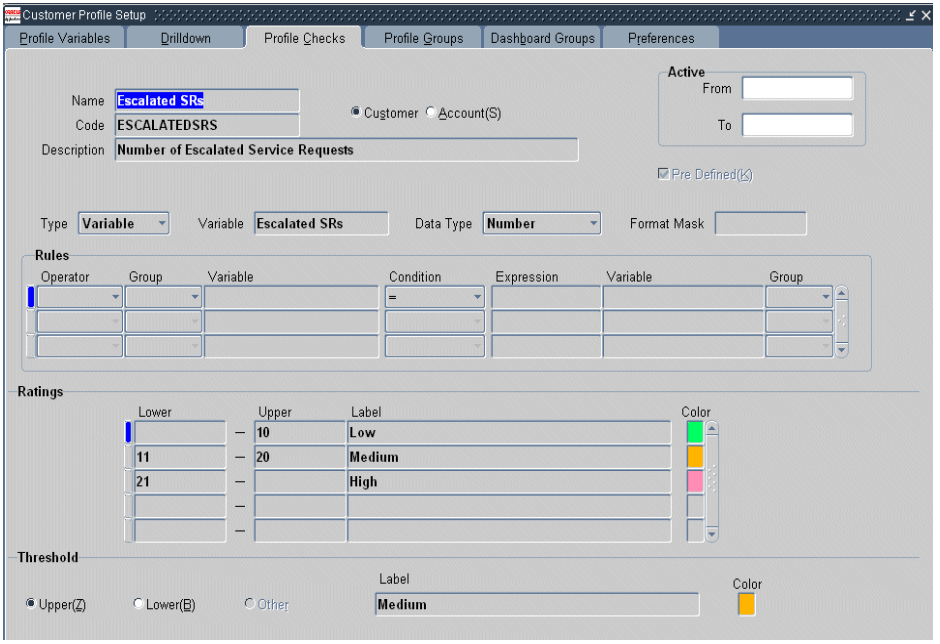
You must define profile variables, ratings, and rating labels first.

### To define profile checks:

1. Navigate to Setup > Customer Management > Customer Profile Setup.

   The Customer Profile Setup window appears.

2. Select the Profile Checks tab.




3. Enter a name for the profile check in the Name field. This is the wording the agents see in the Dashboard tab or Profile window.

4. The Code field defaults to the name you enter in the Name field. You can substitute the value in the Code field with another unique value.
5. Enter a description for the profile check in the Description field.
6. Enter the date when the profile check is to be in force by specifying the Active From and Active To fields.
7. If you wish to create a variable profile check, then select **Variable** from the Type drop-down list and follow the steps described in Defining Variable Profile Checks on page 2-19.
8. If you wish to create a Boolean (Yes or No) profile check, then select **Yes or No** from the Type drop-down list and follow the steps described in Defining Boolean Profile Checks on page 2-20.
9. Save your work.

### 2.1.9.1 Defining Variable Profile Checks

Perform these steps to create a variable profile check after you have completed the basic steps described earlier:

1. In the Variable field, select the variable on which the profile check is to be based. This is one of the variables you have defined in the Profile Variables tab.
2. In the Data Type field, enter the data type for the variable you have selected. The valid choices are Number, Char, and Date.
3. If appropriate, enter a format mask in the Format Mask field.
4. In the Ratings region, define value ranges and their labels. For each range:
   a. Enter a Lower and Upper number. The range you enter here cannot overlap another range.
   b. Select an appropriate label. This is one of the rating labels you have created which combine the rating and color code.

   Omit entering the low value for the first range and the high value for the last range as shown in the example below. In the example, leaving the Upper Value for the High range blank ensures that your profile check displays for customers of any high value above 20.

| Lower Value | Upper Value | Label |
|---|---|---|
| Leave blank | 10 | Low |

| Lower Value | Upper Value | Label |
|---|---|---|
| 11 | 20 | Medium |
| 21 | Leave blank | High |

5. If you wish, you can turn the display of the profile on only when the number is below or above a threshold.

   For example, you may wish call center agents to see information on the number of open service requests for a customer regardless if the number is low or high. But managers may care only about customers with high numbers. By specifying the upper number in the Medium range as your trigger, you can display only the high range to the group of your choice. To define a threshold for a profile check:

   a. Select either the Upper or Lower radio button depending on whether you wish to use the lower or higher number for the range as the threshold.

   b. Using the Label list of values, select the range below or above which you wish the display to be triggered.

      Using the above example, if you wish to display the profile check only when the value is High, then you would select the Upper radio button and select Medium from the Label list of values.

   c. For the threshold to affect the display of information you must select the Display on Threshold check box when you define profile groups and dashboard groups. See Grouping Profile Checks for Display in Customer Profile Windows on page 2-22 and Grouping Profile Checks for Display in Dashboard Tabs on page 2-24 for more information.

### 2.1.9.2 Defining Boolean Profile Checks

Perform these steps in addition to the basic steps described earlier to create a Boolean profile check:

1. Using the Type drop-down list, select **Yes or No**.
2. In the Rules region, build a logical expression (rule) comparing an existing profile variable with a constant or an expression.

   For example, the seeded Critical Customers profile check evaluates the expression `Open Service Requests > 5`.
3. Save the profile check you have defined.

**See Also:**

- Defining Profile Ratings
- Defining Profile Variables

### 2.1.9.3 Predefined Profile Checks

Oracle TeleService provides the following 38 predefined profile checks you can use as a guide to define your own. All but Critical Customer are variable profile checks:

> **Note:** Do not modify seed data. If you want to make modifications, please make a copy and modify your copy.

- Accepted Tasks
- Active Contracts
- Approved Tasks
- Booked Orders
- Cancelled Contracts
- Cancelled Orders
- Cancelled Tasks
- Critical Customer
- Entered Contracts
- Escalated Defects
- Escalated Service Requests
- Expired Contracts
- Expiring Contracts
- Installed Base Size
- Loyalty
- On-Hold Contracts
- On-Hold Tasks
- Open Defects
- Open Orders

- Open Service Request
- Open Tasks
- Profitability
- Rejected Tasks
- Revenue
- Satisfaction
- Terminated Contracts

## 2.1.10 Grouping Profile Checks for Display in Customer Profile Windows

Use this procedure to group profile checks for display in the customer Profile window. By associating the groups you create here with application modules and responsibilities you can provide different groups of agents with different customer views in the Profile window.

### Prerequisites

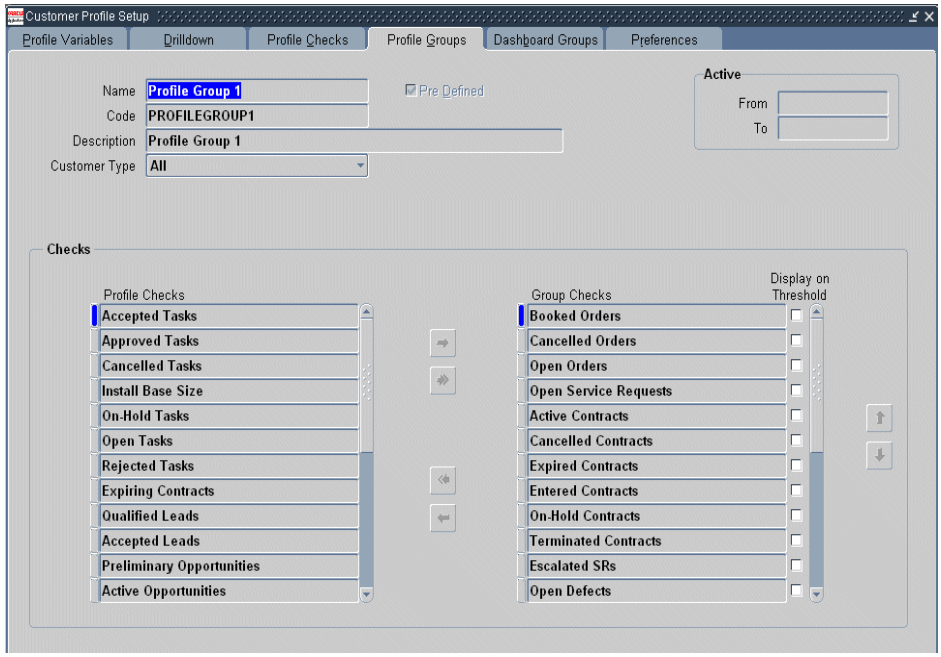
You must define profile checks before you can define Profile Groups.

### To define profile groups:

1. Navigate to Setup > Customer Management > Customer Profiles.

   The Customer Profile Setup window appears.

2. Select the Profile Groups tab.




3. In the Name field, enter a name for the profile group.

   All the profile checks you have defined are displayed in the Checks region.

4. The Code field defaults based on the value you entered in the Name field.

5. In the Description field, optionally enter a description for the profile group you are defining.

6. In the Customer Type field, select the customer type you want to associate with the profile group.

   Selecting All, the default, displays the profile group you are creating for both persons (consumers) and Organizations.

   If you wish to restrict this profile group for display when agents are handling a specific customer type, then select either Person or Organization.

7. You can enter dates in the Active From and Active To fields to limit the availability of this profile group.

8. In the Checks region:

**a.** Select the appropriate profile check from the Profile Checks column and move it to the Group Checks column. Use the right and left arrow buttons to move the profile checks between the columns. Use the up arrow and down arrow buttons to move the selected group check up or down in the list.

**b.** Leave the Display on Threshold check box unselected, unless:

* You have set up your profile check with a threshold.
* You wish this profile check to display only when this threshold is crossed.

For an explanation of using thresholds, please see Defining Variable Profile Checks on page 2-19 for additional information.

**9.** Save the Profile group.

### See Also:

- Defining Profile Ratings
- Defining Profile Variables
- Defining Drilldowns
- Defining Profile Checks
- Defining Dashboard Groups
- Defining Rating Labels
- Defining Categories

## 2.1.11 Grouping Profile Checks for Display in Dashboard Tabs

Use this procedure to group profile checks to create different Dashboard tab views for agents by application module and by responsibility.

### Prerequisites

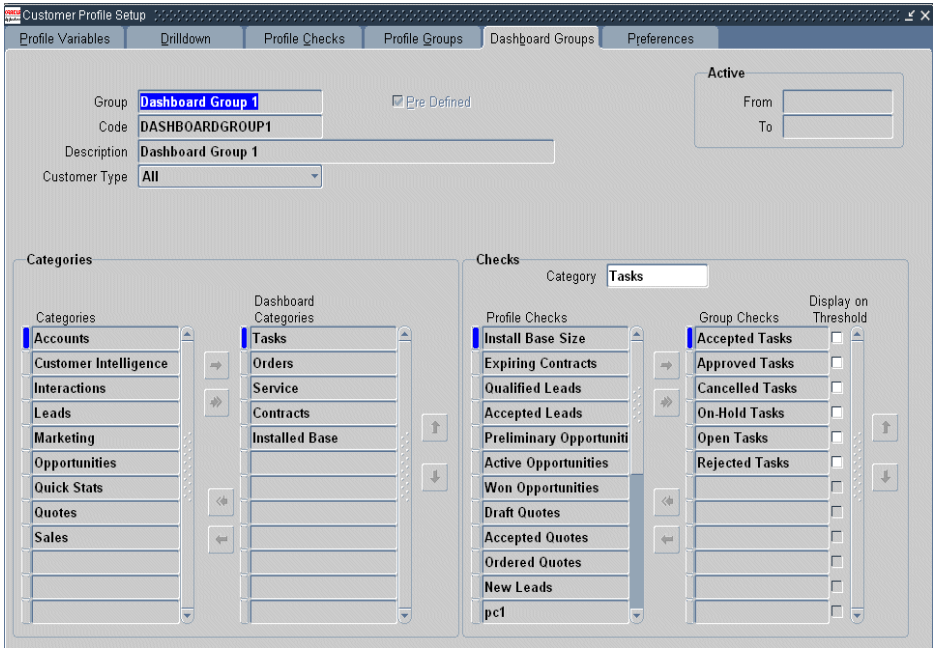
You must define profile checks and categories before you define dashboard groups.

### To define dashboard groups:

**1.** Navigate to Setup > Customer Management > Customer Profiles

The Customer Profile Setup window appears.

2. Select the Dashboard Groups tab.




3. Enter a name for the Dashboard Group in the Group field.
4. The Code field defaults based on the value you entered in the Group field.
5. Enter a description for the Dashboard Group in the Description field.
6. From the Customer Type drop-down list, select the customer type for this dashboard group.

   The default is All, but you can restrict the usage of groups to either Organizations or Persons. Restricting a Dashboard group to a customer type means that the information will display only for customers of that type.
7. If you wish to restrict the use of this dashboard group for a specific time period, then enter dates in the Active From and Active To fields.
8. Select the categories that will organize the information on the Dashboard tab:
   a. In the Categories region, move the categories from the Categories column to the Dashboard Categories column using the right and left arrow buttons.

b. Order the categories in the order you wish them to display in the Dashboard. Use the up-arrow and down-arrow buttons to move a category up or down in the list.

9. Save your changes so your categories are available in the next step.

10. In the Checks region, specify which profile checks you wish to display under each category:

    a. Select the dashboard category you wish to set up from the Category list of values.

    b. Select the profile checks you wish to use by moving them from the Profile Checks column to the Group Checks column. Use the right and left arrow buttons to move the profile checks between the columns.

    c. Determine the display order of the profile checks by ordering those you have selected (the Group Checks column). Use the up-arrow and down-arrow buttons to move the selected profile check up or down in the list.

11. If you have set up a particular profile check with a threshold and wish to restrict the display using that threshold, then select the Display on Threshold check box to the right of the Group Checks column.

12. Save your dashboard group.

### See Also:

- Defining Profile Variables
- Defining Profile Checks
- Defining Profile Groups
- Defining Categories

## 2.1.12 Determining Access by Application Module and Responsibility

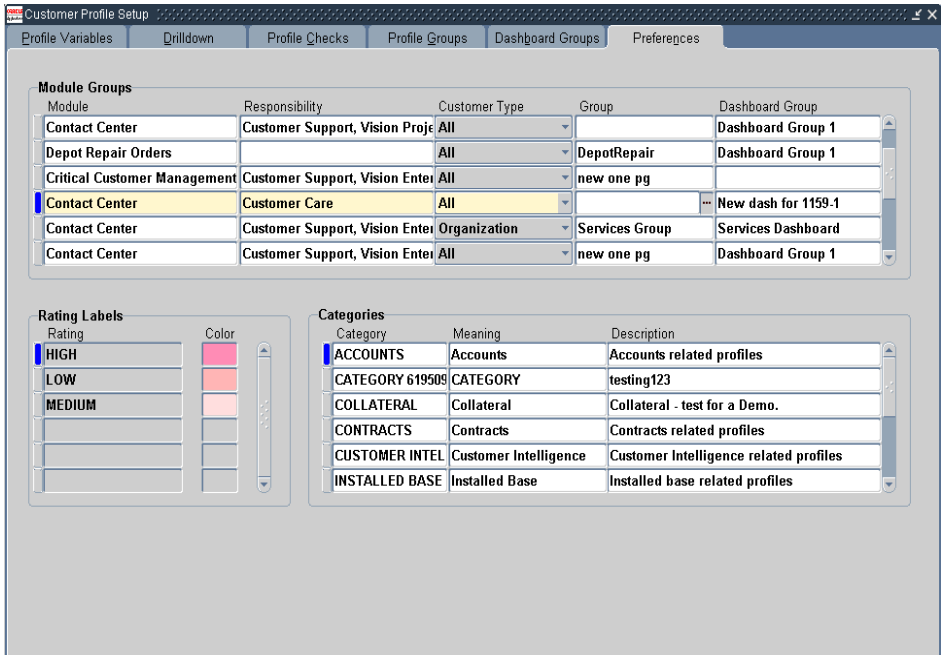
Use this procedure to determine which application module, customer type, and responsibility displays the profile groups and dashboard groups you have created.

### Prerequisites

You must create the profile and dashboard groups first.

### To determine access by modules, customer types, and responsibilities:

1. Navigate to Setup > Customer Management > Customer Profiles.
2. Select the Preferences tab.




3. Select an application module from the Module list of values. Although all application modules from all E-Business Suite applications appear in the list of values, you are restricted to enabling access only to those applications that use the Profile window and Dashboard tabs:

   For Dashboard Groups which are available in Oracle TeleService and Oracle TeleSales, you must select either:

   - Customer Dashboard

     The Contact Center window in Oracle TeleService

   - E-Business Center

     The E-Business Center window in Oracle TeleSales.

4. For Customer Profile groups, which are available in Oracle TeleService and Oracle Depot Repair applications, you must select one of the following:
   - Critical Customer Management

     The Override Customer Criticality window in Oracle TeleService.
   - Enter Service Request

     The Service Request window of Oracle TeleService.
   - Depot Repair Orders

     The Repair Order window in Oracle Depot Repair.
5. If you wish to restrict access to the customer information to agents with a particular responsibility, then select the responsibility using the Responsibility list of values (LOV).

   Leaving this value blank enables agents of all responsibilities to access to customer information.

   You can map more than one responsibility by creating multiple records.
6. Use the Customer Type drop-down list to restrict the type of profile and dashboard groups that appear in the Group and Dashboard Group LOVs:
   - Selecting Organization or Person restricts the selection to the profile and dashboard groups you have set up for that customer type.
   - Selecting All displays all dashboard and profile groups.
7. Select a profile group, a dashboard group, or both you wish to enable using the Group (Profile Group) and Dashboard Group LOVs.
8. Save your work.

## 2.1.13 Running the Customer Profile Engine

The Customer Profile Engine is a server side PL/SQL concurrent program. You must run the Customer Profile Engine after completing the customer profile setup process so that all the SQL statements defined during the setup are executed to generate the appropriate profile values. The Customer Profile Engine, when executed, performs the following operations in the sequence below:

- Evaluates the results of all the effective profile variables.
- Evaluates the results of all effective profile checks based on the profile variables.

- Evaluates the results for all the customers and accounts.

The Customer Profile Engine must be run to:

- Retrieve profile values for new customers
- Retrieve the latest profile values for all customers
- Reflect changes made to the profile setup
- Retrieve values for new profiles

The Customer Profile Engine can be run in two ways:

- As a concurrent program (for more information on running concurrent programs, refer to the *Oracle System Administator's Guide*).
- When agents click the Refresh button available in the Profile section or the Dashboard tab of the Contact Center or the E-Business Center in Oracle TeleSales. This manual method fetches the latest data for the displayed customer.

The parameters for running Customer Profile Engine are:

- Party Name
- Group Name

## 2.1.14 Setting Automatic Refresh of Profiles on the Dashboard

You can have the application automatically recalculate the customer profile values for a customer whenever the agent displays a customer record by setting the profile option Customer Care: To Allow Auto Refresh Facility for Dashboard to Yes. If the profile option is set to No, the profile values are updated only when the agent clicks the Refresh button. Use this procedure to set the profile option.

### To enable or disable the automatic refresh of customer profiles:

1. Navigate to Select > Others> System Profile Values
2. Using the Query by Example method, display Customer Care: To Allow Auto Refresh Facility for Dashboard.
3. Set the profile to Yes to enable the functionality. Setting the value No disables the automatic refresh.

# 2.2 Setting Up Relationship Plans

Relationship plans make it possible for your organization to deliver personalized and proactive customer service by alerting agents to customer needs. Whenever an agent is working on a customer record that meets the conditions you specify, the application can automatically display an alert message or launch a script that guides the agent through the interaction with the customer.

Relationship plans can trigger the alerts and scripts for agents working in the Contact Center and Service Request windows in Oracle TeleService and in the E-Business Center window for Oracle TeleSales.

## 2.2.1 Topic Organization

Here is how this topic is organized:

- The About Relationship Plans section on page 2-31 gives you a general overview of the relationship plan concepts.
- The Relationship Plans Components section on page 2-31 describes some of the modules you must set up to create relationship plans.
- The Process Overview on page 2-33 gives you an overview of the major setup steps in the order you must carry them out. You can access the detailed instructions from each step.

The other topics give you the details for individual implementation steps. They include:

- Defining Plan Groups on page 2-35
- Defining an Action on page 2-36
- Setting Up Messages and Scripts for Association with Relationship Plans on page 2-39
- Defining Messages on page 2-41
- Generating Messages on page 2-42
- Creating the Relationship Plan Itself on page 2-43
- Creating a Condition for the Relationship Plan on page 2-46
- Adding or Modifying Relationship Plan Condition Lines on page 2-50
- Enabling Relationship Plans on page 2-51
- Running the Relationship Plan Assignment Engine on page 2-53

## 2.2.2 About Relationship Plans

A relationship plan is a process which runs in the background periodically checking which customers meet one or more conditions you specify. Here two examples:

- Plan A

  If a customer contract is to expire in five days, then alert the agent with a message.

- Plan B

  If a customer has ten or more open high-priority service requests, then check if the customer has orders of $1 million or more.

  For customers with orders above $1 million, launch a script that guides the agent in handling the customer call.

  For customers below that threshold, alert the agent with a message.

Determining what action to take is a two-step process. The relationship plan first evaluates if a customer qualifies for the plan and then determines what action to take.

If your plan includes only one condition, then all customers that meet that condition qualify for the plan and the plan takes the action you specify. For Plan A, that means alerting agents for all the qualifying customers.

But a plan can take different actions for different subsets of the customers who qualify. Plan B first decides which customers qualify for the plan by checking which have ten or more high-priority open service requests.

The plan evaluates the qualifying customers using two more conditions and takes different actions for those customers who meet them:

- If a customer has orders of less than $1 million, the plan displays an alert.
- If a customer has orders of more than $1 million, the plan displays a script.

## 2.2.3 Relationship Plan Components

Relationship plan implementation involves many steps in multiple modules. You must create a mechanism to decide which customers belong to your plan, for example, and you must also create the actions, the messages, and the scripts, you wish the plan to display. The relationship plan itself (Setup > Relationship Plans > Define Relationship Plans) brings all the pieces together.

This section highlights the different types of setups you need to complete. A later section gives a process overview of the actual setup steps themselves.

### Monitoring the Database of Customers

If you wish to assign your customers to relationship plans automatically, you must set up a way to monitor the database to see if any customers qualify. For Plan A this means finding customers with expiring contracts. For Plan B, finding customers with high-priority service requests.

This involves setting up a Profile Check which retrieves and summarizes customer data based on the SQL query you enter. The Customer Profile Engine concurrent program refreshes this data at the periods you specify. See Setting up Customer Profiles on page 2-2.

### Evaluating if Customers Qualify for the Plan

Now that you are collecting data on your customers, you must create the condition which evaluates the data tracked by the profile check and determines whether a specific customer qualifies for the relationship plan. Does the customer have more than one open high-priority service request? Does the customer have more than five expiring service contracts?

You enter the qualifying condition in the Relationship Plan itself. It is evaluated by the concurrent program Relationship Plan Assignment Engine. If any customers meet the condition, then they qualify for the plan.

### Setting Up the Actions the Plan will Take

A relationship plan can take different actions on subsets of the qualifying customers. Now that you have assigned customers to the plan, you are ready to determine what actions you wish the plan to take for which customers.

You must define the actions and conditions separately.

For Plan B, this means creating two conditions:

- Condition B1: If a customer has orders of less than $1 million
- Condition B2: If a customer has orders of more than $1 million

And two actions:

- Action B1: Display alert.
- Action B2: Display Script

Plan A does not need to evaluate any additional condition because it displays an alert for all customers who qualify, however, you must still create one because conditions are the only way to trigger an action:

- Condition A1: <header information only>

You must also create the action for Plan A:

- Action A1: Display alert message that contract is about to expire.

Please note that the action setup uses a user an interface provided by Oracle Contracts which includes additional functionality you do not need to use for relationship plans.

### Setting Up the Messages or Scripts

You must also set up the alert messages and scripts you wish to display for the different actions your plan will take.

You set up the Messages in the Messages window under the Application Developer responsibility and compile them by running the Generate Messages concurrent program. Messages are similar to the standard application messages used by all Oracle applications and are fully described in the *Oracle Applications Developer's Guide*.

You must create your scripts using Oracle Scripting as described in that applications documentation.

### Hooking Up the Conditions to the Actions

When you create the conditions of your relationship plan, you cannot directly specify the alert messages and scripts the conditions are to launch. You must make this link indirectly by first defining an "Outcome" for each script and message. To create an Outcome, you define the script or message as a Process Definition of the type Outcome and then use the name of this process definition to create the link with the condition that will trigger it.

## 2.2.4 Process Overview

Here is an outline of the steps required to implement Relationship plans:

1. Log into Oracle applications under the Customer Support responsibility.
2. Set up a profile check that retrieve and summarize your customer data. You can use one of the profile checks you set up to display key customer information for agents as part of Customer Profile setup. If you do need to create a profile check

specifically for relationship plans, then you must complete at least the following steps:

a. Navigate to Setup > Customer Management > Customer Profiles.

b. On the Profile Variables tab, define the variable you wish to use in your profile check. This is where you enter the SQL queries. You can use the same variable in multiple profile checks. See Defining Profile Check Variables on page 2-11.

c. On the Profile Checks tab, define the profile check itself. See Defining Profile Checks on page 2-18.

3. Navigate to Setup > Relationship Plans > Events > Define Action and set up one action for every condition in the relationship plan. Every relationship plan must have at least one action. On the Basic tab, you must define all of the variables you want to use for your condition and every token that you wish to use in an alert message. See Defining an Actions on page 2-36 for details.

4. If you wish to have your relationship plan display an alert message, then you must:

   a. Under the Application Developer responsibility, navigate to Application> Messages.

   b. Define all of your messages as described in Defining Messages on page 2-41. For displaying customer information, such as the customer name, in the text of the message you are limited to using only the tokens you have defined in the Action window in the previous step.

   c. Generate the messages (Other > Requests > Run) by running the Generate Messages concurrent program. See Generating Messages on page 2-42.

5. If you wish to have your relationship plan display a script, then prepare the script as described in Oracle Scripting documentation and note down its name. You can obtain this from the Name field of the Script Properties window in the script author's file.

6. You have now created the action(s) your relationship plan will take, but you cannot hook them up to the relationship plan just yet. You must first create an intermediary called a Process Definition of type Outcome. It is this process definition or "Outcome" and not the action itself that you associate with the relationship plan conditions that launch them.

   To create the outcome:

a. Switch to the Customer Support responsibility and navigate to Setup > Relationship Plans > Events > Define Process Definition.

b. Create the outcomes according to the steps outlined in Setting Up Messages and Scripts for Association with Relationship Plans on page 2-39.

7. If you wish to organize your relationship plans in categories, then navigate to Setup > Relationship Plans > Define Plan Groups and define these categories as described in Defining Plan Groups on page 2-35.

8. You are now ready to create the relationship plan itself which ties together all of the setups you have done so far. In this step you enter both the condition that determines which customers qualify for the plan and any other conditions which decide what action the plan will take. The setup is done in the New Plan window (Setup > Relationship Plans > Define Relationship Plans). See Creating the Relationship Plan Itself on page 2-43 and Creating a Condition for the Relationship Plan on page 2-46 for details.

9. Now you must enable your plan for any of the forms where you wish the actions to appear. Navigate to Setup > Relationship Plans > Enable Relationship Plans) and enable the forms as described in Enabling Relationship Plans on page 2-51. The options are: Contact Center, Service Request, and the E-Business Center from Oracle TeleSales. See.

10. As a last step, run the concurrent program Relationship Plan Assignment Engine to assign customers to your plan. See Running the Relationship Plan Assignment Engine on page 2-53.

## 2.2.5 Defining Plan Groups

Use this procedure if you wish to create categories to organize your relationship plans into logical groups. Defining plan groups is optional.

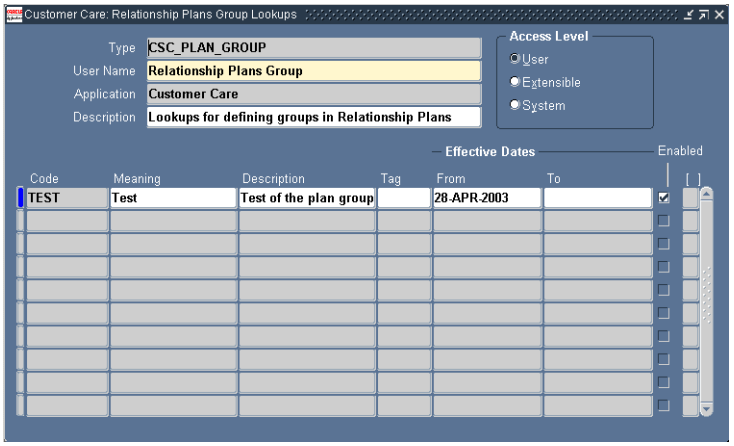
### Prerequisites

None

### To define relationship plan groups:

1. Navigate to Setup > Relationship Plans > Define Plan Groups.

   The Relationship Plans Group Lookup window appears.




2. In a new row, enter a value in the Code field.
3. Enter a value in the Meaning field.
4. Enter a value in the Description.
5. The Tag field is not used in Oracle TeleService .
6. Optionally enter dates in the From and To fields.
7. Save your plan group.

**See Also:**

Defining Relationship Plans

## 2.2.6 Defining an Action

Use this procedure to define the action the application will take when the relationship plan applies to a particular customer.

If you wish to display an alert message for agents as your action, then you must define any tokens you wish to use in the message text in the Element Name field of the Basic tab. For example, if you wish to include the customer's name in the alert message, then you must enter `CUST_PARTY_NAME`.

**Note:** You must enter the same token name when you set up condition that will trigger this action. See Creating a Condition for the Relationship Plan on page 2-46.

If the action you are defining is triggered by evaluating a relationship plan condition, then you must also define the variable you will use for the condition. If your relationship plan contains a condition that will take this action based on the city where the customer is located, then in the Basic tab you must enter the variable `CITY`.

If you wish to view the available list of variables/tokens to choose from, you can query up the action with the name `Customer Care - Generic Action` or refer to Appendix C, "Seed Data for Relationship Plans" of this guide.

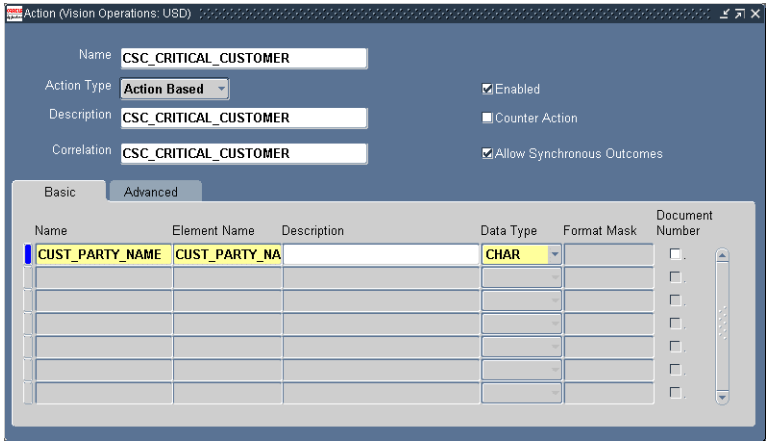
### Prerequisites

None

### To define an action:

1. Navigate to Setup > Relationship Plans > Events > Define Action.

   The Action window appears.




2. Enter a name for your action in the Name field.

3. Select **Action Based** from the Action Type drop-down list. Date Based is not supported.
4. Optionally enter a description of your action in the Description field.
5. Enter a unique name in the Correlation field. You can start the name with the prefix CSC_, for example, to indicate your action is related to Oracle Customer Care.
6. Select the Enabled check box to make this action available for use.
7. Select the Allow Synchronous Outcomes check box to allow a real-time alert or script to pop up.
8. If your action displays an alert message, then in the Basic tab, enter any tokens you wish to use in your message:

   **Note:** You need not define any tokens for generic messages that apply to all customers.

   a. In Element Name, enter a valid variable name.

      You can obtain the list of valid Contact Center header variable names by querying up the `Customer Care - Generic Action` or from Appendix B of this guide.

      For example, if you wish to display the full name of the customer in the alert message, then enter `CUST_PARTY_NAME`.

      You must also enter the same value in two other places: in the text of the message when you define the message in the Messages window and in the Parameters window when defining the condition that uses this action.

   b. You can enter the same value in the Name field as in the Element Name field.

   c. Optionally, enter a description in the Description field.

   d. Select a data type of CHAR, NUMBER, or DATE in the Data Type field.

   e. Select a format mask, if appropriate.
9. If your action is triggered by a condition that evaluates a variable, then define the variable in the Basic tab in the same manner as the message tokens above.
10. Save your action.

### See Also:

- Defining Process Definition
- Defining Relationship Plans
- Adding or Modifying Relationship Plan Condition Lines
- Enabling Plans

## 2.2.7 Setting Up Messages and Scripts for Association with Relationship Plans

You can indicate which script or message your relationship plan is suppose to launch only indirectly by first creating what is called a Process Definition of type Outcome. It is the name of this process definition, or outcome, rather than the name of the message or script itself, that you will associate with the condition of your relationship plan.

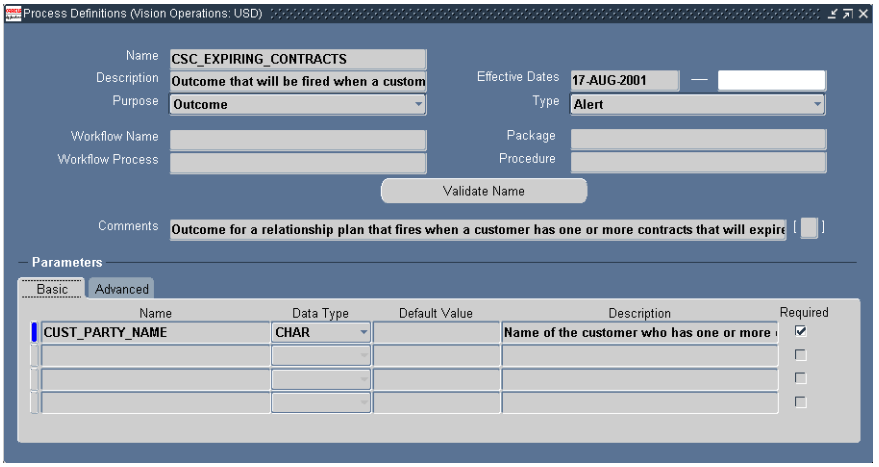
### Prerequisites

- You must create the action, message, and script first.
- For Scripts, you must know the name of the script. You can obtain this from the Name field of the Script Properties window. You can open this window from the script author's file.
- For messages, you must know:
  - The value you entered in the Name field of the Messages window.
  - The value(s) you entered for the individual message tokens used by the message. (These were entered on the Basic tab of the Action window in the Element Name fields.)

### To define process definitions of type outcome:

1. Navigate to Setup > Relationship Plans > Events > Define Process Definitions.

   The Process Definitions window opens.




2. In the Name field, enter the name of the script or message.
3. Optionally enter a description in the Description field.
4. Select **Outcome** from the Purpose drop-down list. This is the only permitted value.
5. From the Type drop-down box, select either Alert or Script. The other types are not supported.
6. Optionally enter comments in the Comments field.
7. If you are creating the outcome for a message, then in the Name fields of the Basic tab of the Parameters region, enter the same values you entered for Element Name fields on the Basic tab of the Action window.
8. The Data Type field defaults to the data type of the Element Name.
9. Save your process definition.

### See Also:

- Defining an Action
- Defining Relationship Plans
- Adding or Modifying Relationship Plan Condition Lines
- Enabling Plans

## 2.2.8 Defining Messages

Use this procedure to define the alert messages the agents will see. Messages are defined for use in displaying messages about customers in real-time alerts.

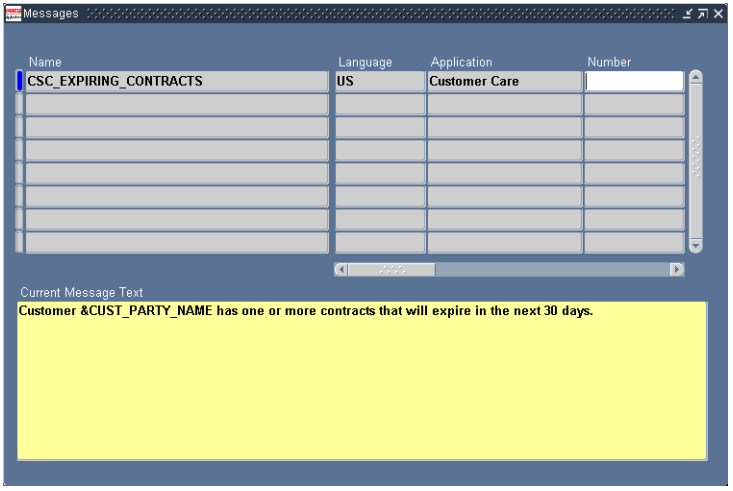
### Prerequisites

None

### To define messages:

1. Under the Application Developer responsibility, navigate to Application > Messages.

   The Messages window opens.




2. Enter a name for your message in the Name field. You must enter this same name when you create a Process Definition.
3. Select a language from the Language list of values (LOV).
4. Select `Customer Care` from the Application LOV.

   Entry in the remaining fields, Number, Type, Maximum Length and Description, are optional.

5. In the Current Message Text field, enter the message that you wish to display as a real-time alert.

   If you wish to display the value of a token, for example, the customer name `CUST_PARTY_NAME`, then you must enter it preceded with an ampersand symbol, for example, `&CUST_PARTY_NAME`.

   You are restricted to using only the tokens that you have entered in the Element Name fields in the Action window. See Defining an Actions on page 2-36.

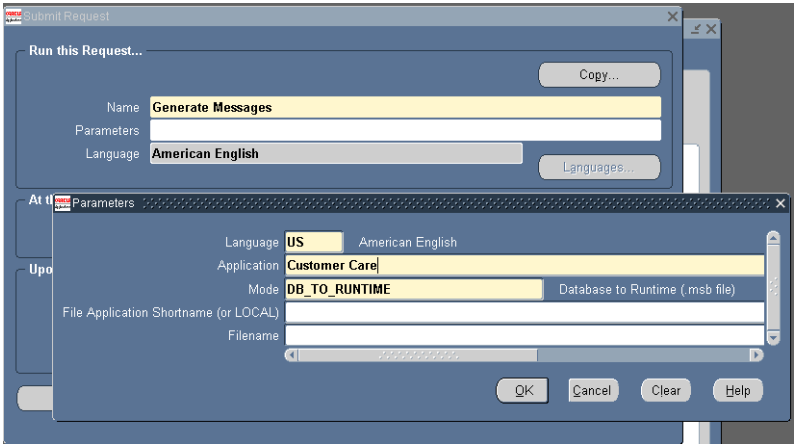
6. Save your message.

## 2.2.9 Generating Messages

Use this procedure to generate the messages you have created.

### To generate messages:

1. Under the Application Developer responsibility, navigate to Other > Requests > Run.
2. Select the Single Request radio button in the Submit a New Request window and click OK.
3. Select Generate Messages from the Name list of values.
4. In the Parameters window enter:
   - Language: `US` or another message language
   - Application: `Customer Care`
   - Mode: `DB_TO_RUNTIME`




5. Click **OK** to close the Parameters window.
6. Click **Submit**.
7. Click **No** to submit another request.
8. You can view the status of your requests by navigating to View > Requests >All My Requests > Find

## 2.2.10 Creating the Relationship Plan Itself

Use this procedure to create the plan which ties all of the pieces of the relationship plan together. It is here you enter the condition that determines whether customers qualify for the plan and also the conditions that decide which action the plan will take.

Oracle TeleService provides a predefined relationship plan, Expiring Contracts, which displays an alert when a customer has any contracts that will expire in the next 30 calendar days. This relationship plan uses the profile check called Expiring Contracts. Refer to Appendix B for more details about the predefined relationship plan.

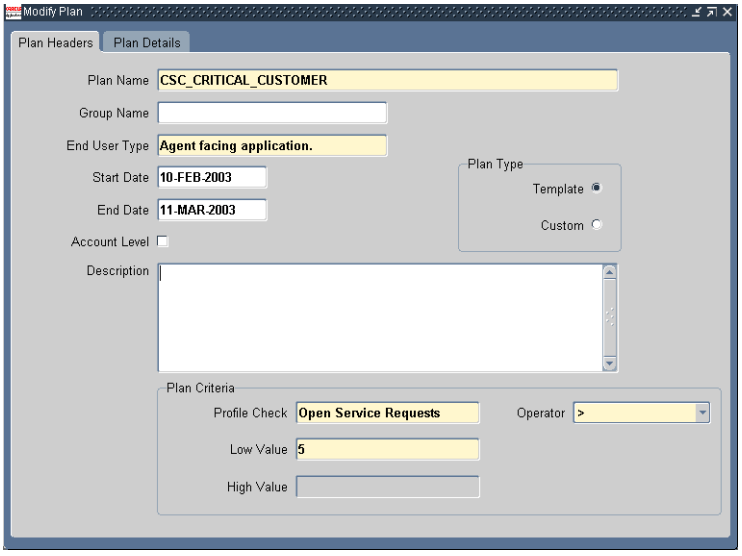
### Prerequisites

You must have defined your actions, process definitions, messages, and scripts.

### To define the plan itself:

1. Navigate to Setup > Relationship Plans > Define Relationship Plans.

   New Plan window appears.




2. In the Plan Headers tab, enter a name for the plan in the Name field.
3. If you wish to group this relationship using relationship plan categories you have created, then select a group name from the Group Name list of values.
4. Optionally enter start and end dates for this relationship plan.
5. Unless you wish to create a plan for individual accounts rather than for customers, leave the Account Level check box unselected. An account relationship plan tracks customers who generate a significant amount of revenue in one account, for example.
6. All relationship plans are initially created from a template, so the Template radio button is selected by default in the Plan Type section. The Custom radio button can be used only from the Inquiry tab when you customize a relationship plan for one specific customer.
7. Enter a description of the Relationship Plan in the Description field.

**a.** In the Plan Criteria section, you define the criteria to be used by the Relationship Plan Assignment Engine to decide which customers qualify for the plan:

**b.** Using the Profile Check list of values, select the profile check that summarizes the customer information you wish to evaluate. The profile checks contain those seeded with the application as well as those you have defined.

**c.** Select an operator from the Operator list of values. The choices are =, <, >, >=. <=, Between, and Not Between.

**d.** Enter the constant in the Low Value field.

**e.** The high value is available only if you previously selected an operator value of Between or Not Between.

**8.** Save your relationship plan.

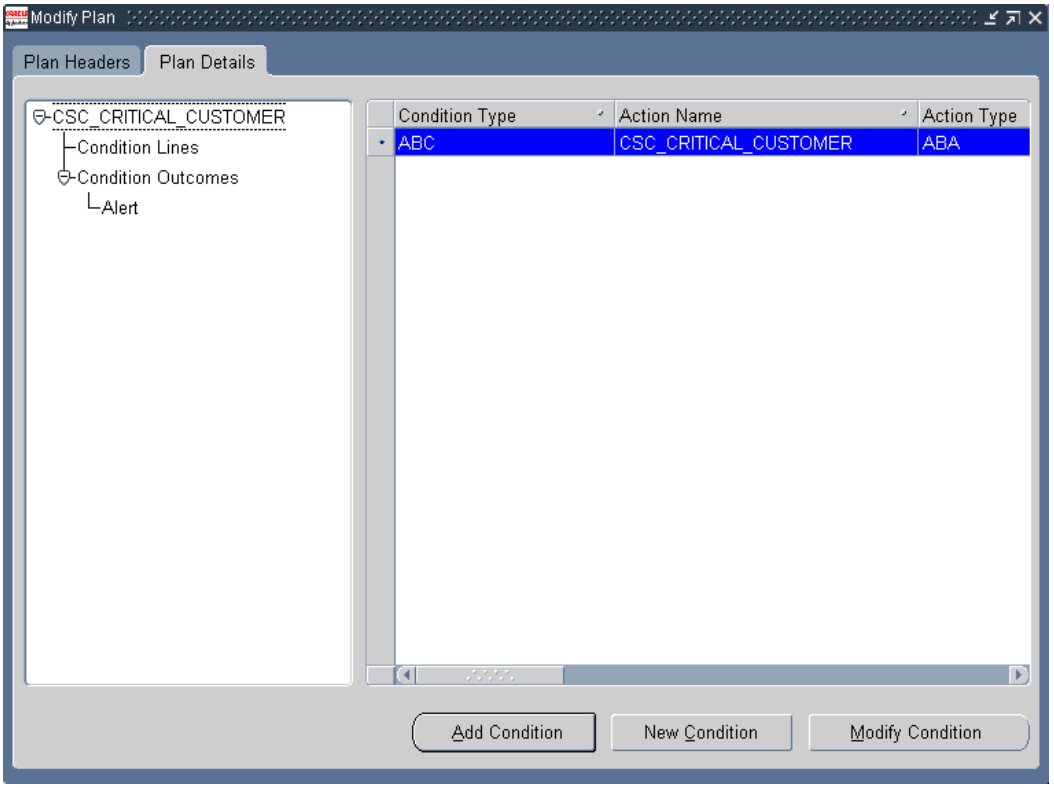
You are now ready to specify the action(s) the relationship plan is to take and any additional conditions will trigger them.

---

**Note:** If you wish the relationship plan to display an alert message or script to agents for all customers who qualify, you must still create a new condition for each action you wish the plan to take. Conditions provide the only way to specify which actions the plan will take. If you wish all qualifying customers to trigger the action, you simply leave the expression tab blank.

---

**9.** Select the Plan Details tab.




10. Click **New Condition** to create a new condition that will trigger one of the action as described in Creating a Condition for the Relationship Plan on page 2-46.

## 2.2.11 Creating a Condition for the Relationship Plan

Use this procedure to create a condition for your relationship plan. Each relationship plan must have at least one condition attached to it.

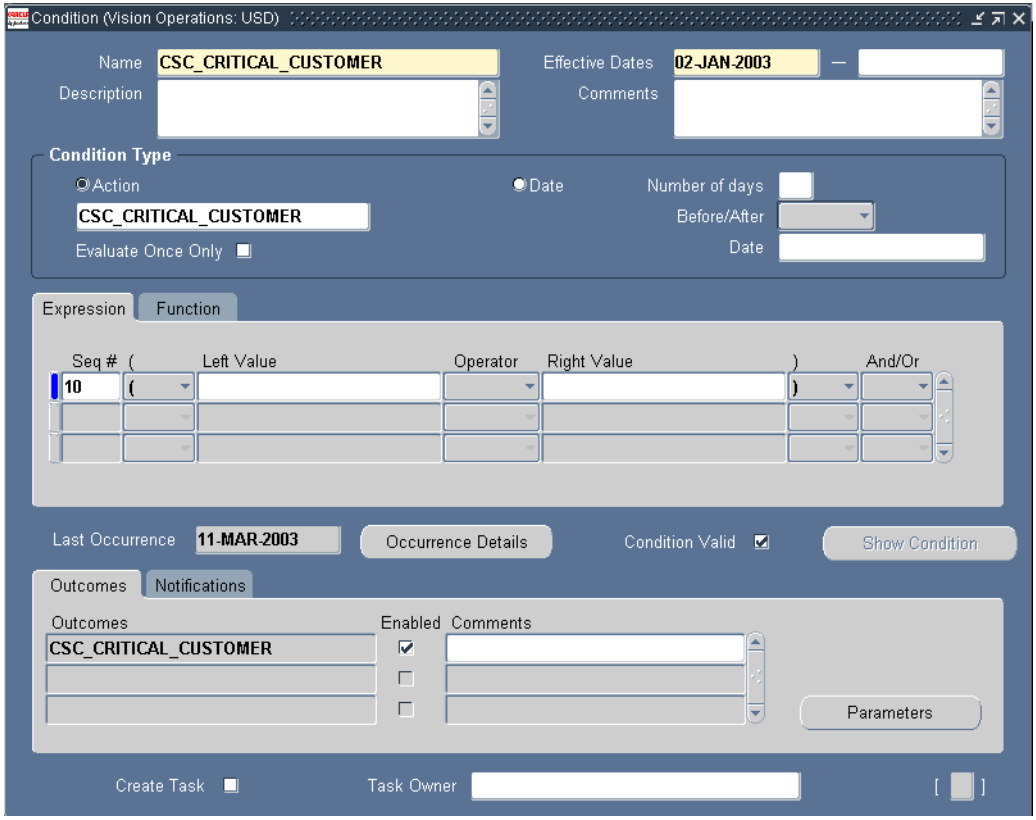
### Prerequisites

You must create the relationship plan itself before adding conditions.

### To create a condition for a relationship plan:

1. Navigate to Setup > Relationship Plans > Define Relationship Plans.
2. Display the relationship plan where you wish to create a condition.
3. Select the Plan Details tab.
4. Click **New Condition**

   The Condition window appears.




5. Enter a name for the condition in the Name field.
6. Optionally enter effective dates, description and comments for your condition.

7. In the Condition Type region, select the Action radio button. The Date condition type is not supported in relationship plans.
8. Select an action from the Action list of values. The list of values contains actions that you previously defined. You must use a unique action for every relationship plan.
9. Leave the Evaluate Only Once check box unselected unless you wish to evaluate your condition only once.
10. In the Outcomes region, use the Outcomes LOV to select the outcome which represents the message or script you wish to display for customers matching this condition.

    A relationship plan can display a message, a script, or both.

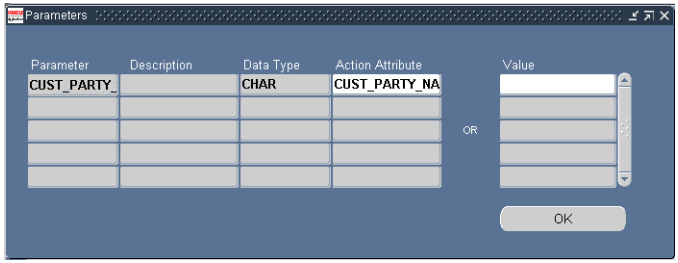
    Outcomes, are the intermediaries which permit you to associate scripts and messages with relationship plans. They are defined in the Process Definition window as described in Registering Messages and Scripts for Use in Relationship Plans on page 2-39.
11. If the outcome is a message which displays the value of one or more tokens, such as the customer's name, then you must re-enter the same message tokens:
    a. Click **Parameters**.
    b. The Parameters window appears.

---

**Note:** If you have defined more than one outcome for this condition, then the application displays a different window. In this case you must first:

- Select the parameter you wish to define.
- Right-click and select **Populate selected rows in PARAMETERS** from the pop-up menu.

---




c. Select the token from the Action Attribute list of values.

For example, if you are displaying the customer name in the body of the message then you must select `CUST_PARTY_NAME` in the Action Attribute.

---

**Note:** Most of the time you will want the value in the Parameter and Action Attributes fields will be the same. But you are permitted to assign different values to the token in your message provided they are of the same data type. For example, if you wish to display the customer name in the message, then you can map the full customer name, the first name, or the last name to that token using this window.

---

d. Click **OK** to return to the Conditions window.

12. If you wish to enter a condition which will trigger the action, then enter it in the Expression tab. You can use the parameters you have just mapped.

13. Save the condition.

## See Also:

- Defining Plan Groups
- Defining an Action
- Defining Process Definitions for Outcomes
- Adding or Modifying Relationship Plan Condition Lines

- Enabling Plans

## 2.2.12 Adding or Modifying Relationship Plan Conditions

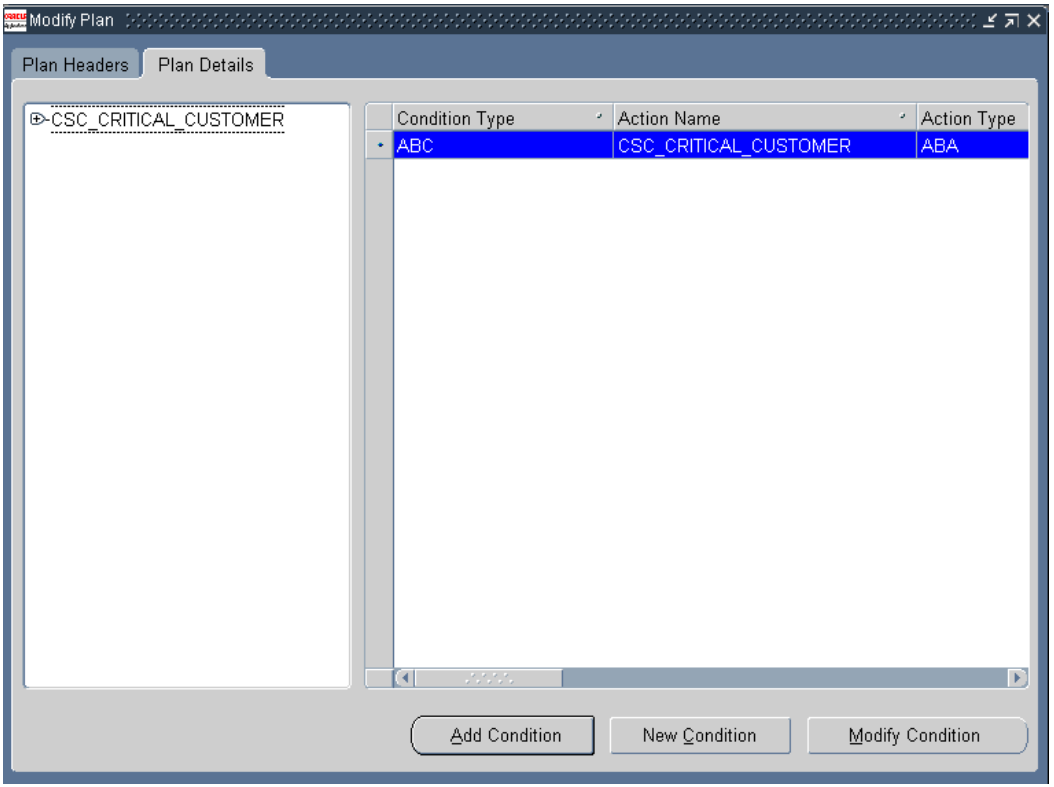
Use this procedure to modify relationship plan conditions. You can add additional conditions to an existing plan or modify a condition already attached to your plan.

### Prerequisites

You must define a relationship plan first.

### To add or modify relationship plan conditions:

1. Navigate to Setup > Relationship Plans > Define Relationship Plans.




2. Display the Relationship Plan to which you want to add condition lines.

3. Select the Plan Details tab in the Modify Plan window.
4. Click **Add Condition**.
5. Select a condition and click **OK** to attach it to the plan header.
6. If you wish to modify a condition line of a relationship plan, then:
   a. Click **Modify Condition** in the Plan Details window.
   b. Make the necessary changes to the condition line or outcome.
   c. Save the modified condition line.

### See Also:

- Defining Plan Groups
- Defining an Action
- Defining Process Definitions for Outcomes
- Defining Relationship Plans
- Enabling Relationship Plans

## 2.2.13 Enabling Relationship Plans

You can enable your relationship plan in the Contact Center and Service Request windows of Oracle TeleService and in the E-Business Center of Oracle TeleSales.

Use this procedure to enable the relationship plan in one or more of these three windows.

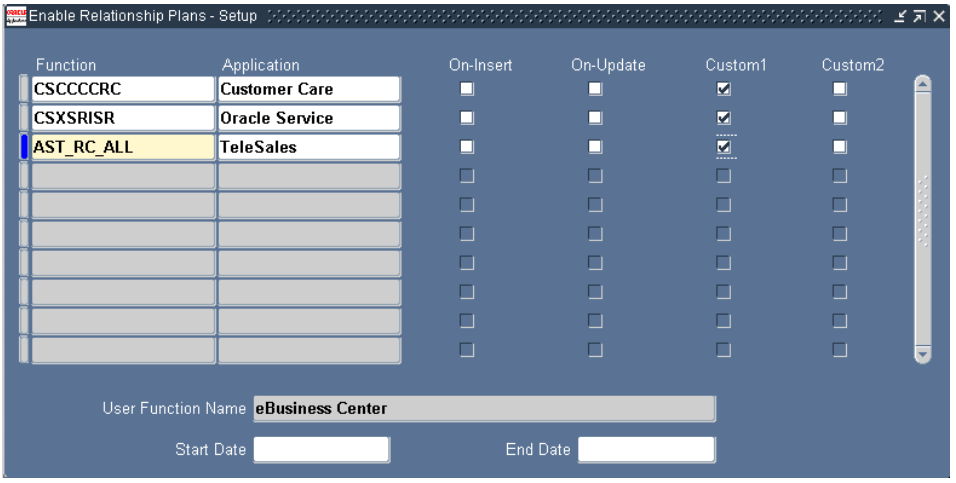
### Prerequisites

None

### To enable relationship plans:

1. Navigate to Setup > Relationship Plans > Enable Relationship Plans.

   The Enable Relationship Plans - Setup window appears.




2. The Function list of values displays all forms, but you must select one of the three following forms:
   - CSCCCCRC for Contact Center
   - CSXSRIXSR for Service Request
   - AST_RC_ALL for E-Business Center
3. Select the Custom1 check box for each of the forms where you wish the relationship plan to be enabled.
4. Save your settings.

---

**Note:** Please see the appendix, Enabling Relationship Plans, for additional information.

---

### See Also:

- Defining Plan Groups
- Defining an Action
- Defining Process Definitions for Outcomes
- Defining Relationship Plans
- Adding or Modifying Relationship Plan Condition Lines

## 2.2.14 Running the Relationship Plan Assignment Engine

The Relationship Plan Assignment Engine is a server side PL/SQL concurrent program. You must run the Relationship Plan Assignment Engine after completing the relationship plan setup process so that all associations between customers and relationship plans are made.

This concurrent program also runs automatically each time you run the Customer Profile Engine.

The Relationship Plan Assignment Engine performs the following operations:

- Evaluates all customers that meet relationship plan criteria and associates the customers with the appropriate relationship plan.
- Evaluates all customer that do not meet relationship plan criteria and disassociates the customer from the relationship plan.
- Checks for all customers who have been manually assigned to relationship plans and ignores automatic association/disassociation rules for these customers.

The following parameters are available for running the Relationship Plan Assignment Engine:

- Plan Name
- Check Name
- Party Name
- Account Name

Each parameter provides a list of values from which to choose. If a value is entered in the Party Name parameter, the Account Name list of values contains only those values for the selected Party Name. If no Party Name is selected, the Account Name list of values displays all accounts.

You can initiate the Relationship Plan Assignment Engine in two ways. First, you can run it as a concurrent program. Secondly, the program is run automatically whenever the customer profiles are updated by the Customer Profile Engine. Customer profiles can be updated by running the Customer Profile Engine as a concurrent program or by selecting the Refresh on the Dashboard tab.

# 2.3 Setting Up Quick Menu

Quick menu is based on seeded filters. Filters have a many to many relationship with AOL functions and are seeded in the Quick Menu tables. This section defines the association of the filter with its function in AOL.

For the Customer Support responsibility, the Quick Menu has already been setup. To familiarize yourself with the quick menu setup, perform the following steps:

- Viewing System Profile Values
- Verifying AOL Menu for Quick Menu
- Configure the Quick Menu to meet your needs

## 2.3.1 Viewing System Profile Values

**To view the profile values:**

1. Open the System Profile Values window from the navigator.
2. Query on the profile name called Start Menu in Quickmenu.
3. The default value is Quick Menu for Customer Support.

## 2.3.2 Verifying AOL Menu for Quick Menu

**To verify an AOL menu for quick menu:**

1. Using the Application Developer responsibility, navigate to Application > Menu.

   The Menus window appears.
2. Run a query to open the seeded menu CSX_CUSTOMER_SUPPORT_QM.
3. View the results of the query.
4. Close the window.

## 2.3.3 Configuring the Quick Menu

You can configure a different quick menu for each responsibility by adding or deleting Quick-Menu-enabled forms.

The seeded quick menu displays all quick-menu enabled forms:

- Manage Critical Customer
- Maintain Relationship Plans
- Order Capture (Order Quoting - Forms
- View Customer Product
- Maintain Systems
- Contact Center
- Service Request
- Depot Repair

The following item is not quick-menu enabled as it uses the HTML interface:

- Search Knowledge Base

You can add any additional form function to the Quick Menu, even though the form function is not quick menu enabled. This allows you to quickly navigate to the form, but the variables in the original form are not passed to the form on the quick menu. Only quick menu enabled forms can accept variables from the original form.

## 2.4 Defining Profile Options

System profile options help you control how Customer Care controls access to and processes data. During implementation, you set a value for each profile option at user, responsibility, application and site levels.

Site-level profile settings apply to all users at an installation site. Application-level profile settings apply to all the users of the specified application. Profile settings at application level override those set at site level. Responsibility-level profile settings apply to all users who use the responsibility to sign on to the application. Responsibility-level settings override those at site and application levels. User-level profile options apply to individual users identified by their application user names. User level profile options override all other profile options.

The following Customer Care system profile options must be set up unless otherwise noted. You must use the System Administrator responsibility to set up profile values. For a detailed explanation of system profile options and the procedures for setting them up, please refer to *Oracle Applications System Administrator's Guide*.

### Customer Care: Critical Customer Check

This profile option sets the Profile Check used to determine criticality of customers.

Default Value: Critical Customers

### Customer Care: Default Outcome for Interactions

See Linking Outcome, Result and Reason Codes for information about this profile option.

### Customer Care: Default Party Type

This profile option sets the default value in the Caller Type field in the Contact Center.

Default Value: Organization

### Customer Care: Default Reason for Interactions

See Linking Outcome, Result and Reason Codes for information about this profile option.

### Customer Care: Default Relation for Person

This profile option sets the default value in the Relation field of the Caller Information window.

Default Value: Employee of

### Customer Care: Default Result for Interactions

See Linking Outcome, Result and Reason Codes for information about this profile option.

### Customer Care: Default Tab for Contact Center

This profile option sets the default navigation tab that is open when the Contact Center form is opened.

Dealt Value: Dashboard

### Customer Care: Install Base Form Check

This profile option identifies which Oracle Install Base window opens when the Installed Base button is clicked in Contact Center. The choices are Customer Products or Comms Customer Products.

Default Value: Customer Products

### Customer Care: Log Task Activities Flag

This profile option sets the flag to control logging of Interaction Activities for Tasks.

Default Value: Yes

### Customer Care: Service Request Form Check

This profile option identifies which Service Request window opens when the Details button is clicked in the Service Request tabbed page of Contact Center. The choices are Enter Service Requests or Enter Comms Service Requests.

Default Value: Enter Service Requests

### Customer Care: Default Country for Contact Center

This profile option used for defaulting COUNTRY when entering addresses.

Default Value: Null

### Customer Care: Default Site Use for Contact Center

This profile option used for defaulting Site use for an address in Contact Center.

Default Value: Address

### Customer Care: To Query Dashboard by Party or Account

This profile option sets the query default level for the dashboard.

Valid Values: Party or Account.

Default Value: Party

### Customer Care: To Use Exact Match for ANI Search

This profile option sets to use the exact match for ANI search to find a customer conversions.

Default Value: Null

### ContactCenter: New CONtact Default Create Account

This profile option specifies whether the Create Account check box in the Caller Information window is selected by default.

Default Value: No

### ContactCenter: New Contact Default Address Type

This profile option specifies the address type that appears by default in the Address Type field of the Caller Information window. The values in the list of values are derived from address types that are defined in Oracle Receivables.

Default Value: Bill to

### ContactCenter: New Contact Default Phone Type

This profile option specifies the default value that appears in the (Phone) Type field in the Caller Information window. The values in the list of values are derived from phone types that are defined in Oracle Receivables.

Default Value: Work

### ContactCenter: Show Active Parties Only

This profile option sets to make sure that only show active parties.

Default Value: Active Part

### ContactCenter: Show Active Account Only

This profile option sets to make sure that only show active account.

Default Value: Account

### CSI: Display HTML UI

This profile option specifies whether you want to see the HTML user interface, or the Oracle Forms-based user interface, when you open Oracle Install Base from the Contact Center.

If you select No, then the Forms-based user interface is displayed, and the form displayed is the one specified in the profile option Customer Care: Install Base Form Check.

If you select Yes, the HTML user interface is displayed, and the page displayed is the one specified in the profile option Customer Care: Install Base Form Check.

Default Value: No

**Note:** The following two conditions must also be verified if you want the Installed Base HTML user interface to be displayed.

- Check to see if Installed Base Search Products exists in the CSX_CUSTOMER_SUPPORT menu.
- The user's login responsibility is Oracle Installed Base User.

### Service: Default new Note Type in Workbench Tab

This value sets the default value for the note type in a service request

Valid value: Any Note Type

Default value: None

### Task Manager: Default task Status

This value sets the default Task Status value when creating a task.

Valid values: Any Task Status

Default value: None

### Task Manager: Default task Type

This value sets the default Task Type value when creating a task.

Valid values: Any Task Type

Default value: None

### Task Manager: Default task Priority

This value sets the default Task Priority value when creating a task.

Valid values: Any Task Priority

Default value: None

### Task Manager: Default task Owner

This value sets the default Task Owner value when creating a task.

Valid values: Any valid resource.

Default value: None

## 2.5 Linking Outcome, Result, and Reason Codes for Interaction History

Every interaction outcome can have multiple results, and each result can have multiple reasons. By linking them, you can restrict the lists of values used by agents in the Call Wrap Up window. The Call Wrap Up window can be invoked from the Contact Center by clicking the Call Wrap Up icon on the toolbar.

### To link Outcome, Result, and Reason Codes:

1. Define the following profile options. These profile options are mandatory:

- Customer Care: Default Outcome for Interactions

  This profile option sets the default outcome for Interactions and Activities in the Contact Center.

  Default Value: Incoming

- Customer Care: Default Result for Interactions

  This profile option sets the default result for Interactions and Activities in the Contact Center.

  Default Value: Message not Sent

- Customer Care: Default Reason for Interactions

  This profile option sets the default reason for Interactions and Activities in the Contact Center.

  Default Value: Too Busy

2. Establish relationships between these profile option values in the Interaction History Administration window:

   a. Open the Interaction History Administration window using the following navigation path:

      Setup > Customer Management > Interaction History Administration

   b. Open the Outcome - Results Tab.

   c. Establish relationships between Outcome and Results from the list of values.

   d. Save your changes.

   e. Open the Result - Reasons tab

   f. Establish relationships between Results and Reasons from the list of values.

**g.** Save your changes.

**h.** Exit out of the window.

## 2.6 Setting Up Party and Account Numbering

Party numbers and account numbers can be generated manually or automatically.

### To setup party and account numbering:

1. Set up automatic account numbering:

   **a.** Select an Oracle Receivables responsibility and navigate to System > System Options.

   The System Options window appears.

   **b.** Open the Trans and Customers tab.

   **c.** Check the Automatic Customer Numbering check box if you want to generate customer account numbers automatically in Customer Care.

   **Note:** Although this number is referred to as Customer Number in Oracle Receivables, this number is referred to as Account Number in Oracle TeleService.

2. To enable automatic party number generation in Oracle TeleService, set the HZ: Generate Party Number system profile option to Yes.

## 2.7 Setting Up Global Address Formatting and Address Validation

Agents can enter customer addresses in the formats appropriate for the country where the customer is located. For example, if a customer has a branch in Germany, agents enter the address in a format recommended by the Bundespost. For addresses in the United Kingdom, they enter the address in the format recommended by the Royal Mail.

There are six preset address formats that are available for your use:

- Japan
- Northern Europe
- South America

- Southern Europe
- UK/Africa/Australia
- United States of America

The different address formats are implemented in the application by using descriptive flexfields. Agents enter the country using a list of values and then enter the rest of the address in a flexfield window. If there is no address style associated with the country, then the application uses the address style defaulted in the profile Customer Care: Default Address Style for Contact Center.

### Prerequisites

Understanding of descriptive flexfields

### To set up global address formatting and address validation:

1. Set up flexible addresses and address validation as described in the Flexible Address and the Address Validation sections of the Customers chapter of the *Oracle Receivables User Guide.*
2. Set up the two following profile options:
   - Customer Care: Default Country for Contact Center

     This profile option defaults the country in the Country field of an address on the Contact Center. The setting of this profile option is mandatory as there is no seeded default value.
   - Customer Care: Default Address Style for Contact Center

     If there is no address style associated with the country, then the application uses the address style defaulted in this profile. The setting of this profile option is mandatory as there is no seeded default value.
3. If you wish to create additional address styles then follow the instructions outlined in Setting Up Additional Address Styles on page 2-63.
4. Map the countries to address styles as described in Mapping Countries to Address Styles on page 2-65.
5. Set up the Customer Care: Default Address Style for Contact Center system profile option to the address style you wish to use in case a user fails to enter a country for an address.

## 2.7.1 Setting Up Additional Address Styles

Use this procedure to create additional address styles or to modify existing ones.

### Prerequisites

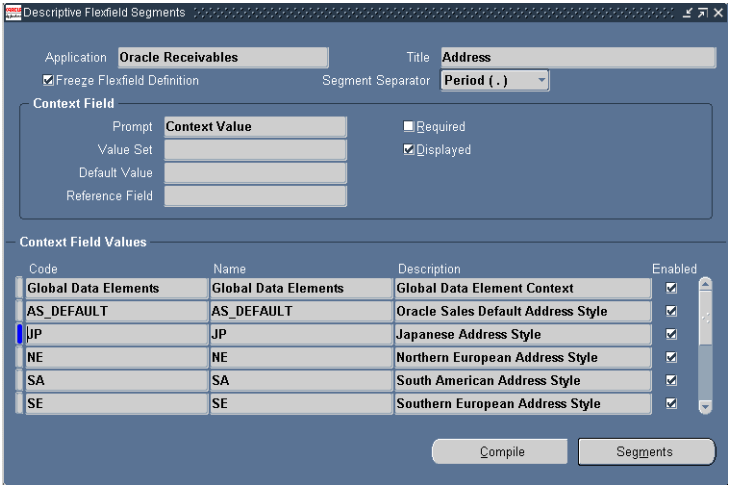
Familiarity with descriptive flexfields.

### To set up additional address styles:

1. Log into Oracle applications under the Application Developer responsibility.
2. Navigate to Flexfield > Descriptive > Segments.

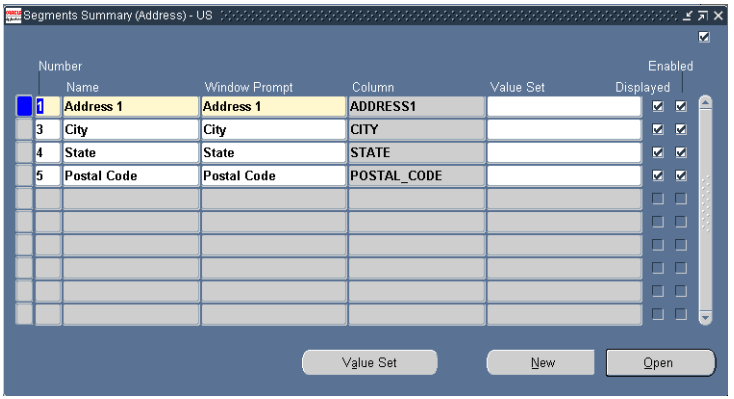
   The Descriptive Flexfield Segments window appears.
3. Search for `Address` in the Title field using the Query Enter / Query Run search method available from the View menu.

   The Descriptive Flexfield Segments window lists the available address styles.




4. With the Freeze Flexfield Definition check box unselected, create a new record in the Context Field Values region.
5. Enter the name of the style in the Code and Name fields.
6. Enter an optional description.
7. Click **Segments**.

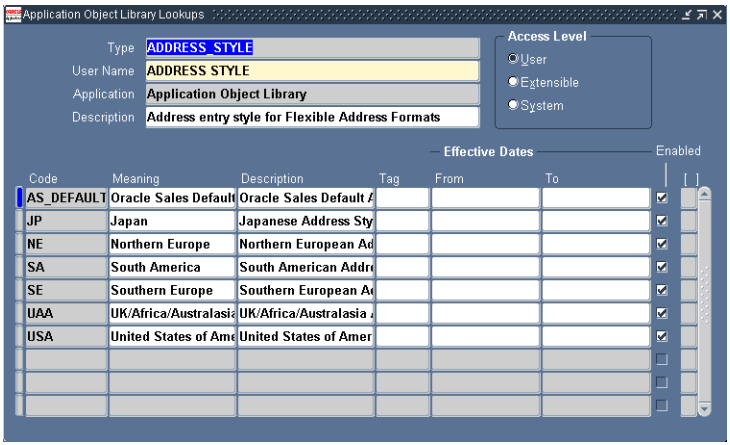
The Segments Summary window appears.




8. Enter each element of your address style:
   a. Enter a number that determines the order of the address element. The address elements appear in ascending order.
   b. Enter the element Name and Window Prompt.
   c. Select one of the available address components using the Column LOV.
   d. Select the Displayed and Enabled check boxes.
9. Click **Save**.
10. Navigate to Application > Lookups > Application Object Library.

    The Application Object Library Lookups window appears.
11. Search for ADDRESS_STYLE in the Type field using the Query Enter / Query Run search method available from the View menu.

    The Application Object Library Lookups window lists the available address styles.




12. Enter the code for your address style. The value you enter in the Code field must match the code you have entered in the Context Field Values.
13. Enter the wording the agents will see in the Meaning field.
14. Enter an optional description.
15. Make sure the Enabled check box is selected.
16. Click **Save**.

## 2.7.2 Mapping Countries to Address Styles

You must make sure that each country you are doing business with is mapped to the appropriate address style. Use the following procedure to map address styles.

### Steps

1. Under the Receivables Manager responsibility, navigate to Setup> System > Countries.

   The Countries and Territories window appears.
2. Search for the countries that you want to set up using the Query/Enter Query/Run method.
3. Use the Address Style List of Values (LOV) to select the address style for each country.
4. Click **Save** on the toolbar.

# 2.8 Setting Up Custom Tabs on the Contact Center

Implementers can create two custom tabs in the Contact Center window by modifying the code for the Contact Center form and the associated library.

The customization must be made on a view or a table which has a reference or intersection with `HZ_PARTIES`. The fields that can be displayed in the custom tabs will be based on this view or table.

The implementer must write public APIs to perform transactions against this table. Code for insert, update and delete must be built into these APIs after the necessary validations.

The form you create can be compiled and run stand-alone. There are no dependencies for running this form.

This topic covers:



## 2.8.1 Guidelines for Setting Up Custom Tabs

These steps provide guidelines to help you with custom tab implementation.

### Prerequisites

Knowledge of Oracle Forms development procedures and an understanding of Oracle TeleService code.

### To set up the two custom tabs:

1. Customize the form using the following guidelines:
   - In the form `CSCCUST1.fmb` and/or `CSCCUST2.fmb`, create a data block with all the fields you wish to display in the custom tab.
   - Code an On Insert trigger for the data block to insert data into the tables.
   - Code an On Update trigger for the data block to update data in the tables.
   - Code an On Delete trigger for the data block to delete data from the tables.
   - The fields in the data block must be placed on the tabbed canvas `CC_MAIN_TAB_CANVAS.CUSTOM1` and/or `CC_MAIN_TAB_CANVAS.CUSTOM2`.

- Any record groups and LOVs must be coded.
- In order to integrate this with the Contact Center, The data block, record groups, and LOVs for `CSCCUST1.fmb` must be dropped in the `CUSTOM1_OBJ_GRP` object group. Those for `CSCCUST2.fmb` must be dropped in the `CUSTOM1_OBJ_GRP2` object group.

2. Modify the library using the following guidelines:
   - You can use any of the following event handlers:

| Trigger Name | Description |
|---|---|
| WHEN-NEW-FORM-INSTANCE | This trigger calls WHEN-NEW-FORM-INSTANCE of the custom form. Code is added to the custom library based on the requirements of the custom form. |
| KEY-CLRFRM | This trigger calls CLEAR_CTRL_FIELDS which clears the control field values populated by values from the header region. |
| WHEN-NEW-FORM-INSTANCE<br>RESTART<br>WHEN-TAB-PAGE-CHANGED | These triggers call the POPULATE_CTRL_FIELDS procedure which populates the control fields from the header region. |

   - All the code to handle API calls can be put in the library `CSCCUST1.pld` and/or `CSCCUST2.pld`.

3. Integrate the forms with the Contact Center:
   a. Copy the forms `CSCCUST1.fmb` and/or `CSCCUST2.fmb` to the $AU_TOP/forms/US directory in the patched environment.
   b. Compile the corresponding library and copy `CSCCUST1.pll` and/or `CSCCUST2.pll` to the `$AU_TOP/`resource directory in the patched environment.
   c. Run the adadmin utility to compile CSC forms and libraries so that the changes made to the custom form and library are reflected in the Contact Center.

## 2.8.2 Parameters Passed from the Contact Center Header

The following parameters are passed from the Contact Center form to the CSCCUST1 and CSCCUST2 forms:

| Parameter | Description |
|---|---|
| Customer_Header_blk.Cust_Party_Type | Customer's party type |
| customer_header_blk.cust_account_number | Customer's account number |
| customer_header_blk.cust_cust_acct_id | Customer's account identification number |
| contact_header_blk.cont_per_party_id | Identification number for the Contact's party of Person |
| contact_header_blk.cont_relationship_id | The identification number for the party relationship |
| contact_header_blk.cont_party_type | Contact's party type |

### 2.8.3 Function Name for Custom Tabs

The function name for `CSCCUST1` is `CSCSFCU1`. For `CSCCUST2`, it is `CSCSFCU2` . These functions are added to the `CSC_CONTACT_CENTER_MENU`. You can use it to hide and show the custom tabs.

By default, the tabs are hidden so you must enable the function in the menu `CSC_CONTACT_CENTER_MENU`.

**Note:** Please back up your custom forms and libraries before applying any future CSC patches for these files so they are not over-written in $AU_TOP/forms/US and $AU_TOP/. After the patch is applied successfully, you must copy the customized files back to these directories and run adadmin to generate CSC forms and libraries.

If Oracle supplies a bug fix in the future for the custom forms and libraries you modify, then you must copy your modifications from the backup to the newer version of the files.

## 2.9 Enabling E-Mail for the Contact Center

E-mail functionality in the Contact Center is provided through the Oracle eMail Center. Please follow the instructions for implementing the Oracle eMail Center described in Implementing Oracle eMail Center on page 1-72.

# 3

# Implementing Charges

This chapter describe the setup and configuration tasks required to successfully implement the Charges module of Oracle TeleService.

It covers:



## 3.1 Overview of Charges

Charges is a component of the Oracle TeleService application which is used by many Oracle applications including Oracle Field Service and Oracle Depot Repair.

Typical business needs that Charges supports are as follows:

- Create orders and shipments
- Create return orders (RMA)
- Bill customers for any services provided to them

Charges has the following basic features and functions:

- Create and view charge lines (orders, returns, and billing)
- Create estimated charge lines
- Roll up charge lines (labor, material, expenses) into a defined item (labor, material, and expenses)
- Associate charge lines with an existing order

- Apply contract terms and conditions to charge lines
- View coverage information for a contract
- Submit charge lines to Oracle Order Management through Quoting - Forms
- Achieve multi-currency compliance

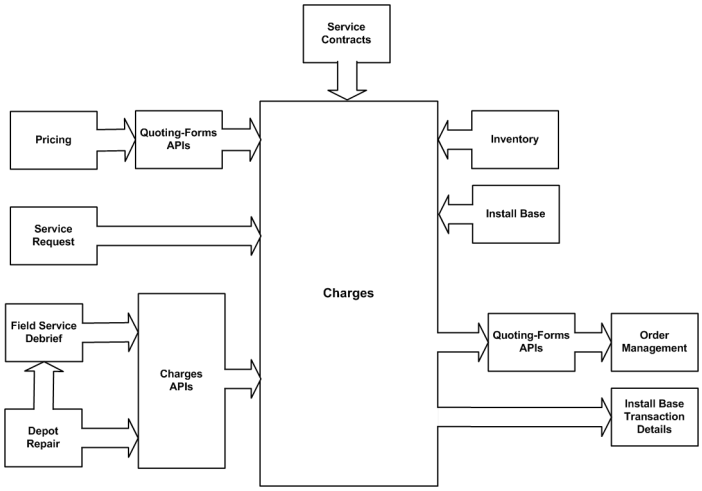
### 3.1.1 Architecture of Charges

Charges windows remain based on the Oracle Forms technology stack.

Charges integrates with the following Oracle applications:

- Oracle Field Service
- Oracle Depot Repair
- Oracle Contracts
- Oracle Quoting - Forms
- Oracle Inventory
- Oracle Receivables
- Oracle Install Base
- Oracle Order Management

The following diagram illustrates the architecture and data flow for Charges:




Charges integrates with Oracle Order Management for returns, shipments, and billing through APIs provided by Oracle Quoting - Forms. When you book an order (shipment or return), Oracle Quoting - Forms APIs pass the order to Oracle Order Management, an application that determines how the order is filled and shipped. Oracle Quoting - Forms APIs then give you a confirmation number that is displayed in the Charges window in the Line Number field. After the invoice cycle is complete, the invoice number and the invoice date are also displayed on the Order Status tabbed page of the Charges window in the Invoice Number and Invoice Date fields respectively. If an order line is split in Oracle Order Management, then Charges displays the status of the first line.

Oracle Service Contracts, Oracle Inventory, and Oracle Install Base integrate directly with Charges, whereas Oracle Field Service Debrief, Oracle Pricing, and Oracle Depot Repair communicate with Charges through Charges APIs. Oracle Pricing communicates with Charges through Oracle Quoting - Forms APIs.

## 3.1.2 Charges Users

Charges has no seeded users, and defining users is not applicable to Charges.

## 3.2 Prerequisites

As a prerequisite to the setup of Charges, confirm the setup of all applications at your site that will use Charges, such as Oracle Field Service.

## 3.3 Implementation and Setup

### 3.3.1 Setup Checklist

This chapter focuses only on setup steps for types and codes required for transaction processing in Charges. The details of each of the steps listed below are covered in the associated section of this chapter:

1. Confirm the Setup of Oracle Inventory Items (Required)
2. Confirm the Setup of Units of Measure (Required)
3. Confirm the Setup of Order Categories (FND_LOOKUPS) (Required)
4. Confirm the Setup of Line Categories (FND_LOOKUPS) (Required)
5. Confirm the Setup of Line Types (Required)
6. Confirm the Setup of the Price List (Required)
7. Confirm the Setup of Return Reason Codes (Required)
8. Confirm the Setup of Currencies (Required)
9. Confirm the Setup of Coverage Templates (Required)
10. Confirm the Setup of Contracts (Required)
11. Confirm the Setup of Service Business Processes (Required)
12. Confirm the Setup of the Charges Sort Orders (FND_LOOKUPS) (Required)
13. Confirm the Setup of the Charges Line Types (FND_LOOKUPS) (Required)
14. Confirm the Setup of the Charges Billing Categories (FND_LOOKUPS) (Required)
15. Set Up Billing Types and Categories (Optional)
16. Set Up Extensible Billing Types (Optional)
17. Confirm the Setup of Service Activity Codes, Billing Types, and Associated Oracle Order Management Mappings (Required)

18. Set Up the Time, Material, and Labor Schedule (Required)
19. Confirm the Setup of the Field Service Report (Required)
20. Confirm the Setup of Oracle Install Base (Required)
21. Confirm the Setup of Oracle Install Base Transaction Types and Sources (Required)
22. Confirm the Proper Setup of Defaulting Rules for Billing Operating Units (Required)
23. Define a Manually Overridable Modifier in Oracle Pricing Manager (Optional)

## 3.3.2 Setup Steps

This section contains information about the setup of types and codes that are required for transaction processing in Charges.

### 3.3.2.1 Confirm the Setup of Oracle Inventory Items (Required)

#### Navigation

Forms mode: Order Management Super User responsibility: (N) Items > Master Items

#### To confirm the setup of inventory items:

By navigating to Master Items, the Organizations window appears so that you can pick an organization for which you are defining items.

Clicking Find brings up a populated Master Item window that can be used to confirm and define an item and its attributes using various tabs. Some tabs are shown when the window opens. More tabs can be accessed by clicking the arrow at the end of the tabs.

The following table shows examples of master items with different attributes.

| Item Description | Serviceable Flag | Service Flag | Shippable Flag | Billing Type | Costed | Priced | Install Base Trackable Flag | Stockable Flag |
|---|---|---|---|---|---|---|---|---|
| Products Repaired/ Upgraded | Y | N | Y | M | N | N | Y | Y |

| Item Description | Serviceable Flag | Service Flag | Shippable Flag | Billing Type | Costed | Priced | Install Base Trackable Flag | Stockable Flag |
|---|---|---|---|---|---|---|---|---|
| Parts used in the Depot/WIP | Y or N | N | Y or N | M | Y | N | N | Y |
| Parts Charges | N | N | N | M | N | Y | Y | N |
| Labor Charges | N | N | N | L | N | Y | N | N |
| Expense Charges | N | N | N | E | N | Y | N | N |
| Service Charges (for example, Major Overhaul, 50,000 Click Service, Oil Change, PM) | N | N | N | L | N | Y | N | N |

### References

*Oracle Inventory User's Guide*

#### 3.3.2.2 Confirm the Setup of Units of Measure (Required)

### Navigation

Forms mode: Order Management Super User responsibility: (N) Setup > UOM > Units

### To confirm the setup of units of measure:

You can use the Units of Measure window to confirm and set up the units of measure.

### References

*Oracle Inventory User's Guide*

#### 3.3.2.3 Confirm the Setup of Order Categories (FND_LOOKUPS) (Required)

The order category attached to the line type is used to populate the charge line's category. You cannot enter or change the order category in the Charges window.

### Navigation

Forms mode: Order Management Super User responsibility: Setup > Quick Codes > Order Management

### To confirm the setup of order categories:

Confirm or define the order categories in the User Name field. The following table shows typical order categories for Oracle Order Management.

| Order Category | Description |
|---|---|
| Order | Sales |
| Return | Return |
| Mixed | Mixed |

## 3.3.2.4 Confirm the Setup of Line Categories (FND_LOOKUPS) (Required)

### Navigation

Forms mode: Order Management Super User: Setup > Quick Codes > Order Management

### To confirm the setup of line categories:

Query for a line category in the User Name field. The following table shows typical line (order) categories for Oracle Order Management.

| Order Category | Description |
|---|---|
| Order | Sales |
| Return | Return |

### References

- *Oracle Order Management Suite Implementation Manual*
- *Oracle Order Management User's Guide*

### 3.3.2.5 Confirm the Setup of Line Types (Required)

#### Navigation

Forms mode: Order Management Super User responsibility: (N) Setup > Transaction Types > Define

#### To confirm the setup of line types:

You enter a line type on each charge line in the Charges window. The line category is used to populate the charge line's category (which you cannot enter or change) and a price list (which you can change).

In the Transaction Types window, the field Transaction Type Code has two values, Order and Line. You use this window to define line types.

If the transaction type code is of type Order, then the values for order category can have three values, Order, Return, and Mixed. If the transaction type code is of type Line, then the values for order category can have two values, Order and Return.

The Charges window creates Oracle Order Management order headers using the order header category for the order header type that is mapped to the billing type.

The following table provides a sample of line types.

| Line Types | Line Category | Price List |
|---|---|---|
| Shipment | Order | Corporate |
| Return | Return | Corporate |
| Charge | Order | Corporate |

#### References

- *Oracle Order Management Suite Implementation Manual*
- *Oracle Order Management User's Guide*

### 3.3.2.6 Confirm the Setup of the Price List (Required)

#### Navigation

Forms mode: Order Management Super User > Pricing > Price Lists > Price List Setup

### To confirm the setup of the price list:

You can confirm or define items to the price list by selecting Order Management Super User: (N) Pricing > Price Lists.

### References

- *Oracle Order Management Suite Implementation Manual*
- *Oracle Order Management User's Guide*

#### 3.3.2.7 Confirm the Setup of Return Reason Codes (Required)

### Navigation

Forms mode: Inventory responsibility: (N) Setup > Transactions > Reasons

### To confirm the setup of return reason codes:

Confirm and define return reason codes as necessary.

### References

- *Oracle Order Management Suite Implementation Manual*
- *Oracle Order Management User's Guide*

#### 3.3.2.8 Confirm the Setup of Currencies (Required)

### Navigation

Forms mode: General Ledger responsibility: (N) Setup > Currencies > Define

### To confirm the setup of currencies:

Use the Currencies window to define the various currencies that you want to use in business transactions.

### References

*Oracle General Ledger User Guide*

#### 3.3.2.9 Confirm the Setup of Coverage Templates (Required)

You can select a contract and an associated coverage on the Charges window for each charge line.

### Navigation

Forms mode: Service Contracts Manager responsibility: (N) Coverage Templates

### To confirm the setup of coverage templates:

Confirm and define the coverage templates as required.

### References

*Oracle Contracts for Service Concepts and Procedures*

## 3.3.2.10 Confirm the Setup of Contracts (Required)

A service contract is associated with a service request and in turn with Charges. The service contract can be defined at the following levels: party, account, system, customer product, and inventory item. A contract and an associated coverage can be selected on the Charges window for each charge line.

### Navigation

Forms mode: Service Contracts Manager responsibility: (N) Launch Contracts > New on the Tools menu. The Create a New Contract window appears.

### To confirm the setup of contracts:

Use the Oracle Contracts window and the Create a New Contract window to confirm or define related contracts. To be used by Charges, contracts must be have the status of Active.

### References

- *Oracle Contracts for Service Concepts and Procedures*
- *Oracle Service Concepts and Procedures*

## 3.3.2.11 Confirm the Setup of Service Business Processes (Required)

A Service business process groups Service activity codes so that you can restrict Service activity code availability. For example, for the field engineers you can define a Service business process called Field Service that allows field engineers only the Service activity types relevant to them.

The Service business process is entered in the Charges window, where it is used to restrict the list of values for the Service activity code. The Business Process Transactions window is used to define Service business processes and associate Service activity codes with them.

### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Charges > Service Business Process

### To confirm the setup of service business processes:

In this step you confirm or define necessary Service business processes. Later, using this window you will associate a Service activity code with each Service business process.

A sample of Service business processes and associated Service activity codes follows:

| Service Business Process | Order Type | Service Request Flag | Depot Repair Flag | Field Service Flag | Service Activity Codes |
|---|---|---|---|---|---|
| Depot Repair/Returns/ Claims | - | Y | Y | Y | All Service activity codes recommended later in this chapter |

This code is entered in the Charges window, where it is used to create the LOV for the Service activity code. It is defaulted from the service request type. When a Service activity code is associated with a Service business process, all billing types for it are available.

### References

*Oracle Install Base Implementation Guide*

## 3.3.2.12 Confirm the Setup of the Charges Sort Orders (FND_LOOKUPS) (Required)

This list determines the contents of the Charges LOV of possible sort orders to display charges lines.

### Navigation

Forms mode: (N) Others > Service: Lookups

### To confirm the setup of the charges sort orders:

Using the lookup type CS_CHG_SORT_ORDER, confirm the following seeded sort values:

- Creation Date
- Service Activity Code (with lookup code TRANSACTION_TYPE)
- Billing Type
- Order Number
- Charge Line Source
- Charge Type

### 3.3.2.13 Confirm the Setup of the Charges Line Types (FND_LOOKUPS) (Required)

This list determines the contents of the LOV of Charges line types.

#### Navigation

Forms mode: (N) Others > Service: Lookups

#### To confirm the setup of the charges line types:

Using the lookup type CS_CHG_LINE_TYPE, confirm the following seeded values:

- Actual
- Estimate
- In Progress

### 3.3.2.14 Confirm the Setup of the Charges Billing Categories (FND_LOOKUPS) (Required)

This list determines the billing categories with which to define new billing types.

#### Navigation

Forms mode: (N) Others > Service: Lookups

#### To confirm the setup of the charges billing categories:

Using the lookup type CS_BILLING_CATEGORY, confirm the following seeded values:

- Material
- Labor
- Expense

### 3.3.2.15 Set Up Billing Types and Categories (Optional)

If you need to create new billing types in addition to the three that are seeded, then complete the following steps.

#### To set up billing types and categories:

1. Navigate to the Lookups setup form, and create the new billing types on the the Lookup Type MTL_SERVICE_BILLABLE_FLAG.
2. Navigate to the Billing Type Attributes mapping window and link the new billing type to one of the three seeded billing categories (Material, Labor, or Expense).
3. Optionally add a roll-up item to the billing type in this window.
4. Define new inventory items that reference the new billing type.
5. Define new Service activity billing types in the Service Activities and Billing Types setup window.
6. Optionally define new contract discounts based on the new Service activity billing type.

### 3.3.2.16 Set Up Extensible Billing Types (Optional)

#### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Charges > Billing Type Attributes

#### To set up extensible billing types:

Use the Billing Type Attributes window to link new billing types to one of the three seeded billing categories of material, labor, and expense. Use it also to establish a rollup item for the billing type. With these types you can create different contract coverages based on the nature of an item. For example, you can define the billing type Consumable Material and Non Consumable Material to have different coverage on a item of billing category Material if the item is consumable rather than non-consumable. Billing types can also be used in contracts to distinguish items with different discounts.

### 3.3.2.17 Confirm the Setup of Service Activity Codes, Billing Types, and Associated Oracle Order Management Mappings (Required)

#### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Charges > Service Activity and Billing Types.

#### Selective Field Descriptions

##### Line Category Code

Line Category Code can be either Order or Return. This is set at the level of Service activity code and prevents a single Service activity code from being an order in one operating unit and a return in another.

##### Order Management Header & Line Types

When a charge line is submitted to Oracle Order Management, the Service activity billing type and the operating unit are used to retrieve an Oracle Order Management header type and line type from the setup. These are used to submit the order.

#### To confirm the setup of service activity codes, billing types, and associated Oracle Order Management mappings:

Use the Service Activities and Billing Types window to confirm or define service activity codes and associated billing types, Oracle Order Management header types, and line types.

For this release, the attributes that control how Oracle Install Base is updated have been moved to the setup for subtypes.

Here is more information about billing types from FND_LOOKUPS:

- M: Material
- L: Labor
- E: Expense

Currently the Item Master window permits assignment of only these codes.

### 3.3.2.18 Set Up the Time, Material, and Labor Schedule (Required)

This step is required to create charge lines from the Field Service Debrief module. Charges uses the schedule to break a labor debrief into multiple charge lines

### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Charges > T&M Labor Schedule

### To set up the time, material, and labor schedule:

1. Use the Business Process LOV to select a Service business process. The window displays existing values for that selection.
2. As necessary, edit the start and end times, applicable days, and billing rate for a selected item.
3. Select Validate Setup to check your entries.

   If your entries form a complete schedule for the selected business process, then a note appears that says that your schedule is validated, and a Y appears in the Schedule Complete field. If your schedule is not complete, then a note appears that states this, and an N appears in the Schedule Complete field.
4. Repeat editing and validating as necessary.

## 3.3.2.19 Confirm the Setup of the Field Service Report (Required)

This step is required only if you have installed Oracle Field Service.

### References

*Oracle Field Service Implementation Guide*

## 3.3.2.20 Confirm the Setup of Oracle Install Base Instance Statuses (Required)

### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Installed Base > Instance Status

### To confirm the setup of Oracle Install Base instance statuses:

Refer to the latest *Oracle Install Base Implementation Guide* for the current table of status codes. Use the Instance Statuses window to confirm and define these codes and the related flags.

Here is more information about the flags:

- Status Change Allowed not selected: Install Base update programs do not update the status code for the Install Base item.

- Incident Allowed not selected: Users cannot open a Service Request against the Oracle Install Base item.
- Service Order Allowed not selected: Order Management prevents sale of a contract or extended warranty for the Oracle Install Base item.

#### References

*Oracle Install Base Implementation Guide*

### 3.3.2.21 Confirm the Setup of Oracle Install Base Transaction Types and Sources (Required)

#### Navigation

Forms mode: Customer Support responsibility: (N) Setups > Charges > Install Base Transaction Types

#### To confirm the setup of Oracle Install Base transaction types and sources:

Confirm the setup of Oracle Install Base transaction types and subtypes and their sources.

#### References

*Oracle Install Base Implementation Guide*

### 3.3.2.22 Confirm the Proper Setup of Defaulting Rules for Billing Operating Units (Required)

#### Navigation

Forms mode: Customer Support responsibility: (N) Setup > Charges > Multi Org Setup

#### To confirm the proper setup of defaulting rules for billing operating units:

Use the Service Request Multi Org Set Up window to set up the defaulting rules for the Charges operating unit. With this window you can select which rules to use and their precedence.

Typically your system administrator uses this window to assign a sequence to the four available rules. Based on these rules, an operating unit is derived for use in a calling window, currently from Charges and Field Service. The following table describes the rules.

| Rule Name | Description |
|---|---|
| Operating Unit of the Contract | If a contract is associated with the service request, then the operating unit of the contract is used. |
| Operating Unit of the Installed Base Product | If an Oracle Install Base item is associated with the service request, then the operating unit of the Oracle Install Base item is used. |
| Profile Option MO: Operating Unit of the Charge | The operating unit set in this profile is used. |
| Operating Unit Stamped on the Service Request | The operating unit associated with the service request is used. |

Based on the sequence associated with these rules, the appropriate operating unit appears in the Charges window and the Field Service window.

### Example

If a user has set sequence 1 for rule b and 2 for rule d, then the program to derive the operating unit first checks if rule b returns any operating unit. If yes, it returns that operating unit to the calling program. If no, it checks if rule d returns any operating unit. If none of the rules returns any operating unit, then the program returns null.

## 3.3.2.23 Define a Manually Overridable Modifier in Oracle Pricing Manager (Optional)

If you will be adding freight charges in Oracle Order Management and modifying the Calculate Price flag in Oracle Order Management, then you need to complete the following additional setup. If you do not modify the Calculate Price flag in Oracle Order Management, then you do not need to complete these steps.

### Navigation

Forms mode: Oracle Pricing Manager responsibility: (N) Modifiers > Modifier Setup

### To define a manually overridable modifier in Oracle Pricing Manager:

1. In the Define Modifier window, add a modifier that has the following attributes:
   - Header level Automatic flag cleared
   - Header level Type set to Discount List

- Line level Modifier Type set to Discount
- Application Method set to Percent
- No qualifiers or items attached to it

For details on modifiers, see the *Oracle Advanced Pricing Users Guide*.

2. Set the profile Service: Default Manual Modifier to the modifier created in step 1.

### References

*Oracle Advanced Pricing User's Guide*

## 3.4 Profile Options

Use the following procedure to set up the profile options for Charges that are listed in the table below.

### To set profile options:

1. Select the forms mode and the System Administrator responsibility.
2. Navigate Profile > System.
3. Query for the name of the desired profile option for Charges.
4. Confirm or set profile options for Charges according to the following table.

| Profile Option Name | Possible Values | User | Rest | App | Site | Functionality |
|---|---|---|---|---|---|---|
| Service: Allow Charge Operating Unit Update | Yes and No | x | x | x | x | Controls whether the operating unit on the charge line can be updated. |
| Service: Charges Default Conversion Type | - | - | - | - | - | Specifies the default conversion type for converting the transactional currency from Service Debrief to the billing currency that is used in Charges. This is used only on expense lines. Possible values: Corporate, EMU Fixed, Spot, and User. |
| Service: Charges Default Interface Flag | - | - | - | - | - | Sets the default value of the Order Management Interface check box. By default, this profile is set to Y. |

| Profile Option Name | Possible Values | User | Rest | App | Site | Functionality |
|---|---|---|---|---|---|---|
| Service: Charges Default Sort Order | - | - | - | - | - | Sets the default profile value for the default sort order. By default, this profile is set to CREATION_DATE. |
| Service: Charges - Default Max Roll Days | - | - | - | - | - | Specifies the maximum roll-up days in the past that can be used while checking a rate for conversion. This is used for converting the transactional currency from Service Debrief to the billing currency that is used in Charges. This is used only for expense lines. Possible values: Any positive integer |
| Service: Charges - Default Service business process from SR | Yes and null | - | - | - | - | If this profile option is set to `Yes`, then the Charges window defaults the Service business process from the service request type. If the profile option is not set, then the Charges window does not default the Service business process. |
| Service: Charges - Default Interface to OM Flag | Yes and No | x | x | x | x | Specifies the default value for the OM Interface checkbox. Seeded with a value of Y.<br><br>This check box indicates whether a charge line is ready to be processed by Oracle Order Management. After the box is selected, the charge line will be processed the next time the Submit button is selected. |
| Service: Charges - Default Price List | - | x | x | x | x | Sets the default price list in the Charges window |
| Service: Charges - Default Sort Order | All lookups of type CS_CHARGE_SORT_ ORDER | x | x | x | x | Specifies the default sort order of charge lines. Seeded with a value of Creation Date. |
| Service: Default Manual Modifier | Any modifier defined in the modifier setup of Pricing Manager | - | - | - | - | Controls the ability to add freight charges in Oracle Order Management when modifying the Calculate Price flag. |

# A

# Integrating Relationship Plans in Other Applications

This appendix details the necessary steps involved in integrating any application within the Oracle E-Business Suite to the relationship plans functionality offered by Oracle TeleService. Integration can be achieved by coding one or more of the following form level triggers in the integration form:

- On Insert
- On Update
- Custom1
- Custom2

## A.1 Overview of Integrating Relationship Plans

Modules within the CRM application need to be integrated with each other to ensure smooth flow of data and business logic through out the application. The relationship plans module within Customer Care allows organizations to set up plans that they can offer to their customers and ensure a constantly growing customer base.

The relationship plans module is a busy point of integration, because of the very fact that almost every other module within CRM needs to have a consistent view of all the existing plans in the application. To help in this integration process, the relationship plans module facilitates an "ENABLE PLAN" form, which enables other modules to integrate with relationship plans.

# A.2 Technical Overview

## A.2.1 Forms Used:

CSCUTILS.fmb - Setup Form and Outcomes Window

## A.2.2 Tables Used:

CSC_PLAN_ENABLE_SETUP

OKC_PROCESS_DEFS_V

## A.2.3 List of Packages and Procedures:

```
PACKAGE CSC_PLAN_OUTCOMES
PROCEDURE GET_OUTCOMES (
p_FUNCTION_ID                          NUMBER,
p_TRIGGER_EVENT                        VARCHAR2,
p_PARTY_ID                             NUMBER,
p_CUST_ACCOUNT_ID                      NUMBER,
p_APPLICATION_SHORT_NAME               VARCHAR2,
p_MSG_TBL                              OKC_AQ_PVT.MSG_TAB_TYP)
```

### Description of Parameters:

- P_FUNCTION_ID - The integrating form function id from table FND_FORM_FUNCTIONS
- P_TRIGGER_EVENT - The triggering event that calls the outcomes. The allowed values are INSERT, UPDATE, CUSTOM1, and CUSTOM2.
- P_PARTY_ID - Party Id from HZ_PARTIES.
- P_CUST_ACCOUNT_ID - Customer account Id from HZ_CUST_ACCOUNTS_ALL
- P_APPLICATION_SHORT_NAME - The short name of the application from FND_APPLICATION.
- P_MSG_TBL - The msg_tbl has the following structure:

  **a.** ELEMENT_NAME VARCHAR2(4000)

  **b.** ELEMENT_VALUE VARCHAR2(4000)

  The element name is the name of the action attribute element name. The element value is the actual value of the element name.

## A.3 Prerequisites

1. Relationship Plans have to be defined and attached to the specific customer.
2. Actions, Conditions, and Outcomes have to be defined using the Events screens.

## A.4 Steps to Enable Modules in Relationship Plans

1. Register the integrating module form with the relationship plan's setup form.
2. Attach library `CSCUTILS.pll` (from `$CSC_TOP/resource`) in the integrating form.
3. Subclass relationship_plans_outcomes and enable_relationship_plans object groups from the `CSCUTILS.fmb` form (from `$CSC_TOP/forms/US`).
4. Call `CSC_PLAN_OUTCOMES.GET_OUTCOMES` procedure from all of the above mentioned form level triggers. The outcomes are fired only for triggers that are enabled in the setup form.
5. Add the following code to the `APP_CUSTOM` package body in the integrating form:

```
if (wnd='OUTCOMES') then
cscutils_app_custom.close_windo('OUTCOMES');
--move focus to required block/field. for eg:
GO_BLOCK('target block');
end if;
```

6. User defined triggers, Custom1 and Custom2 have to be explicitly executed from the required integration point. Following is an example:

```
execute_trigger('CUSTOM1');
```

## A.5 Functionality of the Setup Form - CSCUTILS

### A.5.1 Enable Relationship Plans - Setup Window:

To enable a module for relationship plans, the module should register its form functions with the relationship plans registering table called `CSC_PLAN_ENABLE_SETUP`. This can be done via the Enable Relationship Plans - Setup form.

The Function column specifies the integrating form function name.

The Application column specifies the application the form function belongs to.

The integrating module has four form level triggers which can be coded to achieve integration with the relationship plans module. They are:

1. On Insert
2. On Update
3. Custom1
4. Custom2

By checking one or more of these triggers, the enable plan APIs can be executed from the integrating module.

## A.5.2 Outcomes Window

Outcomes of a plan can be of two types - synchronous and asynchronous. Asynchronous outcomes are those that are fired immediately but the resulting outcomes are processed in the background. Synchronous outcomes are those which are fired immediately and the results are shown in the outcomes window. The user can then select or deselect the outcomes and execute them.

Outcome Name is the name of the outcome.

Outcome Type is the name of the outcome type, such as Alert or Script.

Description is the description of the outcome, such as the alert text for alerts.

The Execute button executes the outcome.

The Cancel button closes the Outcome window.

## A.5.3 Defining an Alert

1. Create a message with the alert text as the message text. This is the same as defining an error message.
2. Define the outcome in the Process Definition screen. Use the same message name defined in step 1 for the outcome name.
3. Define the parameters for the outcomes. Use the same message token names defined step 1 as the parameter names.
4. For more information about defining outcomes, refer to Events documentation.

## A.6 Limitations

Defining a plan which has only one outcome to be fired raises no issues. The outcome is fired and the control passes on normally to the next logical step. If there are two or more outcomes to be fired, the control passes to the 'OUTCOMES' block which lists the different outcomes and the user can select which of the outcomes are to be executed. This raises a potential error situation if the enable relationship plans triggers are called from triggers that do no allow restricted procedures to be part of their code. This is due to the fact that there is a 'GO_BLOCK' statement executed in the GET_OUTCOMES program unit when transferring control to the OUTCOMES block to display the various outcomes.

You must make sure that the form functions integrating with the Relationship Plans module takes into account the above situation and performs the right code calls. A workaround for the above scenario would be to create a timer and then execute the Enable Relationship plan trigger from the 'WHEN-TIME-EXPIRED' trigger.

# B

# Frequently Asked Questions About Setting Up Relationship Plans

### What does Plan Type mean?

Relationship Plans can be of two types:

- Template Plan - These types of plans can be associated to any customer.
- Customized Plan - This type of plan is created only for specific customers.

### Can a customized plan be created when the Define Relationship Plan responsibility is invoked?

No, a customized plan can be created only when a plan already exists and is associated to a group of customers. This plan can then be customized for a specific sub-group of those customers. When a plan is customized, a new plan in created in the application.

### What is the Account Level check box in the Define Plan form?

Plans can be associated to customers at two levels.

- Party Level, in which case this check box is left unchecked. Party Level plans can be associated to customers only at party level.This means the plan is associated to all the customer accounts as well.
- Account Level, in which case this check box is checked. Account Level plans can be associated to customers only at account level. This means the plan is associated to a specific account of the customer and not to all the accounts.

### What is the Group Name field in the Define Plan form?

Plans can be grouped together to form a logical set of plans. This group of plans can then be associated to customers and the customer gets the benefits and outcomes of

all the plans in the group. This functionality is not yet implemented in the current release.

### Sometimes the Low Value and High Value values cannot be modified. Why is this?

These fields define the Relationship Plan header criteria. For example, consider the following Plan definition:

**Open Service Requests > 5**

This plan will be associated to customers who have more than five open service requests. Assume that Customer A has eight open service requests. If you try to modify the plan criteria so the high value is 10 instead of 5, this will not be allowed because Customer A, who has eight open service requests is already associated to the plan and would not satisfy the new plan criteria.

This update is allowed in the following scenarios:

- The plan is not associated to any customers.
- The new plan criteria does not violate any of the existing customer-plan associations.

### The Relationship Plan is setup and the customer is associated to the plan, but the plan does not execute.

Relationship Plans are currently integrated with the Contact Center and Service Request forms. No other form can execute plans. If the plan is failing to execute from these two forms, check the following:

Navigate to Customer Support > Setup > Relationship Plans > Enable Relationship Plans and verify if the following entries exist and that the Custom1 check box is checked:

| Function | Application |
|---|---|
| CSCCCCRC | Customer Care |
| CSXSRISR | Oracle Service |

### How do I set up Relationship Plans to pop up an alert?

In this example, we are creating a new message called RV30 and specifying the message text that needs to pop up in the alert window.

1. Navigation Path: Application Developer > Messages

2. Define an action called RV30 with the following attributes:
   - Action Type = Action Based
   - Correlation = CUSTOMER_RV30
   - Check the Enabled and Allow Synchronous Outcomes check boxes.
   - Name = CUSTOMER NAME
   - Element = CUST_NAME
   - Navigation Path: Customer Support > Setup > Relationship Plans > Events > Define Actions
3. Define a Process Definition (Outcome) called RV30 with the following attributes using this navigation path:

   Customer Support: Setup > Relationship Plans > Events > Define Process Definition
   - Purpose = "OUTCOME"
   - Type = "ALERT"
   - Name = "CUST_NAME"

   The process definition name should be the same as the message name).
4. Define a relationship plan called RV30. The plan header criteria is 'Open Service Request > 10'. The navigation path is Customer Support: Setup > Relationship Plans > Define Relationship Plans
5. In the Plan Details tab, create a new condition called RV30 using the action called RV30 and outcome RV30.
6. Click Parameters. In the Parameters window, right click and select 'Populate selected rows in parameters'. Invoke the list of values to populate the value for the Action Attribute field.
7. Attach the plan to the customer. In the Relationship Plans Search Screen, search for the customer to which the plan is to be attached.
8. Click **OK** to invoke the Plans Summary window.
9. Click **Add/Remove** to invokes the Add/Remove Plans window. This window displays all the plans the customer can be associated to. Select RV30 to add the plan called RV30 to the customer.
10. Open the Contact Center window and query the Customer. The Plan's alert message should pop up.

### What manuals are available that describe Relationship Plans?

- *Oracle TeleService Implementation Guide*
- *Oracle TeleService User Guide*
- *Oracle Service Contracts Concepts and Procedures*

### Why do extra outcomes sometime fire?

The same action is being used in multiple conditions.

# C

# Seed Data for Relationship Plans

The Expiring Contracts relationship plan is included with this release. This relationship plan is designed to display an alert when an agent views a record of a customer who has one or more contracts that will expire in the next 30 days.

> **Note:** Do not modify seed data. Seed data is provided for you to use as delivered. If you want to make modifications, please make a copy and modify your copy.

This appendix describes the following:

- Relationship Plan Header
- Condition Header
- Condition Type
- Outcome
- Parameter
- Outcome Header
- Outcome Argument
- Message Definition
- Profile Definition
- Action Header
- Action Attributes

### Relationship Plan Header

- Plan Name - Expiring Contracts
- Plan Type - Template
- Description - Plans that display an alert when querying a customer who has one or more contracts that will expire in the next 30 days
- Profile Check - Expiring Contracts
- Operator - >
- Low Value - 0

### Condition Header

- Condition Name - Customer Care - Condition for Expired Contracts.
- Description - Seeded condition for Relationship Plans.

### Condition Type

Action - Customer Care - Generic Action

### Outcome

CSC_EXPIRING_CONTRACTS

### Parameter

- Parameter - CUST_PARTY_NAME
- Description - Name of the customer who has one or more contracts that will expire in the next 30 days.
- Data Type - CHAR
- Action Attribute - CUST_PARTY_NAME

### Outcome Header (Process Definition)

- Name - CSC_EXPIRING_CONTRACTS
- Application - Oracle Customer Care
- Description - Outcome that will be fired when a customer has one or more contracts that will expire within the next 30 days. An alert message will pop up warning the agent about this.
- Purpose - Outcome

- Type - Alert
- Comments - Outcome for a relationship plan that fires when a customer has one or more contracts that will expire in the next 30 days.

### Outcome Argument

- Name - CUST_PARTY_NAME
- Data Type - Char
- Description - Name of the customer who has one or more contracts that will expire in the next 30 days.

### Message Definition

Message Name - CSC_EXPIRING_CONTRACTS

Message Tokens - CUST_PARTY_NAME

Message Description - Message used as an outcome in a Relationship Plan that fires when a customer has one or more contracts that will expire in the next 30 days.

Message Text - Customer &CUST_PARTY_NAME has one or more contracts that will expire in the next 30 days.

### Profile Definition

Block:

- Name - Expiring Contracts
- Code - EXPIRINGCONTRACTS
- Description - Number of contracts that will expire in the next 30 days.
- Application - Oracle Customer Care
- Select - count(*)
- From - OKS_ENT_HEADERS_V A
- Where - (a.party_id =:party_id and:cust_account_id is null) and a.end_date_active <= sysdate + 30

Check

- Name - Expiring Contracts
- Code - EXPIRINGCONTRACTS

- Description - Number of contracts that will expire in the next 30 days.
- Application - Oracle Customer Care
- Type - Variable
- Variable - Expiring Contracts

### Action Header

Action Name - Customer Care - Generic Action

Action Type - Action Based

Description - Customer Care generic action that has all the attributes from the contact center header region as well as some from SR, that can be used as additional condition or as arguments to an outcome.

Correlation - CUSTOMER_CARE_ACTION

Application - Oracle Customer Care

Enable - Yes

Counter Action - No

Allow Synchronous Outcomes - Yes

### Action Attributes

The following is a list of the Action Attribute Name, Element Name, Description and Data Type for the attributes provided with the seeded Action named Customer Care - Generic Action.

- Customer Party Id, CUST_PARTY_ID, Customer Party Id, NUMBER
- Customer Party Number, CUST_PARTY_NUMBER, Customer Party Number, CHAR
- Customer Name, CUST_PARTY_NAME, Customer Party Name, CHAR
- Customer Type, CUST_PARTY_TYPE, Customer Party Type, CHAR
- Customer Status, CUST_PARTY_STATUS, Customer Party Status, CHAR
- Customer Account Id, CUST_ACCOUNT_ID, Customer account Id, NUMBER
- Contact Party Id, CONT_PARTY_ID, Contact Party Id, NUMBER
- Contact Party Number, CONT_PARTY_NUMBER, Contact Party Number, CHAR

- Contact First Name, CONT_FIRST_NAME, Contact First Name, CHAR
- Contact Last Name, CONT_LAST_NAME, Contact Last Name, CHAR
- Area Code, AREA_CODE, Area Code, CHAR
- Phone Number, PHONE_NUMBER, Phone Number, CHAR
- Extension, EXTENSION, Extension, CHAR
- E-mail Address, EMAIL, E-mail Address, CHAR
- Address, ADDRESS, Address, CHAR
- City, CITY, City, CHAR
- State, STATE, State, CHAR
- Postal Code, POSTAL_CODE, Postal Code, CHAR
- Province, PROVINCE, Province, CHAR
- Country, COUNTRY, Country, CHAR
- Caller Type, CALLER_PARTY_TYPE, Caller Type, CHAR
- Caller Status, CALLER_PARTY_STATUS, Caller Status, CHAR
- Category Code, CATEGORY_CODE, Category Code, CHAR
- Country Code, COUNTRY_CODE, Country Code, CHAR
- Contact Relation, CONTACT_RELATION, Contact Relation, CHAR
- Contract Id, CONTRACT_ID, Contract Id, NUMBER
- Contract Line Id, CONTRACT_LINE_ID, Contract Line Id, NUMBER
- Critical Overidden Flag, CRIT_OVERRIDE_FLAG, Critical Overidden Flag, CHAR
- Relation Meaning, RELATION_MEANING, Relationship code meaning, CHAR
- Account Number, ACCOUNT_NUMBER, Account Number, CHAR
- Contract Number, CONTRACT_NUMBER, Contract Number, NUMBER
- Contract Line Number, CONTRACT_LINE_NUMBER, Contract Line Number, NUMBER

# D

# Service Request Public APIs

This appendix provides information on service-request related public APIs for Oracle TeleService.

It includes details on the two service request-related public packages:

- CS_SERVICEREQUEST_PUB used for creating and updating service requests.
- CS_INCIDENTLINKS_PUB used for linking a service request to other service requests and to other Oracle E-Business Suite business objects. See Service Request Linking Package (CS_INCIDENTLINKS_PUB) on page D-9.

Other topics covered include:



## D.1 Service Request Package (CS_SERVICEREQUEST_PUB)

This topic describes the CS_SERVICEREQUEST_PUB package.

It covers:

## D.1.1 Package Overview

The service request APIs in this package are used to:

- Create a service request.
- Update a service request.
- Attach notes to a service request.
- Link a knowledge management statement to a service request.
- Link a knowledge management solution to a service request.
- Create contacts for a service request.

During create and update, the service request APIs raise the business events described in Section 1.23, "Setting Up Notifications" on page 1-62.

### Version Information

The current version of the service request APIs is 3.0. To make use of the 11.5.9 features, users must use the new signatures and call the APIs with the new version.

The API is backward-compatible with version 2.0 and users can use the APIs with the same signature as the 2.0 version.

### Procedures

The service request public package (CS_SERVICEREQUEST_PUB) contains the following procedures:

| Procedure Name | Description |
|---|---|
| Create ServiceRequest | This procedure creates a service request. If contact information is provided, the API creates contacts as well. |
| Update ServiceRequest | This procedure updates a service request with the given parameters. |
| Update Status | This procedure updates the status of a service request. |
| Update Owner | This procedure updates the owner of a service request. |
| Link KB Statement | This procedure links an Oracle Knowledge Management statement with a service request. |
| Link KB Solution | This procedure links an Oracle Knowledge Management solution with a service request. |

## D.1.2 Create_ServiceRequest Procedure Specifications

```
PROCEDURE Create_ServiceRequest
( p_api_version         IN         NUMBER,
  p_init_msg_list       IN         VARCHAR2   := FND_API.G_FALSE,
  p_commit              IN         VARCHAR2   := FND_API.G_FALSE,
  x_return_status       OUT NOCOPY VARCHAR2,
  x_msg_count           OUT NOCOPY NUMBER,
  x_msg_data            OUT NOCOPY VARCHAR2,
  p_resp_appl_id        IN         NUMBER     := NULL,
  p_resp_id             IN         NUMBER     := NULL,
  p_user_id             IN         NUMBER     := NULL,
  p_login_id            IN         NUMBER     := NULL,
  p_org_id              IN         NUMBER     := NULL,
  p_request_id          IN         NUMBER     := NULL,
  p_request_number      IN         VARCHAR2   := NULL,
  p_service_request_rec IN         service_request_rec_type,
  p_notes               IN         notes_table,
  p_contacts            IN         contacts_table,
  p_auto_assign         IN         VARCHAR2  Default 'N',
  x_request_id          OUT NOCOPY NUMBER,
  x_request_number      OUT NOCOPY VARCHAR2,
  x_interaction_id      OUT NOCOPY NUMBER,
  x_workflow_process_id OUT NOCOPY NUMBER,
  x_individual_owner    OUT NOCOPY NUMBER,
  x_group_owner         OUT NOCOPY NUMBER,
  x_individual_type     OUT NOCOPY VARCHAR2
);
```

### D.1.2.1 Parameter Descriptions

This section describes the parameters for the Create_Servicerequest procedure.

#### Create_Servicerequest In Parameters

This table lists the In parameters.

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| p_api_version | NUMBER | Yes | Version Number |
| p_init_msg_list | VARCHAR2 | Optional | Initialize message stack |
| p_commit | VARCHAR2 | Optional | Commit work |
| p_resp_appl_id | NUMBER | Optional | Responsibility Application ID |

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| p_resp_id | NUMBER | Optional | Responsibility ID |
| p_user_id | NUMBER | Optional | User ID |
| p_login_id | NUMBER | Optional | Login ID |
| p_org_id | NUMBER | Optional | Org ID in which the SR is created |
| p_request_id | NUMBER | Optional | Request ID of the SR to be created/updated |
| p_request_number | VARCHAR2 | Optional | Request Number of the SR to be created/updated |
| p_service_request_rec | SERVICE_ REQUEST_ REC_TYPE | Yes | Service Request Record Type which holds the SR attributes |
| p_notes | NOTES_TABLE | Yes | PL/SQL table used to create notes in JTF_NOTES |
| p_contacts | CONTACTS_ TABLE | Yes | PL/SQL table used to create contacts in CS_HZ_SR_ CONTACT_POINTS |

### Create_Servicerequest Out Parameters

The following table describes the out parameters associated with this procedure.

| Parameter | Data Type | Description |
|---|---|---|
| x_return_status | VARCHAR2 | Return status of the API |
| x_msg_count | NUMBER | Message Count |
| x_msg_data | VARCHAR2 | Message Data |
| x_request_id | NUMBER | Request ID of the SR which is created |
| x_request_number | VARCHAR2 | Request Number of the SR which is created |
| x_interaction_id | NUMBER | Interaction ID created by the SR |

## D.1.3 Update_ServiceRequest Procedure Specifications

```
PROCEDURE Update_ServiceRequest(
```

```
  p_api_version            IN     NUMBER,
  p_init_msg_list          IN     VARCHAR2      := FND_API.G_FALSE,
  p_commit                 IN     VARCHAR2      := FND_API.G_FALSE,
  x_return_status          OUT NOCOPY VARCHAR2,
  x_msg_count              OUT NOCOPY NUMBER,
  x_msg_data               OUT NOCOPY VARCHAR2,
  p_request_id             IN     NUMBER        := NULL,
  p_request_number         IN     VARCHAR2      := NULL,
  p_audit_comments         IN     VARCHAR2      := NULL,
  p_object_version_number  IN     NUMBER,
  p_resp_appl_id           IN     NUMBER        := NULL,
  p_resp_id                IN     NUMBER        := NULL,
  p_last_updated_by        IN     NUMBER,
  p_last_update_login      IN     NUMBER        := NULL,
  p_last_update_date       IN     DATE,
  p_service_request_rec    IN     service_request_rec_type,
  p_notes                  IN     notes_table,
  p_contacts               IN     contacts_table,
  p_called_by_workflow     IN     VARCHAR2      := FND_API.G_FALSE,
  p_workflow_process_id    IN     NUMBER        := NULL,
  x_workflow_process_id    OUT NOCOPY NUMBER,
  x_interaction_id         OUT NOCOPY NUMBER
);
```

### D.1.3.1 Parameter Descriptions

This section describes the parameters for the Update_Servicerequest procedure.

### Update_Servicerequest In Parameters

This table lists the In parameters associated with this procedure.

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| p_api_version | NUMBER | Yes | Version Number |
| p_init_msg_list | VARCHAR2 | Optional | Initialize message stack |
| p_commit | VARCHAR2 | Optional | Commit work |
| p_resp_appl_id | NUMBER | Optional | Responsibility Application ID |
| p_resp_id | NUMBER | Optional | Responsibility ID |
| p_user_id | NUMBER | Optional | User ID |
| p_login_id | NUMBER | Optional | Login ID |

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| p_org_id | NUMBER | Optional | Org ID in which the SR is created |
| p_request_id | NUMBER | Optional | Request ID of the SR to be created/updated |
| p_request_number | VARCHAR2 | Optional | Request Number of the SR to be created/updated |
| p_service_request_rec | SERVICE_REQUEST_REC_TYPE | Yes | Service Request Record Type which holds the SR attributes |
| p_notes | NOTES_TABLE | Yes | PL/SQL table used to create notes in JTF_NOTES |
| p_contacts | CONTACTS_TABLE | Yes | PL/SQL table used to create contacts in CS_HZ_SR_CONTACT_POINTS |
| p_object_version_number | NUMBER | Yes | Version number of SR being updated |
| p_last_updated_by | NUMBER | Optional | User_Id |
| p_last_update_login | NUMBER | Optional | Login ID |
| p_last_update_date | DATE | Optional | Update date |
| p_called_by_workflow | VARCHAR2 | Optional | Whether this update is being called from Workflow |
| p_workflow_process_id | NUMBER | Optional | Process ID of workflow performing the update |

### Update_Servicerequest Out Parameters

The following table lists the out parameters associated with this procedure:

| Parameter | Data Type | Description |
|---|---|---|
| x_return_status | VARCHAR2 | Return status of the API |
| x_msg_count | NUMBER | Message Count |
| x_msg_data | VARCHAR2 | Message Data |
| x_interaction_id | NUMBER | Interaction ID created by the SR |

| Parameter | Data Type | Description |
|---|---|---|
| x_workflow_process_id | NUMBER | If a new Oracle Workflow process gets launched during update, then this parameter stores the workflow process ID. New workflow gets launched only if the service request does not already have an active workflow associated with it. |

## D.1.4 Update_Status Procedure Specifications

```
PROCEDURE Update_Status
( p_api_version           IN  NUMBER,
  p_init_msg_list         IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit                IN  VARCHAR2 := FND_API.G_FALSE,
  x_return_status         OUT NOCOPY VARCHAR2,
  x_msg_count             OUT NOCOPY NUMBER,
  x_msg_data              OUT NOCOPY VARCHAR2,
  p_resp_appl_id          IN  NUMBER   := NULL,
  p_resp_id               IN  NUMBER   := NULL,
  p_user_id               IN  NUMBER   := NULL,
  p_login_id              IN  NUMBER   := FND_API.G_MISS_NUM,
  p_request_id            IN  NUMBER   := NULL,
  p_request_number        IN  VARCHAR2 := NULL,
  p_object_version_number IN  NUMBER,
  p_status_id             IN  NUMBER   := NULL,
  p_status                IN  VARCHAR2 := NULL,
  p_closed_date           IN  DATE     := FND_API.G_MISS_DATE,
  p_audit_comments        IN  VARCHAR2 := NULL,
  p_called_by_workflow    IN  VARCHAR2 := FND_API.G_FALSE,
  p_workflow_process_id   IN  NUMBER   := NULL,
  p_comments              IN  VARCHAR2 := NULL,
  p_public_comment_flag   IN  VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id        OUT NOCOPY NUMBER
);
```

## D.1.5 Update_Owner Procedure Specifications

```
PROCEDURE Update_Owner
( p_api_version         IN  NUMBER,
  p_init_msg_list       IN  VARCHAR2 := FND_API.G_FALSE,
  p_commit              IN  VARCHAR2 := FND_API.G_FALSE,
  x_return_status       OUT NOCOPY VARCHAR2,
  x_msg_count           OUT NOCOPY NUMBER,
  x_msg_data            OUT NOCOPY VARCHAR2,
```

```
  p_resp_appl_id        IN  NUMBER   := NULL,
  p_resp_id             IN  NUMBER   := NULL,
  p_user_id             IN  NUMBER   := NULL,
  p_login_id            IN  NUMBER   := FND_API.G_MISS_NUM,
  p_request_id          IN  NUMBER   := NULL,
  p_request_number      IN  VARCHAR2 := NULL,
  p_object_version_number IN NUMBER,
  p_owner_id            IN  NUMBER,
  p_owner_group_id      IN  NUMBER,
  p_resource_type       IN  VARCHAR2,
  p_audit_comments      IN  VARCHAR2 := NULL,
  p_called_by_workflow  IN  VARCHAR2 := FND_API.G_FALSE,
  p_workflow_process_id IN  NUMBER   := NULL,
  p_comments            IN  VARCHAR2 := NULL,
  p_public_comment_flag IN  VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id      OUT NOCOPY NUMBER
);
```

## D.1.6 Link_KB_Statement Procedure Specifications

```
PROCEDURE Link_KB_Statement
(
 p_api_version            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,
 x_return_status          OUT NOCOPY VARCHAR2,
 x_msg_count              OUT NOCOPY NUMBER,
 x_msg_data               OUT NOCOPY VARCHAR2,
 p_request_id             IN  NUMBER,
 p_statement_id           IN  NUMBER,
 p_is_statement_true      IN  VARCHAR2,
 x_statement_link_id      OUT NOCOPY NUMBER
);
```

## D.1.7 Link_KB_Solution Procedure Specifications

```
PROCEDURE Link_KB_Solution
(
 p_api_version            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,
 x_return_status          OUT  NOCOPY VARCHAR2,
 x_msg_count              OUT  NOCOPY NUMBER,
```

```
 x_msg_data              OUT  NOCOPY VARCHAR2,
 p_request_id            IN   NUMBER,
 p_solution_id           IN   NUMBER,
 p_is_solution_true      IN   VARCHAR2,
 x_solution_link_id      OUT  NOCOPY NUMBER
);
```

# D.2 Service Request Linking Package (CS_INCIDENTLINKS_PUB)

This topic describes the CS_INCIDENTLINKS_PUB package.

It covers:



## D.2.1 Package Overview

The CS_INCIDENTLINKS_PUB package allows for the creation, updating, and deletion of related documents for a service request.

The service request relationships APIs are used to:

- Create a relationship between a service request and another service request.
- Create a relationship between a service request and an external object.
- Delete the relationship between a service request and another service request.
- Delete the relationship between a service request and an external object.
- Update the relationship between a service request and another service request.
- Update the relationship between a service request and an external object.

### Version Information

The API is backward-compatible with version 1.2 and users can use the APIs with the same signature as the 1.2 version.

### Procedures

The relationships public package (CS_INCIDENTLINKS_PUB) contains the following procedures:

| Procedure Name | Description |
|---|---|
| Create Incident Link | This procedure creates a link between a service request and another service request or an external object. |
| Delete Incident Link | This procedure deletes the link between a service request and another service request or an external object. |
| Update Incident Link | This procedure updates the link between two service requests or between a service request and an external object. |

## D.2.2 CREATE_INCIDENTLINK Procedure Specifications

```
PROCEDURE CREATE_INCIDENTLINK (
      P_API_VERSION          IN     NUMBER,
      P_INIT_MSG_LIST        IN     VARCHAR2  := FND_API.G_FALSE,
      P_COMMIT               IN     VARCHAR2  := FND_API.G_FALSE,
      P_RESP_APPL_ID         IN     NUMBER    := NULL, -- not used
      P_RESP_ID              IN     NUMBER    := NULL, -- not used
      P_USER_ID              IN     NUMBER    := NULL,
      P_LOGIN_ID             IN     NUMBER    := FND_API.G_MISS_NUM,
      P_ORG_ID               IN     NUMBER    := NULL, -- not used
      P_LINK_ID              IN     NUMBER    := NULL, -- new for 1159
      P_SUBJECT_ID           IN     NUMBER    := NULL, -- new for 1159
      P_SUBJECT_TYPE         IN     VARCHAR2  := NULL, -- new for 1159
      P_OBJECT_ID            IN     NUMBER    := NULL, -- new for 1159
      P_OBJECT_NUMBER        IN     VARCHAR2  := NULL, -- new for 1159
      P_OBJECT_TYPE          IN     VARCHAR2  := NULL, -- new for 1159
      P_LINK_TYPE_ID         IN     NUMBER    := NULL, -- new for 1159
      P_LINK_TYPE            IN     VARCHAR2  := NULL,
      P_REQUEST_ID           IN     NUMBER    := NULL, -- new for 1159
      P_PROGRAM_APPLICATION_ID  IN  NUMBER    := NULL, -- new for 1159
      P_PROGRAM_ID           IN     NUMBER    := NULL, -- new for 1159
      P_PROGRAM_UPDATE_DATE  IN     DATE      := NULL, -- not used
      P_FROM_INCIDENT_ID     IN     NUMBER    := NULL, -- not used
      P_FROM_INCIDENT_NUMBER  IN    VARCHAR2  := NULL, -- not used
      P_TO_INCIDENT_ID        IN    NUMBER   := NULL, -- not used
      P_TO_INCIDENT_NUMBER    IN    VARCHAR2 := NULL, -- not used
      P_LINK_SEGMENT1         IN    VARCHAR2  := FND_API.G_MISS_CHAR,
```

```
P_LINK_SEGMENT2          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT3          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT4          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT5          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT6          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT7          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT8          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT9          IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT10         IN    VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT11         IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT12         IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT13         IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT14         IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT15         IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_CONTEXT           IN    VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
X_RETURN_STATUS          OUT NOCOPY   VARCHAR2,
X_MSG_COUNT              OUT NOCOPY   NUMBER,
X_MSG_DATA               OUT NOCOPY   VARCHAR2,
X_RECIPROCAL_LINK_ID     OUT NOCOPY   NUMBER, -- new for 1159
X_OBJECT_VERSION_NUMBER OUT NOCOPY   NUMBER, -- new for 1159
X_LINK_ID                OUT NOCOPY   NUMBER );
```

### D.2.2.1 Parameter Descriptions

This sections describes the parameters for the Create_Incidentlink procedure.

### Create_Incidentlink In Parameters

The following table describes the in parameters associated with the Create_ Incidentlink procedure:

| Parameter | Data Type | Required | Description |
|---|---|---|---|
| p_api_version | NUMBER | Yes | Version Number |
| p_init_msg_list | VARCHAR2 | Optional | Initialize message stack |
| p_commit | VARCHAR2 | Optional | Commit work |
| p_validation_ level | NUMBER | Optional | The level of Validation |
| p_resp_appl_id | NUMBER | Optional | Responsibility Application Identifier (ID) |
| p_resp_id | NUMBER | Optional | Responsibility ID |
| p_user_id | NUMBER | Optional | User ID |

| | | | |
|---|---|---|---|
| p_login_id | NUMBER | Optional | Login ID |
| p_org_id | NUMBER | Optional | Org ID in which the SR is created |
| p_link_id | NUMBER | Optional | p_link_id |
| p_subject_id | NUMBER | Optional | p_subject_id |
| p_subject_type | VARCHAR2 | Optional | p_subject_type |
| p_from_ incident_id | NUMBER | Yes | The service request ID from where the link is created. This parameter is obsolete. |
| p_from_ incident_number | VARCHAR2 | Yes | The service request number from where the link is created. This parameter is obsolete. |
| p_to_incident_id | NUMBER | Yes | The service request ID to which the link is created. This parameter is obsolete. |
| p_ to_incident_ number | VARCHAR2 | Yes | The service request number to which the link is created.This parameter is obsolete. |
| p_link_type_id | NUMBER | Yes | The Link type ID. |
| p_link_type | VARCHAR2 | Yes | The Link type. This parameter is obsolete. |
| p_link_segment1 | VARCHAR2 | Optional | Flex Field 1 |
| p_link_segment2 | VARCHAR2 | Optional | Flex Field 2 |
| p_link_segment3 | VARCHAR2 | Optional | Flex Field 3 |
| p_link_segment4 | VARCHAR2 | Optional | Flex Field 4 |
| p_link_segment5 | VARCHAR2 | Optional | Flex Field 5 |
| p_link_segment6 | VARCHAR2 | Optional | Flex Field 6 |
| p_link_segment7 | VARCHAR2 | Optional | Flex Field 7 |
| p_link_segment8 | VARCHAR2 | Optional | Flex Field 8 |
| p_link_segment9 | VARCHAR2 | Optional | Flex Field 9 |
| p_link_ segment10 | VARCHAR2 | Optional | Flex Field 10 |
| p_link_ segment11 | VARCHAR2 | Optional | Flex Field 11 |

| | | | |
|---|---|---|---|
| p_link_ segment12 | VARCHAR2 | Optional | Flex Field 12 |
| p_link_ segment13 | VARCHAR2 | Optional | Flex Field 13 |
| p_link_ segment14 | VARCHAR2 | Optional | Flex Field 14 |
| p_link_ segment15 | VARCHAR2 | Optional | Flex Field 15 |
| p_link_context | VARCHAR2 | Yes | The context for flex fields |
| p_request_id | NUMBER | Optional | Request ID of concurrent program creating this link |
| p_program_ application_id | NUMBER | Optional | Application ID of concurrent program |
| p_program_id | NUMBER | Optional | Concurrent program ID |
| p_program_ update_date | DATE | Optional | Program update date |
| p _object_id | NUMBER | Optional | The unique internal ID of the Object of the relationship. |
| P_object_number | Number | Optional | User key of the object |
| p _object_type | VARCHAR2 | Optional | The code that identifies the object of the relationship. This must be a valid code in JTF_OBJECTS. |

### Create_incidentlink Out Parameters

The following table describes the out parameters for the Create_incidentlink procedure:

| Parameter | Data Type | Descriptions |
|---|---|---|
| x_return_status | VARCHAR2 | Return status of the API |
| x_msg_count | NUMBER | Message Count |
| x_msg_data | VARCHAR2 | Message Data |
| x_reciprocal_ link_id | NUMBER | Reciprocal link ID |
| x_object_ version_number | NUMBER | Object version number |

| | | |
|---|---|---|
| x_link_id | NUMBER | The unique ID that represents each link. |

## D.2.3 DELETE_INCIDENTLINK Procedure Specifications

```
PROCEDURE DELETE_INCIDENTLINK (
   P_API_VERSION              IN     NUMBER,
   P_INIT_MSG_LIST            IN     VARCHAR2  := FND_API.G_FALSE,
   P_COMMIT                   IN     VARCHAR2  := FND_API.G_FALSE,
   P_RESP_APPL_ID             IN     NUMBER    := NULL, -- not used
   P_RESP_ID                  IN     NUMBER    := NULL, -- not used
   P_USER_ID                  IN     NUMBER    := NULL, -- not used
   P_LOGIN_ID                 IN     NUMBER    := FND_API.G_MISS_NUM,
   P_ORG_ID                   IN     NUMBER    := NULL, -- not used
   P_LINK_ID                  IN     NUMBER,
   X_RETURN_STATUS           OUT NOCOPY   VARCHAR2,
   X_MSG_COUNT               OUT NOCOPY   NUMBER,
   X_MSG_DATA                OUT NOCOPY   VARCHAR2 );
```

### D.2.3.1 Parameter Descriptions

This sections describes the parameters for the Delete_Incidentlink procedure.

#### Delete_incidentlink In Parameters

The following table describes the in parameters associated with delete_incidentlink procedure:

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| P_api_version | NUMBER | Yes | API version number |
| P_init_msg_list | VARCHAR2 | Optional | Initialize message stack |
| P_commit | VARCHAR2 | Optional | Commit work |
| P_resp_appl_id | NUMBER | Optional | Responsibility application ID |
| P_resp_id | NUMBER | Optional | Responsibility ID |
| P_user_id | NUMBER | Optional | User ID |
| P_login_id | NUMBER | Optional | Login ID |
| P_org_id | NUMBER | Optional | Organization ID where the service request is created |
| P_link_id | NUMBER | Optional | Link ID |

### Delete_Incidentlink Out Parameters

The following table describes the out parameters for the Delete_incidentlink procedure:

| Parameter | Data Type | Description |
|---|---|---|
| x_return_status | VARCHAR2 | Return status of the API |
| x_msg_count | NUMBER | Message Count |
| x_msg_data | VARCHAR2 | Message Data |

## D.2.4 UPDATE_INCIDENTLINK Procedure Specifications

```
PROCEDURE UPDATE_INCIDENTLINK (
    P_API_VERSION             IN NUMBER,
    P_INIT_MSG_LIST           IN VARCHAR2  := FND_API.G_FALSE,
    P_COMMIT                  IN VARCHAR2  := FND_API.G_FALSE,
    P_RESP_APPL_ID            IN NUMBER    := NULL,  -- not used
    P_RESP_ID                 IN NUMBER    := NULL,  -- not used
    P_USER_ID                 IN NUMBER    := NULL,
    P_LOGIN_ID                IN NUMBER    := FND_API.G_MISS_NUM,
    P_ORG_ID                  IN NUMBER    := NULL,  -- not used
    P_LINK_ID                 IN NUMBER,
    P_OBJECT_VERSION_NUMBER   IN NUMBER,          -- new for 1159
    P_SUBJECT_ID              IN NUMBER    := NULL,  -- new for 1159
    P_SUBJECT_TYPE            IN VARCHAR2  := NULL,  -- new for 1159
    P_LINK_TYPE_ID            IN NUMBER    := NULL,  -- new for 1159
    P_LINK_TYPE               IN VARCHAR2  := NULL,  -- no change
    P_OBJECT_ID               IN NUMBER    := NULL,  -- new for 1159
    P_OBJECT_NUMBER           IN VARCHAR2  := NULL,  -- new for 1159
    P_OBJECT_TYPE             IN VARCHAR2  := NULL,  -- new for 1159
    P_REQUEST_ID              IN NUMBER    := NULL,  -- new for 1159
    P_PROGRAM_APPLICATION_ID  IN NUMBER    := NULL,  -- new for 1159
    P_PROGRAM_ID              IN NUMBER    := NULL,  -- new for 1159
    P_PROGRAM_UPDATE_DATE     IN DATE      := NULL,  -- new for 1159
    P_FROM_INCIDENT_ID        IN NUMBER    := NULL,  -- not used
    P_FROM_INCIDENT_NUMBER    IN VARCHAR2  := NULL,  -- not used
    P_TO_INCIDENT_ID          IN NUMBER    := NULL,  -- not used
    P_TO_INCIDENT_NUMBER      IN VARCHAR2  := NULL,  -- not used
    P_LINK_SEGMENT1           IN VARCHAR2  := FND_API.G_MISS_CHAR,
    P_LINK_SEGMENT2           IN VARCHAR2  := FND_API.G_MISS_CHAR,
    P_LINK_SEGMENT3           IN VARCHAR2  := FND_API.G_MISS_CHAR,
    P_LINK_SEGMENT4           IN VARCHAR2  := FND_API.G_MISS_CHAR,
    P_LINK_SEGMENT5           IN VARCHAR2  := FND_API.G_MISS_CHAR,
```

```
P_LINK_SEGMENT6           IN VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT7           IN VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT8           IN VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT9           IN VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT10          IN VARCHAR2  := FND_API.G_MISS_CHAR,
P_LINK_SEGMENT11          IN VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT12          IN VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT13          IN VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT14          IN VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_SEGMENT15          IN VARCHAR2  := FND_API.G_MISS_CHAR, -- new for 1159
P_LINK_CONTEXT            IN VARCHAR2  := FND_API.G_MISS_CHAR,
X_RETURN_STATUS           OUT NOCOPY   VARCHAR2,
X_OBJECT_VERSION_NUMBER   OUT NOCOPY   NUMBER, -- new for 1159
X_MSG_COUNT               OUT NOCOPY   NUMBER,
X_MSG_DATA                OUT NOCOPY   VARCHAR2  );
```

### D.2.4.1 Parameter Descriptions

This sections describes the parameters for the Update_Incidentlink procedure.

### Update_Incidentlink In Parameters

The following table describes the in parameters associated with Update_ Incidentlink procedure:

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| P_api_version | NUMBER | Yes | Version Number |
| P_init_msg_list | VARCHAR2 | Optional | Initialize message stack |
| P_commit | VARCHAR2 | Optional | Commit work |
| P_resp_appl_id | NUMBER | Optional | Responsibility Application ID |
| P_resp_id | NUMBER | Optional | Responsibility ID |
| P_user_id | NUMBER | Optional | User ID |
| P_login_id | NUMBER | Optional | Login ID |
| P_org_id | NUMBER | Optional | Org ID in which the SR is created |
| P_link_id | NUMBER | Optional | Link ID |
| P_object_version_ number | NUMBER | Yes | Object version number |

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| P_subject_id | NUMBER | Optional | ID of the subject of the relationship |
| P_subject_type | VARCHAR2 | Optional | Code that identifies the subject of the relationship. This must be a valid code in JTF_OBJECTS. |
| P_from_incident_id | NUMBER | Yes | The service request ID from where the link is created. This parameter is not used anymore. |
| P_from_incident_ number | VARCHAR2 | Yes | The service request number from where the link is created. This parameter is not used anymore |
| p_to_incident_id | NUMBER | Yes | The service request ID to which the link is created. This is not used anymore |
| p_ to_incident_number | VARCHAR2 | Yes | The service request number to which the link is created. This is not used anymore |
| p_link_type_id | NUMBER | Yes | Link type ID |
| p_link_type | VARCHAR2 | Yes | The Link type - Not used anymore. |
| p_link_segment1 | VARCHAR2 | Optional | Flex Field 1 |
| p_link_segment2 | VARCHAR2 | Optional | Flex Field 2 |
| p_link_segment3 | VARCHAR2 | Optional | Flex Field 3 |
| p_link_segment4 | VARCHAR2 | Optional | Flex Field 4 |
| p_link_segment5 | VARCHAR2 | Optional | Flex Field 5 |
| p_link_segment6 | VARCHAR2 | Optional | Flex Field 6 |
| p_link_segment7 | VARCHAR2 | Optional | Flex Field 7 |
| p_link_segment8 | VARCHAR2 | Optional | Flex Field 8 |
| p_link_segment9 | VARCHAR2 | Optional | Flex Field 9 |
| p_link_segment10 | VARCHAR2 | Optional | Flex Field 10 |
| p_link_segment11 | VARCHAR2 | Optional | Flex Field 11 |

| Parameter | Data Type | Required? | Description |
|---|---|---|---|
| p_link_segment12 | VARCHAR2 | Optional | Flex Field 12 |
| p_link_segment13 | VARCHAR2 | Optional | Flex Field 13 |
| p_link_segment14 | VARCHAR2 | Optional | Flex Field 14 |
| p_link_segment15 | VARCHAR2 | Optional | Flex Field 15 |
| p_link_context | VARCHAR2 | Yes | The context for flex fields |
| p_request_id | NUMBER | Optional | Request ID of concurrent program creating this link |
| p_program_application_id | NUMBER | Optional | Application ID of concurrent program |
| p_program_id | NUMBER | Optional | Concurrent program ID |
| p_program_update_date | DATE | Optional | Program update date |
| p _object_id | NUMBER | Optional | The unique internal ID of the Object of the relationship. |
| P_object_number | NUMBER | Optional | User key of the object |
| p _object_type | VARCHAR2 | Optional | The code that identifies the object of the relationship. This must be a valid code in JTF_OBJECTS. |

### Update_incidentlink Out Parameters

The following table describes the out parameters associated for the Update_ incidentlink procedure:

| Parameter | Data Type | Description |
|---|---|---|
| x_return_status | VARCHAR2 | Return status of the API |
| x_msg_count | NUMBER | Message Count |
| x_msg_data | VARCHAR2 | Message Data |
| x_object_version_number | NUMBER | Object version number |

## D.3 Data Structure Specifications

The following data structures are used in CS_SERVICEREQUEST_PUB:

- NOTES_REC

  The Notes record type holds the attributes for the Notes table.

- CONTACTS_REC

  The Contacts record type holds the attributes for the Contacts table.

- SERVICE_REQUEST_REC_TYPE

  Service Request record type holds the attributes for service request creation.

## D.3.1 Notes_Rec

The Notes record type holds the attributes for the Notes Table.

This is used to create records in the JTF_NOTES table. Notes are created using the JTF_NOTES API.

### Record Specification

```
TYPE notes_rec IS RECORD (
    NOTE                     VARCHAR2(2000)  := FND_API.G_MISS_CHAR,
    NOTE_DETAIL              VARCHAR2(32767) := FND_API.G_MISS_CHAR,
    NOTE_TYPE                VARCHAR2(240)   := FND_API.G_MISS_CHAR,
    NOTE_CONTEXT_TYPE_01     VARCHAR2(30)    := FND_API.G_MISS_CHAR,
    NOTE_CONTEXT_TYPE_ID_01  NUMBER          := FND_API.G_MISS_NUM,
    NOTE_CONTEXT_TYPE_02     VARCHAR2(30)    := FND_API.G_MISS_CHAR,
    NOTE_CONTEXT_TYPE_ID_02  NUMBER          := FND_API.G_MISS_NUM,
    NOTE_CONTEXT_TYPE_03     VARCHAR2(30)    := FND_API.G_MISS_CHAR,
    NOTE_CONTEXT_TYPE_ID_03  NUMBER          := FND_API.G_MISS_NUM
);
```

## D.3.2 Contacts_Rec

The Contacts record type holds the attributes for the Contacts table. The Service Request API creates contacts in CS_HZ_SR_CONTACT_POINTS.

### Record Specification

```
TYPE contacts_rec IS RECORD
(
   SR_CONTACT_POINT_ID       NUMBER         := FND_API.G_MISS_NUM,
   PARTY_ID                  NUMBER         := FND_API.G_MISS_NUM,
   CONTACT_POINT_ID          NUMBER         := FND_API.G_MISS_NUM,
   CONTACT_POINT_TYPE        VARCHAR2(30)   := FND_API.G_MISS_CHAR,
   PRIMARY_FLAG              VARCHAR2(1)    := FND_API.G_MISS_CHAR,
```

```
   CONTACT_TYPE              VARCHAR2(30)   :=FND_API.G_MISS_CHAR
);
```

## D.3.3 Service_request_rec_type

The Service Request record type holds the attributes for service request creation.

Within this record type, the following input parameters are mandatory:

- request_date
- type_id
- type_name
- status_id
- status_name
- severity_id
- severity_name
- urgency_id
- urgency_name
- summary
- caller_type
- customer_id
- customer_number

### Record Specification

```
TYPE service_request_rec_type IS RECORD (
     request_date               DATE,
     type_id                    NUMBER,
     type_name                  VARCHAR2(30),
     status_id                  NUMBER,
     status_name                VARCHAR2(30),
     severity_id                NUMBER,
     severity_name              VARCHAR2(30),
     urgency_id                 NUMBER,
     urgency_name               VARCHAR2(30),
     closed_date                DATE,
     owner_id                   NUMBER,
     owner_group_id             NUMBER,
```

```
publish_flag                VARCHAR2(1),
summary                     VARCHAR2(240),
caller_type                 VARCHAR2(30),
customer_id                 NUMBER,
customer_number             VARCHAR2(30),
employee_id                 NUMBER,
employee_number             VARCHAR2(30),
verify_cp_flag              VARCHAR2(1),
customer_product_id         NUMBER,
platform_id                 NUMBER,
platform_version            VARCHAR2(250),
db_version                  VARCHAR2(250),
platform_version_id         NUMBER,
cp_component_id             NUMBER,
cp_component_version_id     NUMBER,
cp_subcomponent_id          NUMBER,
cp_subcomponent_version_id NUMBER,
language_id                 NUMBER,
language                    VARCHAR2(4),
cp_ref_number               NUMBER,
inventory_item_id           NUMBER,
inventory_item_conc_segs    VARCHAR2(800),
inventory_item_segment1     VARCHAR2(200),
inventory_item_segment2     VARCHAR2(200),
inventory_item_segment3     VARCHAR2(200),
inventory_item_segment4     VARCHAR2(200),
inventory_item_segment5     VARCHAR2(200),
inventory_item_segment6     VARCHAR2(200),
inventory_item_segment7     VARCHAR2(200),
inventory_item_segment8     VARCHAR2(200),
inventory_item_segment9     VARCHAR2(200),
inventory_item_segment10    VARCHAR2(200),
inventory_item_segment11    VARCHAR2(200),
inventory_item_segment12    VARCHAR2(200),
inventory_item_segment13    VARCHAR2(200),
inventory_item_segment14    VARCHAR2(200),
inventory_item_segment15    VARCHAR2(200),
inventory_item_segment16    VARCHAR2(200),
inventory_item_segment17    VARCHAR2(200),
inventory_item_segment18    VARCHAR2(200),
inventory_item_segment19    VARCHAR2(200),
inventory_item_segment20    VARCHAR2(200),
inventory_item_vals_or_ids VARCHAR2(1),
inventory_org_id            NUMBER,
current_serial_number       VARCHAR2(30),
```

```
original_order_number      NUMBER,
purchase_order_num         VARCHAR2(50),
problem_code               VARCHAR2(50),
exp_resolution_date        DATE,
install_site_use_id        NUMBER,
request_attribute_1        VARCHAR2(150),
request_attribute_2        VARCHAR2(150),
request_attribute_3        VARCHAR2(150),
request_attribute_4        VARCHAR2(150),
request_attribute_5        VARCHAR2(150),
request_attribute_6        VARCHAR2(150),
request_attribute_7        VARCHAR2(150),
request_attribute_8        VARCHAR2(150),
request_attribute_9        VARCHAR2(150),
request_attribute_10       VARCHAR2(150),
request_attribute_11       VARCHAR2(150),
request_attribute_12       VARCHAR2(150),
request_attribute_13       VARCHAR2(150),
request_attribute_14       VARCHAR2(150),
request_attribute_15       VARCHAR2(150),
request_context            VARCHAR2(30),
external_attribute_1       VARCHAR2(150),
external_attribute_2       VARCHAR2(150),
external_attribute_3       VARCHAR2(150),
external_attribute_4       VARCHAR2(150),
external_attribute_5       VARCHAR2(150),
external_attribute_6       VARCHAR2(150),
external_attribute_7       VARCHAR2(150),
external_attribute_8       VARCHAR2(150),
external_attribute_9       VARCHAR2(150),
external_attribute_10      VARCHAR2(150),
external_attribute_11      VARCHAR2(150),
external_attribute_12      VARCHAR2(150),
external_attribute_13      VARCHAR2(150),
external_attribute_14      VARCHAR2(150),
external_attribute_15      VARCHAR2(150),
external_context           VARCHAR2(30),
bill_to_site_use_id        NUMBER,
bill_to_contact_id         NUMBER,
ship_to_site_use_id        NUMBER,
ship_to_contact_id         NUMBER,
resolution_code            VARCHAR2(50),
act_resolution_date        DATE,
public_comment_flag        VARCHAR2(1),
parent_interaction_id      NUMBER,
```

```
contract_service_id        NUMBER,
contract_service_number    VARCHAR2(150),
contract_id                NUMBER,
project_number             VARCHAR2(120),
qa_collection_plan_id      NUMBER,
account_id                 NUMBER,
resource_type              VARCHAR2(30),
resource_subtype_id        NUMBER,
cust_po_number             VARCHAR2(50),
cust_ticket_number         VARCHAR2(50),
sr_creation_channel        VARCHAR2(50),
obligation_date            DATE,
time_zone_id               NUMBER,
time_difference            NUMBER,
site_id                    NUMBER,
customer_site_id           NUMBER,
territory_id               NUMBER,
initialize_flag            VARCHAR2(1),
cp_revision_id             NUMBER,
inv_item_revision          VARCHAR2(240),
inv_component_id           NUMBER,
inv_component_version      VARCHAR2(90),
inv_subcomponent_id        NUMBER,
inv_subcomponent_version   VARCHAR2(90),
tier                       VARCHAR2(250),
tier_version               VARCHAR2(250),
operating_system           VARCHAR2(250),
operating_system_version   VARCHAR2(250),
database                   VARCHAR2(250),
cust_pref_lang_id          NUMBER,
category_id                NUMBER,
group_type                 VARCHAR2(30),
group_territory_id         NUMBER,
inv_platform_org_id        NUMBER,
component_version          VARCHAR2(3),
subcomponent_version       VARCHAR2(3),
product_revision           VARCHAR2(240),
comm_pref_code             VARCHAR2(30),
cust_pref_lang_code        VARCHAR2(4),
last_update_channel        VARCHAR2(30),
category_set_id            NUMBER,
external_reference         VARCHAR2(30),
system_id                  NUMBER,
error_code                 VARCHAR2(250),
incident_occurred_date     DATE,
```

```
incident_resolved_date     DATE,
inc_responded_by_date      DATE,
resolution_summary         VARCHAR2(250),
incident_location_id       NUMBER,
incident_address           VARCHAR2(960),
incident_city              VARCHAR2(60),
incident_state             VARCHAR2(60),
incident_country           VARCHAR2(60),
incident_province          VARCHAR2(60),
incident_postal_code       VARCHAR2(60),
incident_county            VARCHAR2(60),
owner                      VARCHAR2(360),
group_owner                VARCHAR2(60),
cc_number                  VARCHAR2(48),
cc_expiration_date         DATE,
cc_type_code               VARCHAR(30),
cc_first_name              VARCHAR(250),
cc_last_name               VARCHAR(250),
cc_middle_name             VARCHAR(250),
cc_id                      NUMBER  ,
bill_to_account_id         NUMBER,
ship_to_account_id         NUMBER,
customer_phone_id          NUMBER,
customer_email_id          NUMBER,
creation_program_code      VARCHAR2(30),
last_update_program_code   VARCHAR2(30),
bill_to_party_id           NUMBER,
ship_to_party_id           NUMBER,
program_id                 NUMBER,
program_application_id     NUMBER,
conc_request_id            NUMBER,
program_login_id           NUMBER,
bill_to_site_id            NUMBER,
ship_to_site_id            NUMBER,
incident_point_of_interest        VARCHAR2(240) ,
incident_cross_street             VARCHAR2(240) ,
incident_direction_qualifier      VARCHAR2(30),
incident_distance_qualifier       VARCHAR2(240) ,
incident_distance_qual_uom        VARCHAR2(30),
incident_address2                 VARCHAR2(240) ,
incident_address3                 VARCHAR2(240),
incident_address4                 VARCHAR2(240) ,
incident_address_style            VARCHAR2(30),
incident_addr_lines_phonetic      VARCHAR2(560) ,
incident_po_box_number            VARCHAR2(50) ,
```

```
incident_house_number             VARCHAR2(50),
incident_street_suffix            VARCHAR2(50) ,
incident_street                   VARCHAR2(150),
incident_street_number            VARCHAR2(50) ,
incident_floor                    VARCHAR2(50) ,
incident_suite                    VARCHAR2(50) ,
incident_postal_plus4_code        VARCHAR2(30) ,
incident_position                 VARCHAR2(50) ,
incident_location_directions      VARCHAR2(640),
incident_location_description     VARCHAR2(2000) ,
install_site_id                   NUMBER
);
```

# D.4 Oracle Workflow Processes and Business Events

This topic covers the integration of service request APIs with Oracle Workflow and business events.

## D.4.1 Oracle Workflow Processes

The service request APIs can launch workflow processes when service requests are created and updated. You can enable this functionality by mapping workflow processes to service request types and by setting up rules to launch the workflow processes automatically. See Setting Up Service Request Types on page 1-53 for details.

Customers can modify the seeded workflows or create new custom workflow processes.

## D.4.2 Business Events

The service request and service request relationships APIs raise business events. Customers can write subscriptions to these events and have them automatically invoked when any of these events are raised.

Service request APIs raise the following business events:

- oracle.apps.cs.sr.ServiceRequest.created
- oracle.apps.cs.sr.ServiceRequest.updated
- oracle.apps.cs.sr.ServiceRequest.statuschanged
- oracle.apps.cs.sr.ServiceRequest.reassigned

- oracle.apps.cs.sr.ServiceRequest.newcontactadded

The service request relationships APIs raise the following business events:

- oracle.apps.cs.sr.ServiceRequest.relationshipcreated
- oracle.apps.cs.sr.ServiceRequest.relationshipdeleted

# D.5 Error and Warning Messages

This topic discusses error and warning messages.

## D.5.1 ServiceRequest API Public Package Messages

The following table describes some of the common error messages and notifications generated by the Service Request package.

| Number | Type | Name | Text |
|---|---|---|---|
| 1388 | - | PROFILES_CANNOT_READ | Cannot read value for profile option &OPTION in routine &ROUTINE.<br>Possible value &OPTION:<br>■ CS_ID_FLEX_CODE<br>Possible values for &ROUTINE:<br>■ G_PKG_NAME.Create_ServiceRequest<br>■ G_PKG_NAME.Update_ServiceRequest<br>■ G_PKG_NAME.Update_Status<br>■ G_PKG_NAME.Update_Serverity<br>■ G_PKG_NAME.Update_Urgency<br>■ G_PKG_NAME.Update_Owner<br>■ G_PKG_NAME.Update_Problem_Code |

| Number | Type | Name | Text |
|---|---|---|---|
| 0 | - | CS_API_ALL_MISSING_ ORG_ID | API Programming Error (&API_NAME): The parameter ORG_ID is required in a multi-org environment.<br>Possible values for &API_NAME:<br>■ Create_ServiceRequest<br>■ Update_ServiceRequest<br>■ Update_Status<br>■ Update_Severity<br>■ Update_Urgency<br>■ Update_Owner |
| 0 | - | CS_API_ALL_ORG_ID_ IGNORED | API Programming Warning (&API_ NAME): The parameter ORG_ID cannot be set because Multi-Org is not enabled.<br>Possible values for &API_NAME:<br>■ CS_ServiceRequest_PUB.Create_ ServiceRequest<br>■ CS_ServiceRequest_PUB.Update_ ServiceRequest |

| Number | Type | Name | Text |
|---|---|---|---|
| 0 | - | CS_API_ALL_VALUE_ TRUNCATED | API Programming Warning (&API_ NAME): The parameter &TRUNCATED_ PARAM was truncated because the character value (&VAL_LEN) is longer than the defined width of the VARCHAR2 column (&DB_LEN).<br>Possible Values for &TRUNCATED_ PARAM:<br>■ p_status<br>■ p_severity<br>■ p_urgency<br>■ p_type_name<br>■ p_status_name<br>■ p_severity_name<br>■ p_urgency_name<br>■ p_employee_number<br>■ p_cp_ref_number<br>■ p_request_number<br>&VAL_LEN is the length of the input string and &DB_LEN is the defined length of that string. |

The following table lists some of the messages generated from the service request private package which is called from the public package:

| Type | Message Name | Text |
|---|---|---|
| ERROR | CS_SR_INVALID_AUTOASSIGN_ PARAM | The value of auto_assign parameter is not 'Y' or 'N', we assume that caller does not want to run auto assignment. |
| - | CS_SR_DESTINATION_UPDATED | Service Request record is of higher version and cannot be updated. |
| - | CS_SR_LESSER_OBJ_VERSION | The version number of Service Request is lesser than the object version number passed. |

| Type | Message Name | Text |
|---|---|---|
| - | CS_API_SR_OWNER_READONLY | API Programming Error (&API_NAME): The owner of the service request cannot be updated because there is an active workflow |
| - | CS_API_SR_TYPE_READONLY | API Programming Error (&API_NAME): The service request type cannot be updated because there is an active workflow. |
| ERROR | CS_API_SR_PRODUCT_NOT_UPDATED | API Programming Error (&API_NAME): The Product and Product related fields have not been updated for this Service Request |
| - | CS_SR_PARENT_CHILD_CHECK | Cause: &CHILD_PARAM cannot be validated without &PARENT_PARAM.<br>Action: Please provide valid value for &PARENT_PARAM, if &CHILD_PARAM is provided.<br>Possible values for &CHILD_PARAM:<br>■ Bill_to_account_id<br>■ Bill_to_contact_id<br>■ Bill_to_site_id<br>■ Ship_to_account_id<br>■ Ship_to_contact_id<br>■ Ship_to_site_id<br>Possible values for &PARENT_PARAM:<br>■ Bill_to_party_id<br>■ Ship_to_party_id |

## D.5.2 Service Request Relationships Package Messages

The following table lists some of the messages generated from the service request relationships private and utility packages which are called from the public package:

| Type | Message Name | Text |
|---|---|---|
| Error | CS_SR_SAME_SUBJECT_OBJECT | Invalid Relationship. Object and Related Object cannot be the same.API validation error - &API_NAME |
| Error | CS_SR_LINK_NO_VALID_OBJ | Valid objects do not exist for the given Object, Related Object, Link Type combination. Please define valid objects for this combination.API validation error - &API_NAME |
| Error | CS_SR_LINK_NO_OBJ_JTF_TYPE | Object type is not valid. Please specify a valid type from JTF Objects for object type. Also, verify if the type is defined as a Valid object that can be used in Linking. |
| Error | CS_SR_LINK_NO_JTF_SEL_ NAME | Object &OBJECT_CODE is not fully defined in JTF Objects. |
| Error | CS_SR_LINK_NO_SUB_JTF_TYPE | Subject type is not valid. Please specify a valid type from JTF Objects for subject type. Also, verify if the type is defined as a Valid object that can be used in Linking. |
| Error | CS_SR_LINK_INVALID_LINK | Relationship specified is not valid.API validation error - &API_NAME |
| Error | CS_SR_LINK_INVALID_LINK_ CHILD | No record with primary key &SUB_OBJ_ID exists in table &SUB_OBJ_TABLE_ NAME.API validation error - &API_NAME |
| Error | CS_SR_LINK_DUP_LINK | Duplicate relationship. Relationship already exists between specified Object and Related Object.API validation error - &API_NAME |
| Error | CS_SR_LINK_CIRCULAR_LINK | You cannot create a &LINK_TYPE link to &OBJECT_TYPE &OBJECT_NUM because it would result in a circular relationship.API validation error - &API_NAME |
| Error | CS_SR_LINK_CIRCULAR_LINK_ UPD | You cannot update the existing relationship to &LINK_TYPE &OBJECT_TYPE &OBJECT_NUM because it would result in a circular relationship. |
| Error | CS_SR_LINK_ORIGINAL_EXISTS | You cannot add new relationships to duplicate &SUB_OBJ_TYPE &SUB_OBJ_ NUM. Please add any new relationships to the original &REL_OBJ_TYPE &REL_OBJ_ NUM.API validation error - &API_NAME |

| Type | Message Name | Text |
|---|---|---|
| Error | CS_SR_LINK_DUP_ORG_NOT_ALLOW | You cannot create a &LINK_TYPE relationship to &SUB_OBJ_TYPE &SUB_OBJ_NUM because it has an existing &EXISTING_LINK_TYPE relationship. |

# E

# Lookup Codes

This appendix lists the lookup types and codes for the Service Request and Customer Care modules of Oracle TeleService.

## E.1 Lookup Codes for Customer Care

The following table lists lookup codes for Customer Care functionality within Oracle TeleService:

| Lookup Type | Lookup Code | Description |
|---|---|---|
| CSC_CC_DEFAULT_SEARCH_TAB | ■ BASIC_SEARCH<br>■ DVANCED_SEARCH<br>■ SERVICE_KEY_SEARCH | Default Contact Center search window tab. |
| CSC_CC_DEFAULT_SERVICE_KEY | ■ SERIAL_NUMBER<br>■ EXTERNAL_REFERENCE<br>■ SYSTEM_NUMBER | Default service key lookup for Contact Center |

| Lookup Type | Lookup Code | Description |
|---|---|---|
| CSC_CC_DEFAULT_TAB | ■ INTERACTION<br>■ DASHBOARD<br>■ NOTES<br>■ CONTACT_POINTS<br>■ ADDRESS<br>■ QUICK_SERVICE_REQUEST<br>■ TASKS<br>■ RELATIONSHIPS<br>■ ACCOUNTS<br>■ PARTY_INFORMATION<br>■ CUSTOM1<br>■ CUSTOM2 | Default tab in the Contact Center. |
| CSC_CC_VIEW_BY | ■ CONTACT<br>■ CUSTOMER | Values for the View By drop-down list on Contact Center tabs |
| CSC_CRITCUST_OVERRIDE_REASON | ■ CRUCIAL_IMPLEMENTATION<br>■ DELICATE_SITUATION | Reasons to override customer criticality |
| CSC_END_USER_TYPE | ■ CUST<br>■ AGENT | End-user type of the application that needs to integrate with Relationship Plans |
| CSC_PLAN_GROUP | ■ | Lookup used for defining relationship plan groups. |
| CSC_PLAN_STATUS | ■ APPLIED<br>■ DISABLED<br>■ REMOVED<br>■ ENABLED<br>■ MERGED<br>■ TRANSFERRED | Defines the association of a relationship plan to a customer |

| Lookup Type | Lookup Code | Description |
|---|---|---|
| CSC_PROF_ CATEGORIES | ■ ACCOUNTS<br>■ ONTRACTS<br>■ CUSTOMER INTELLIGENCE<br>■ INSTALLED BASE<br>■ MARKETING<br>■ ORDERS<br>■ SALES<br>■ INTERACTIONS<br>■ LEADS<br>■ OPPORTUNITIES<br>■ QUOTES<br>■ INTERESTS<br>■ TASKS<br>■ SERVICE | Categories used to group customer profiles. |
| CSC_PROF_ RATINGS | ■ HIGH<br>■ LOW<br>■ MEDIUM | Rating for customer profile check values. |

## E.2 Lookup Codes for Service Requests

The following table lists lookups for service requests:

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| CS_COLOR_VISUAL_ ATTRIBUTES | - | - | Visual attribute names for setting foreground colors for Service Request |
| - | DARKBLUE | Dark Blue | Dark Blue |
| - | DARKGREEN | Dark Green | Dark Green |
| - | DARKRED | Dark Red | Dark Red |
| - | LIGHTBLUE | Light Blue | Light Blue |
| - | LIGHTGREEN | Light Green | Light Green |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | LIGHTRED | Light Red | Light Red |
| - | LIGHTYELLOW | Light Yellow | Light Yellow |
| - | MEDBLUE | Medium Blue | Medium Blue |
| - | MEDGREEN | Medium Green | Medium Green |
| - | MEDRED | Medium Red | Medium Red |
| - | MEDIUMYELLOW | Medium Yellow | Medium Yellow |
| CS_CONTACT_ CATEGORY | - | - | CS Contact Category |
| - | EMPLOYEE | Employee | - |
| - | PARTY | Party | - |
| - | GROUP | Group | - |
| - | TEAM | Team | - |
| CS_CONTACT_TYPE | - | - | CS Contact Type |
| - | SHIP | Ship To | - |
| - | BILL | Bill To | - |
| - | SERV_ADMIN | Service Administration | - |
| - | TECH | Technical | - |
| CS_CREDIT_CARD_ TYPES | - | - | Credit Card Types |
| - | MASTERCARD | Master Card | Master Card |
| - | VISA | Visa | Visa Card |
| - | AMEX | Amex | American Express |
| - | DINERS | Diners | Diners Club |
| - | DISCOVER | Discover | Discover |
| - | ENROUTE | Enroute | En route |
| - | JCB | JCB | JCB |
| CS_MULTI_ORG_RULE | - | Service Request Multi Org Rule | Service Request Multi Org Rule |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | RULE_INSTALLED_ BASE | Operating Unit of the Installed base product | Operating Unit of the Installed base product |
| - | RULE_PROFILE | Profile Option MO: Operating Unit | Profile Option MO: Operating Unit |
| - | RULE_SR | Operating Unit Stamped on the Service Request | Operating unit stamped on the service request |
| - | RULE_CONTRACT | Operating Unit of the Contract | Operating unit of the contract |
| CS_SR_AUDIT_FIELDS | - | - | Service request audit fields |
| - | ALL | All | All |
| - | BILL_TO_CONTACT_ ID | Bill to Contact | Bill to Contact |
| - | CLOSE_DATE | Close Date | Close Date |
| - | CP_COMPONENT_ID | Component | Component |
| - | CP_COMPONENT_ VERSION_ID | Component Version | Component Version |
| - | CP_REVISION_ID | Revision | Revision |
| - | CP_ SUBCOMPONENT_ID | Sub Component | Subcomponent |
| - | CP_ SUBCOMPONENT_ VERSION_ID | Sub Component Version | Subcomponent version |
| - | CUSTOMER_ PRODUCT_ID | Customer Product | Customer product |
| - | EXPECTED_ RESOLUTION_DATE | Expected Resolution Date | Expected resolution date |
| - | INCIDENT_OWNER_ ID | Owner | Owner |
| - | INCIDENT_SEVERITY_ ID | Severity | Severity |
| - | INCIDENT_STATUS_ ID | Status | Status |
| - | INCIDENT_TYPE_ID | Type | Type |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | INCIDENT_ URGENCY_ID | Urgency | Urgency |
| - | INV_COMPONENT_ID | Inventory Component | Inventory Component |
| - | INV_COMPONENT_ VERSION | Inventory Component Version | Inventory Component Version |
| - | INV_ITEM_REVISION | Inventory Item Revision | Inventory Item Revision |
| - | INV_ SUBCOMPONENT_ID | Inventory Sub Component | Inventory Sub Component |
| - | INV_ SUBCOMPONENT_ VERSION | Inventory Sub Component Version | Inventory Sub Component Version |
| - | OBLIGATION_DATE | Obligation Date | Obligation Date |
| - | PLATFORM_ID | Platform | Platform |
| - | PLATFORM_ VERSION_ID | Platform Version | Platform Version |
| - | RESOURCE_TYPE | Resource Type | Resource Type |
| - | SHIP_TO_CONTACT_ ID | Ship to Contact | Ship to contact |
| - | SITE_ID | Site | Site |
| - | TERRITORY_ID | Territory | Territory |
| - | GROUP_TYPE | Group Type | Group type |
| - | GROUP_ID | Group Name | Group name |
| CS_SR_CALLER_TYPE | - | Service Request Caller Types | Service request caller types |
| | CALLER_EMP | Employee | Employee |
| | ORGANIZATION | Organization | Organization |
| | PERSON | Person | Person |
| CS_SR_CONTACT_ TYPE | - | Valid Contact Types | Valid contact types for creating a service request |
| - | PARTY_ RELATIONSHIP | Relation | Relation |
| - | PERSON | Person | Person |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | EMPLOYEE | Employee | Employee |
| - | EMPLOYEE | Employee | Contact type for the service request is an employee |
| - | PERSON | Person | Contact type for the service request is a person |
| CS_SR_CREATION_ CHANNEL | - | Service Request Creation Channels | Creation channels for a service request |
| - | EMAIL | Email | Create a service request via email |
| - | PHONE | Phone | Create a service request via phone |
| - | WEB | Web | Create a service request via Web |
| - | AGENT | Agent | Create a Service Request by Agent |
| - | AUTOMATIC | SR created without User Intervention | Service request created automatically (For example, via a concurrent program) |
| CS_SR_DEFAULT_TAB | - | Service Request Default Tab | Tab page to be defaulted on service request form |
| - | PRODUCT_ COVERAGE | Product Coverage | Product coverage tab |
| - | WORKBENCH | Workbench | Workbench tab |
| - | LOG | Log | Log tab |
| - | CONTACTS | Contacts | Contacts tab |
| - | ADDRESSES | Addresses | Addresses tab |
| - | TASKS | Tasks | Tasks tab |
| - | INTERACTIONS | Interactions | Interactions tab |
| - | RELATED_ DOCUMENTS | Related Objects | Related Objects tab |
| - | SERVICE_HISTORY | Service History | Service History tab |
| - | CHARGES | Charges | Charges tab |
| CS_SR_DIRECTIONS | - | Directions | Direction Qualifier. |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | N | North | North |
| - | S | South | South |
| - | E | East | East |
| - | W | West | West |
| - | NE | North East | North East |
| - | SW | South West | South West |
| - | NW | North West | North West |
| - | SE | South East | South East |
| CS_SR_DISTANCE_UOM | - | Directions Unit of Measure | Directions Unit of Measure |
| - | KM | Kilometers | Kilometers |
| - | MILE | Miles | Miles |
| - | METER | Meters | Meters |
| - | YARD | Yards | Yards |
| CS_SR_LOG_CHECK_BOX | - | Service: Log tab default checkbox | Service: Log tab default check box |
| - | - | - | - |
| - | ALL | All Records | All Records |
| - | AUDIT | Audit Records | Audit Records |
| - | TASK | Task Records | Task Records |
| - | ACTIVITY | Activity Records | Activity Records |
| - | NOTE | Notes Records | Notes Records |
| - | SOLN | KB Solution Records | KB Solution Records |
| CS_SR_SOURCE_PROGRAMS | - | Service Request Source Program Codes | Service Request Source Program Codes |
| - | SRAMCONC | Service Auto-assignment concurrent program | Service Auto-assignment concurrent program |
| - | CSCCCQSR.SCR | Seeded Script | Script (Seeded Script from SR/CSC) |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | OKS_CT_EVENTS_ WFA | Service Contract Events API | Service Contract Events API |
| - | XNSSRISR | Service for Comms | Service for Comms |
| - | ISUPPORTSRUI | iSupport Service Request UI | iSupport Service Request UI |
| - | CSZSRC | Service Online: Create call from CszIncidentEO | Service Online: Create call from CszIncidentEO |
| - | CSZSRU | Service Online: Update call from CszIncidentEO | Service Online: Update call from CszIncidentEO |
| - | CSM_UPSYNC_ WRAPPER | CSM Upward Synchronization Wrapper Package | CSM upward synchronization wrapper package |
| - | EMAILCENTER | eMail Center | eMail Center |
| - | PMCON | Preventive Maintenance - Concurrent Request | Preventive maintenance - concurrent request |
| - | CSD_REPAIR_ORDER_ FORM | Depot Repair - Repair Order form | Depot Repair - Repair Order form |
| - | AHL_NONROUTINE | Service Request for a Non-Routine job | Service Request for a Non-Routine job |
| - | AHL_ROUTINE | Service Request for a routine job | Service Request for a routine job |
| - | CSL_LAPTOP | Field Service / Laptop | Field Service / Laptop |
| - | CSXSRISR | Teleservice-Service Request Form | Teleservice-Service Request Form |
| - | CSXSRTAB | Teleservice Service Request Tab | Teleservice Service Request Tab |
| - | SUPPORT.WF | Workflow for Customer Support | Workflow for Customer Support |
| - | BIV_SR_UPDATE | Service Intelligence | Service Intelligence |
| - | UNKNOWN | Source Program Code not specified | Source Program Code not specified |
| CS_SR_STATUS_ CONDITIONS | - | Status Change Event Conditions | Status Change Event Conditions |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | STATUS_CHANGED_FROM | Status Changed From | Status Changed From |
| - | STATUS_CHANGED_TO | Status Changed To | Status changed to |
| CS_SR_TIMEZONE_TYPE | - | Timezone type for Service Request | Time zone type for service request |
| - | AGENT_TZ | Agent | Agent time zone |
| - | CONTACT_TZ | Contact | Contact time zone |
| - | SERVER_TZ | Server | Server time zone |
| CS_SR_UPDATED_FIELDS | - | Update Fields of Service Request | Update fields of service request |
| - | CS_SR_SEVERITY | Severity | Severity |
| - | CS_SR_STATUS | Status | Status |
| - | CS_SR_SUMMARY | Summary | Summary |
| - | CS_SR_TYPE | Type | Type |
| - | CS_SR_URGENCY | Urgency | Urgency |
| CS_SR_UWQ_GROUP_LABEL | - | Labels used by UWQ SR extension | Labels used by UWQ SR extension |
| - | MY_GROUP_LABEL | My Groups | My Groups |
| - | GROUP_LABEL | Groups | Groups |
| CS_SR_UWQ_MY_LABEL | - | Service: SR UWQ My label | Service: SR UWQ My label |
| | MY_LABEL | My Service Request | My service request |
| CS_SR_UWQ_TEAM_LABEL | - | Service: SR UWQ Team label | Service: SR UWQ Team label |
| - | MY_TEAM_LABEL | My Teams | My teams |
| - | TEAM_LABEL | Teams | Teams |
| INC_AUDIT_ATTRIBUTES | - | Incident Attributes that have audit history | Incident attributes that have audit history |
| - | TYP | Type | Type of incident |
| - | URG | Urgency | Urgency of incident |

| Lookup Type | Lookup Code | Meaning | Description |
|---|---|---|---|
| - | ALL | All Attributes | All audited attributes of incident |
| - | AG | Assigned Group | Service group assigned to incident |
| - | C | Component | Product component associated with incident |
| - | SE | Service Entity | Service entity associated with support organization |
| - | O | Owner | Owner of incident |
| - | P | Product | Product associated with incident |
| - | SEV | Severity | Severity of incident |
| - | STAT | Status | Status of incident |
| MESSAGE_PRIORITY | | Message Priorities | Message priorities |
| - | LOW | Low | Low priority |
| - | MED | Medium | Medium priority |
| - | HIGH | High | High priority |
| MESSAGE_RESPONSE | - | Message Responses | Message responses |
| - | ACCEPT | Accept | Accept the action request |
| - | REJECT | Reject | Reject the action request |
| - | DONE | Done | Action request completed |
| - | ACKNOWLEDGE | Acknowledge | Message acknowledged |
| REQUEST_PROBLEM_ CODE | - | -Service Request Problem Codes | Service request problem codes for service request form |
| REQUEST_ RESOLUTION_CODE | - | Service Request Resolution Codes | Service request resolution codes |
| - | CS_SR_CLOSED_AS_ DUP | Closed as duplicate | Closed as duplicate |