Oracle® Support

Implementation Guide

Release 11*i*

October 2001

Part No. A95148-01

ORACLE®

Oracle Support Implementation Guide, Release 11*i*

Part No. A95148-01

Contents

2 Implementation Overview

3 Implementation Tasks

A Service Request Public APIs

Send Us Your Comments

Oracle Support Implementation Guide, Release 11*i*

Part No. A95148-01

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 in the following way:

Oracle Corporation
CRM Content Development Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

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 Oracle Support Implementation Guide.

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

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

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

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

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

How To Use This Guide

This document contains the information you need to understand and use ***Oracle Support***.

- Chapter 1 provides a high level description of the Oracle Supports and the key features of Oracle Support. A list of new and obsolete features is also provided.
- Chapter 2 provides a high level description of the implementation process for Oracle Support.

- Chapter 3 discusses detailed implementation steps for setting up Oracle Support.
- Chapter 4 describes the Oracle Support Public 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 Support***.

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

Online Documentation

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

Related Documentation

Oracle Support 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 Support***.

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 Support*** (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

Oracle Support Concepts and Procedures

This guide provides information required for using Oracle Support.

Oracle Customer Care Concepts and Procedures

This guide describes how to use Oracle Customer Care. Oracle Customer Care provides the functionality for using the Contact Center, Relationship Plans and tracking critical customers.

Oracle Customer Care Implementation Guide

This guide provides information required for implementing and setting up Oracle Customer Care.

Oracle Service Implementation Guide

This guide describes how to implement and setup Oracle Service. Oracle Service provides the functionality required for Counters and Charges.

Oracle Service Concepts and Procedures

This guide describes how to use Oracle Service. Oracle Service provides the functionality required for Counters and Charges.

Installation and System Administration

Oracle Applications Concepts

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

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Oracle Applications Supplemental CRM Installation Steps

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

Upgrading Oracle Applications

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

Maintaining Oracle Applications

Use this guide to help you run the various 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

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 Support*** 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 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 Support*** 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 Support*** working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8*i* server, and your hardware and software environment.

Oracle*MetaLink*

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

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

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

Do Not Use Database Tools to Modify Oracle Applications Data

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

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

Because Oracle Applications 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

Introduction to Oracle Support

This chapter provides an overview of the Oracle Service product family and Oracle Support. A summary of new features and obsolete features is also discussed.

The following topics are covered:

- Oracle Support Overview
- Oracle Support Features
- New in this Release
- Obsolete in this Release

1.1 Oracle Support Overview

Oracle Support allows an organization to manage customer requests for service assistance. With Oracle Support, your support organization can equip customer care agents, support analysts, engineers and managers with a knowledge driven application for management of customer information and service requests, defects, and enhancements.

Oracle Support offers an intuitive user interface for creating service requests and provides a seamless integration across the enterprise.

1.2 Oracle Support Features

This section discusses the key features of Oracle Support. The Oracle Support Service Request module helps you to manage customer requests for service in the following ways:

- Record customer information quickly.

- Confirm customer and product information; name, number, product, system, or serial number.
- Alert support agents about critical customers.
- Describe the customer's request for service.
- Define the service request status, type, severity, and urgency.
- Verify service contract entitlements automatically.
- Assign problem and resolution codes to each service request.
- Link service requests to defects, enhancements and related service requests.
- Search a knowledge base to expedite the resolution of service requests.
- Contribute to the knowledge base.
- View interaction history.
- View notes and saved knowledge base solutions for a service request all at once.
- Create multiple tasks for a service request.
- Assign the appropriate resource to perform the task.
- Provide an audit trail of service request changes.
- Send a service request report by email or fax by using Fulfillment.
- Create a defect related to a service request.
- Find existing service requests by using a comprehensive set of search criteria.
- Escalate a service request.
- Update counters.
- Use Oracle Workflow to automate business processes used during service request resolution.
- Use Oracle Quality to collect quality control information linked to service requests.

1.3 New in this Release

The following features of Oracle Support are new in Release 11.5.6:

- Enhanced Service Request find window
- Enhanced Service Request Log tab
- Enhanced Service Request header region
- Enhanced Service Request to Knowledge Management integration
- Enhanced Service Request node structure in the Universal Work Queue
- Entry points for Service Request creation
- New Service History tab
- Ability to print Service Requests
- Integration with the new Installed Base

1.3.1 Enhanced Service Request Find window

Additional fields have been added to the Service Request Find window.

1.3.2 Enhanced Service Request Log tab

The enhanced Service Request Log tab allows users to view a complete chronology of events that took place against a specific service request. This allows individuals who have not worked on a particular service request to have a centralized location for viewing all events, activities, and notes in an organized manner. Items such as Notes, Tasks, Knowledge Base documents, Interactions and Activities can all be displayed in the log tab.

The Log tab provides a centralized location for viewing the history of a service request. It displays a chronological view of all activities, tasks and notes entered for a service request, as well as all changes to the audited fields, and all knowledge notes linked to this service request.

The Log tab now contains check boxes to determine the type of information the user wants to view in the log. The choices are All, Audit, Notes, Knowledge, Activity, Tasks and Latest First.

A button has been added to the Log tab to allow the user to print the service request.

1.3.3 Enhanced Service Request header region

The following enhancements were implemented in the Service Request header region of the Service Requests window:

Clear distinction between Contact Type and Customer Type

There is now a clear distinction in the Service Request header between Contact Type and Customer Type.

Display of Contact, Customer, Subject and Request information

The Service Request header displays information in four regions: Contact, Customer, Subject and Request. The Contact region displays the details about the primary contact for the Service Request. The Customer region displays information about the Customer for which the Service Request is logged. The Subject region displays information about the Product and Contract. The Request region provides details about the Service Request attributes.

1.3.4 Enhanced Service Request to Knowledge Management integration

The Service Request is integrated with the new Knowledge Management HTML interface. The values in the Product and Platform fields are passed to the Knowledge search page. Once a knowledge solution has been found, these statements are linked to the Service Request. Linked statements can be viewed in the Service Request Workbench tab and Log tab. For reference, the solution ID and the version of the knowledge solution are displayed.

Users can "cut and paste" the details of the knowledge solution or statement to make a local copy as notes, and could then make modifications to it.

Users can modify solutions or statements in the new Knowledge Management page.

1.3.5 Enhanced Service Request node structure in the Universal Work Queue

The Service Request node of the Univeral Work Queue is now comprised of three tiers. The first tier, My Service Requests, displays a list of service requests owned by the user. The second tier, My Group's Service Requests, displays a list of service requests that have no owner and are assigned to the user's Group. The third tier, My Team's Service Requests, displays a list of service requests that have no owner or group and are assigned to the user's Team. The service requests are sorted by severity level.

1.3.6 Entry points for Service Request creation

To expedite the service request creation process, fourteen entry points for the service request are available. By using an entry point when creating a service request, data entry is minimized because all related fields are populated

automatically. For example, if a support agent enters the customer's account number, the customer name, address, phone number and primary contact information is defaulted.

The following entry points are available:

- Account
- Contact Email
- Contact Name
- Contact Number
- Contact Phone
- Contract
- Customer Email
- Customer Name
- Customer Number
- Customer Phone
- Employee Number
- Installed Base Serial Number
- Reference
- System

1.3.7 New Service History tab

The Service History tab provides a centralized view of the service requests that have been created for a product or for a product and customer combination. The list of service requests in this tab is based on the product and customer that is entered in the header region of the service request.

The information in the Service History tab allows a user to quickly determine how many time a customer has requested service for a specific product. A user can also easily find out if a large number of cusotmers have requested service for a product, which may indicate that the product is defective.

1.3.8 Ability to Print Service Requests

You can print service requests from the Log tab, the Workbench tab or from the Tools Menu. When you print a service request, you can specify by Note Status which notes you want to include on the printout. Public, Publish and Private are the available Note Status codes.

1.3.9 Integration with New Install Base

You can choose to display the new Install Base HTML page from the Service Request window.

1.4 Obsolete in this Release

No features are obsolete in this release.

2

Implementation Overview

This chapter presents an overview of the implementation steps that you need to perform to set up Oracle Support. The following topics are covered:

2.1 Process Description

Proper planning is essential for the success of any implementation project. Planning, coupled with the use of a structured methodology, such as Oracle Application Implementation Methodology (AIM) would ensure the smooth progress of your implementation project. The following section presents some of the key implementation considerations.

2.2 Implementation Task Sequence

This section indicates the steps you have to complete to set up Oracle Support and integrate it with other applications.

1. Confirm the setup of System Administration.
2. Confirm the setup of Key Flexfields.
3. Confirm the setup of the Calendar, Currency and Sets of Books.

   Refer to *Oracle General Ledger's User's guide* and *Multiple Organizations in Oracle Applications*.
4. Confirm the setup of organizations.

   Refer to *Implementing Oracle HRMS and Oracle Inventory User's Guides*.

5. Confirm the setup of locations.

   Refer to *Implementing Oracle HRMS.*

6. Define Employees.

   Service Requests must be owned by a Resource. A Resource can be an Employee defined in HRMS and imported into CRM Foundation Resources.

   Refer to *Implementing Oracle HRMS* and *Implementing Oracle CRM Foundation.*

7. Define users for Oracle Support.

   Refer to *Oracle Applications System Administrator's Guide.*

8. Confirm the setup of Resources

   Service Requests must be owned by a Resource. A Resource can be an Employee or a non-Employee. Resources are associated with Application Users. Import the resource for Service Requests into CRM Foundation Resources.

   Refer to *Implementing Oracle CRM Foundation*.

9. Confirm Oracle Inventory setup.

   - Define Categories for your products.
   - Define Category Sets for your products.
   - Define Items for your products.
   - Set up Units of Measure.

   One Inventory Item can be identified in a Service Request. Units of Measure are used for Service Requests charges.

   Refer to *Oracle Inventory User's Guide*.

10. Confirm the setup of Bills of Materials.

    Refer to *Oracle Bills of Materials User's Guide.*

11. Confirm the setup of Order Management (Transaction Types, Taxes codes, Pricing) if you are using Charges.

    Refer to *Oracle Order Management User's Guide*.

12. Confirm the setup of Oracle Pricing if you are using Charges.

    Refer to *Oracle Pricing User's Guide.*

13. Confirm the setup of Territory Management

    Service Request ownership and Task ownership and assignment can be achieved with the Assignment Manager using established Territories. It is recommended that you use the Territory Administrator's Responsibility for this Setup.

    Refer to *Implementing Oracle CRM Foundation.*

14. Confirm the setup of Oracle Workflow

    Service Requests can use workflow processes based on Service Request Type.

    Refer to *Oracle Workflow Guide*.

15. Confirm the setup of Counters. Counters are used to track Inventory Items and Contract Lines. Counters are updated from the Service Requests. This step is required only if you are using Counters.

    Refer to *Oracle Service Implementation Guide*.

16. Confirm the setup of Install Base. Service Requests can optionally track Install Base Items instead of Inventory Items. Installed Base items are associated with specific customers.

    Refer to *Oracle Install Base Implementation Guide*.

17. Confirm the setup of Service Contracts and Contracts Core. Service Request entitlement coverage is accomplished using Service Contracts. Contracts Core is required for Service Contracts.

    Refer to *Oracle Contracts for Service Concepts and Procedures*

    Refer to *Oracle Contracts Core Concepts and Procedures*

18. Confirm the setup of Charges. Service Requests can capture charges to be billed to the customer. This step is required only if you are using Charges.

    Refer to *Oracle Service Implementation Guide*.

19. Confirm the setup of Notes. Notes are used to track activity in service requests, track Knowledge Management information, track Escalation activity and track information regarding customer interactions. One Note Type is required for the default type used for service request updates in the Workbench tab.

    Refer to *Oracle CRM Foundation Implementation Guide*.

20. Confirm the setup of Knowledge Management. Service Request users can search, score and contribute to the Knowledge Management system. This step is required only if you are using Knowledge Management.

    Refer to *Oracle iSupport Implementation Guide*.

21. Confirm the setup of Tasks. Service Requests track and manage required work using Tasks. Resources can track and manage non-Service Request work using Tasks.

    Refer to *Oracle CRM Foundation Implementation Guide*.

22. Confirm the setup of Universal Work Queue. Users can view and select all owned Service Requests, Task, and Escalations with the Universal Work Queue. Users can also receive media work (i.e. inbound calls and email) from the Universal Work Queue.

    Refer to *Oracle Universal Work Queue Implementation Guide*.

23. Confirm the setup of Interaction History. You can track and view customer interactions from the service request.

    Refer to *Oracle CRM Foundation Implementation Guide*.

24. Confirm the setup of Assignment Manager. Service Request, Task and Escalations Resource ownership can be identified using the Assignment Manager.

    Refer to *Oracle CRM Foundation Implementation Guide*.

25. Confirm the setup of Escalation Management. Service Requests and Tasks can be escalated using the Escalation Manager. Escalation notifications can be driven by configured business rules.

    Refer to *Oracle CRM Foundation Implementation Guide*.

26. Confirm the setup of Oracle Quality. Service Requests can be used to collect quality information.

    Refer to *Oracle Quality User's Guide.*

27. Confirm the setup of Fulfillment. The Service Request report can be sent by email and fax using Fulfillment.

    Refer to *Oracle CRM Foundation Implementation Guide*.

28. Confirm the setup of Field Service Report. Service Request Task debriefing information can be tracked using the Field Service Report. This step is required only if you want to use Field Service.

Refer to *Oracle Field Service Implementation Guide*.

29. Confirm the setup of Customer Profiles. Customer Profile information can be viewed in the Service Request.

    Refer to *Oracle Customer Care Implementation Guide*.

30. Confirm the setup of Relationship Plans. This step is optional.

    Refer to *Oracle Customer Care Implementation Guide*.

31. Confirm the setup of Oracle Quality Online to track defects and enhancements. Defects and enhancements can be created from the Service Request. This step is required only if Oracle Quality Online is installed.

    Refer to *Oracle iSupport Implementation Guide*.

32. Confirm the setup of Oracle Scripting. This step is required only if Oracle Scripting will be used.

    Refer to *Oracle Scripting Implementation Guide*.

33. Set up Service Request Status codes. Service Request resolution progress can be tracked and rules enforced using Statuses. See Setting Up Service Request Status Codes.

34. Set up Service Request Types. Service Requests Types are used to enforce business rules and start Workflows. See Setting Up Service Request Types.

35. Set up Service Request Severity codes. Service Request Severity codes represent the user's view of the severity. See Setting Up Service Request Severities.

36. Set up Service Request Urgency codes. Service Requests Urgency codes represent the customer's view of the business urgency. See Setting Up Service Request Urgencies. This step is optional.

37. Set up Service Request Problem Codes. The Problem Code is a numeric entity used to easily identify the specific problem described in the Service Request. Each problem code can be linked to a corresponding resolution code to simplify the resolution search process. See Setting Up Service Request Problem Codes. This step is optional.

38. Set up Service Request Resolution Codes. The Resolution Code is a numeric entity used to easily identify the specific resolution described in the Service Request. Each resolution code can be linked to a corresponding problem code to simplify the resolution search process.See Setting Up Service Request Solution Codes. This step is optional.

39. Set up Message Action Codes if you want to use the message functionality provided in Oracle Support. See Setting Up Message Action Codes.
40. Set up linking for nonvalidated documents if you want to link service requests to nonvalidated documents. See Setting Up Linking of Nonvalidated Documents.
41. Set up the display of task types and task priorities if you want these displayed in the Task tab of the Service Request window. See Setting Up Display of Task Type and Task Priorities.
42. Set up the display of Install Base Products if you want this information displayed from the Service Request window. See Setting Up Display of Installed Base Products.
43. Set up an operating unit for charges if you are using Charges. See Setting Up Operating Unit for Charges.
44. Set up HTML applications for Service Requests if you are using Knowledge Management, Quality Online or Install Base. See Setting Up HTML Applications for Service Requests.
45. Set up Service Request Profile Options. See Setting Up Service Request Profile Options.
46. Set up Create Owners and Supervisors. See Creating Owners and Supervisors.

3

Implementation Tasks

This chapter presents a detailed description of the setup and implementation steps required to successfully implement Oracle Support.

- Setting Up Task Types and Priorities for Service Request Usage
- Setting Up Display of Installed Base Products
- Setting Up Operating Unit for Charges
- Setting Up HTML Applications for Service Requests
- Setting Up the Service Request Synchronize Index Concurrent Program
- Creating a Support Site

3.1 Setting Up Service Request Statuses

You define service request statuses to indicate the current state of reported service requests. For example, a customer calls to report a broken switch on his personal computer. When the service request is created, you can set the service request status to Open.

You can also set up status transition rules and associate a rule with a user responsibility. This allows the system administrator to create different business rules for different sets of users by managing the service request transition from one status to another. The status transition rules determine the progress of a service request through the various statuses as defined in the user privilege for each status type. For example, a system administrator can setup a special transition rule to ensure that all service requests get customer approval before closure. The transition rule would be that the users can only change the status of the service request from Open to Working, from Working to Pending Approval and then from Pending Approval to Closed.

Following are examples of service request statuses:

- New
- Open
- Working
- Assigned
- Closed

Prerequisites:

Appropriate user responsibilities must be defined before you can associate a status transition rule with a user responsibility. All statuses must be defined before you can include them in a status transition rule. Status is a mandatory field in the service request.

To set up a service request status:

1. Navigate to the Service Request Statuses window. All existing statuses are displayed.
2. Enter a status name in the Status field. The value in this name field is displayed in the Status list of values in the Service Request workbench tab.
3. If the status name you are creating indicates a closed status, then select the Closed check box. The closed indicator attribute allows you to enable Oracle Workflow processes when a service request is closed.
4. Optionally, enter Start and End dates to control the use of statuses. 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.
5. If a status is seeded, the predefined check box will be checked. The seeded statuses are Open, Closed and Clear. If you do not want to use a seeded status, enter an end date.
6. You can assign colors to each status to assist the support agent with understanding the status quickly. For example, a status color of red may indicate no progress is being made, while a status color of green may indicate that the support agent is waiting for the customer to respond. If you want to use colors for your statuses, then select from the list of colors in the Text Color field.
7. If a descriptive flexfield has been enabled for this form, select the appropriate values.

8. Enter a description of the status. This description appears in the status list of values.
9. Optionally, check the Service Request Status Restriction check boxes as per the business rules of your organization.

   The following check boxes are available:

   - Disallow Request Update
   - Disallow Task Update
   - Disallow Charge Update - When a service request is in this status and has related charges, the charges cannot be updated.
   - Disallow Owner Update
   - Disallow Product Update
   - Disallow Charge - When a service request is in this status, charges cannot be created for the service request.
10. Click **Transitions** to define status transition rules.
11. Select Support from the Application list of values.
12. Enter a name for the rule in the Rule Name field.
13. Select the status from the list of values to which you want to apply this transition rule.
14. In the Next State field, select the status from the list of values to which you want to allow the current status to change.
15. In the Responsibility tab, select the name of the responsibility that will use this status transition rule.
16. Select the rule name and click OK.

    Now this transition rule has been associated with the selected responsibility.
17. Save your changes.

References

- Setting Up Service Request Types

3.2 Setting Up Service Request Types

You define service request types to categorize your service requests. For each service request type, you can set up related service request statuses that correspond with each service request type. Each type can be linked to an Oracle Workflow process that can be automatically launched when the service request is created or updated if the profile option to automatically launch workflow is enabled. Workflow can also be launched manually in the service request window. Service request type is a mandatory field in the service request.

You can also associate a service request type with a business process. Business processes are defined in Installed Base and are used in generating charges for service requests. For additional information about setting up Oracle Charges, refer to *Oracle Service Implementation Guide.*

Following are some examples of service request types:

- Request for Information
- Customer Complaint
- Installation Request
- Preventive Maintenance Visit
- Helpdesk Request
- Technical Problem

Prerequisites:

- Before selecting the related status for the service request type, you must define the service request status.
- Business processes must be defined if you want to associate a service request type with a business process.
- The workflow process must be defined in Oracle Workflow before you can associate a workflow with a service request type

To set up service request types and their related statuses:

1. Navigate to Service Request Types window.
2. Enter a type name in the Type field. This value is displayed in the Type list of values in the Service Request header.Select a business process from the Business Process list of values. Business processes are used for Charges.

3. 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.
4. If a descriptive flexfield has been enabled for this form, select the appropriate values.
5. Enter a description for the service request type.
6. You can associate a workflow with each service request type by selecting a workflow in the Workflow field. A seeded workflow process is provided that sends a notification to the owner of the service request.
7. Optionally, select the available check boxes
   - Auto Launch Workflow launches workflow automatically when a service request is created and updated.
   - Abort Workflow on Close aborts workflow when a service request is closed.
   - Web Entry is used by Oracle *i*Support.
8. Click Related Statuses to define the statuses that can be used with this service request type. The user sees only this list of statuses in the Status list of values when a service request uses this type.
9. Select a value from the Status list of values. The description of the status is displayed.
10. Optionally, enter the effective dates in the Start Date and End Date fields and click **OK**.
11. Save your service request type.

References:

- Setting Up Service Request Statuses

3.3 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.
2. Enter a severity name in the Severity field. The severity name appears in the Severity list of values of the Service Request Workbench tab.
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.
4. Enter a 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 Defect Severity. This field is enabled only if OQO is installed. Defect Severities are defined in the OQO application.

   When creating defects from a service request, the defect severity field defaults to the defect severity value associated with the service request severity you define in this window.
7. Optionally, select a text color in the Text Color field. The severity will be shown in this color on the service request window.
8. If a descriptive flexfield has been enabled for this form, select the appropriate values.
9. Save your service request severity.

References:

- Setting Service Request Urgency

3.4 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 determine service request urgency for your users:

1. Navigate to the Service Request Urgencies window.
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.

References:

- Setting Service Request Severity

3.5 Setting Up Service Request Problem Codes

You define service request problem codes to isolate the detailed reason for the service request. Problem Code is an optional field in a service request.

Problem codes are implemented in Oracle Support 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.

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 a service request problem code:

1. Navigate to the Service Request Problem Codes window. 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 available for use.
8. Optionally, enter the descriptive flexfield, if defined.
9. Save your problem code

3.6 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 Oracle Support 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 the Service Request Resolution Codes window. 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

References:

none

3.7 Setting Up Message Action Codes

You can use message action codes to specify an action you want a message recipient to take. The messages can be sent from the service request by getting to the messages window. To open this window, Select Messages option from the Tools menu.

To setup a message action code:

1. Navigate to the Message Action Request quick codes Codes window. 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.

3.8 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 Task tab.

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.

3.9 Setting Up Usage of Install Base Products in Support

This step is required to show Install Base products in the Service Request window.

1. Navigate to the Customer Product Statuses setup window.

Setup > Installed Base > Customer Product Status

2. Check the Request Allowed check box for the customer product statuses that you want to be displayed in the Service Request window.

3.10 Setting Up Operating Unit for Charges

This step is required if you are using Charges. In this step, you define the operating unit that will be used for creating charges from a service request. You also assign a sequence to the four seeded multi-org rules. Based on these rules, the operating unit is derived in the calling form. Currently, the calling form is either Charges or Field Service.

Following is a description of the seeded multi-org rules:

- Operating Unit of the Contract -- If there is a contract associated to the Service Request then the operating unit of the contract is used in the calling form.
- Operating Unit of the Installed Base product -- If there is an Install Base item associated to the Service Request then the operating unit of the Install Base item is used in the calling form.
- Profile Option MO: Operating Unit of the Charges -- The operating unit set in this profile is used in the calling form.
- Operating Unit Stamped on the Service Request -- The operating unit associated to the Service Request is used by the calling program.

Based on the sequence associated with these rules , the appropriate operating unit is used in the Charges form and the Field Service Form.

For example, if you assign Sequence 1 to the Operating Unit of the Installed Base product rule and Sequence 2 to the Operating Unit Stamped on the Service Request rule, then the program to derive the operating unit will first check if the rule assigned to Sequence 1 returns an operating unit , and if so, it will return that operating unit to the calling program. If no operating unit is returned, then a check is performed to determine if the rule assigned to sequence 2 returns an operating unit. If none of the rules return an operating unit , then the program will return null.

1. Navigate to the Multi-Org setup window.

   Setup > Service Requests > Multi-Org

2. Assign a sorting order to the Multi Org rules.

3. Set the value in the profile option called Service: Allow Charge Operating Unit Update. If profile is set to yes, you can override the default operating unit.

3.11 Setting Up HTML Applications for Service Requests

This step is required to enable HTML applications from the Service Request. HTML applications are Oracle Quality Online, Oracle Knowledge Management and Install Base.

Oracle Quality Online

1. Login to jtflogin as sysadmin with responsibility CRM HTML administration
2. Select Users under User Mgmt.
3. Search for an appropriate user name.
4. Click on Assign Roles.
5. Assign the 'OQO Defect User' role to your support user.
6. Click **Update**.

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**.

Install Base

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

3.12 Setting Up Support Related 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. 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 profile options override those set at site and application level. User level profile options apply to individual users identified by their application user names. User level profile options 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 Support related 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 Oracle Support unless otherwise indicated.

ASO Product Organization

This profile values identifies the organization used for products for Oracle Quality. This is required even if Oracle Quality is not used.

Valid values: Any valid 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 11.5.6 with the customer of 'EMPLOYEE'

Valid values: All internal customers from HZ_PARTIES

Default value: None

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: Show web available resource

This profile is used while invoking Assignment Manager from a service request. If this profile is set to Yes, only the resources available for taking web calls are returned by Assignment Manager.

Valid values: Yes, No

Default value: No

Service: Skill Bank Usage for Assignment Manager

This profile is used to determine if Assignment Manager checks the skill bank to return the skill rate.

Valid values: Yes, No

Default value: No

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

Obsolete Profile Options

- Service: Default web Service Request Owner
- Service: Default web Service Request Severity
- Service: Default web Service Request type
- Service: Default web Service Request Urgency

3.13 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.

3.14 Creating a Support Site

A Support Site is created in order to stripe a support organization into different locations or divisions, and can be included as one of the qualifiers in the Assignment manager. An individual resource can be attached to a support site in the Resource Manager.

Follow these steps 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 Organization 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.

3.15 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 Responsibilty 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.

A

Service Request Public APIs

Oracle Support provides a broad horizontal solution that may be used across a large range of industries. It provides customer support organizations with the knowledge and tools to help resolve customer issues consistently and quickly, thereby decreasing costs, increasing profitability, and enhancing customer relatioships.

The Oracle Customer support application offers an integrated approach to managing customer information and handling customer issues.

A.1 Service Request Public Packages

The Service Request API is used to create and update a Service Request.

This can also be used to create notes using the JTF_NOTES API. The Link KB Statement Procedure links a Knowledge Management statement with a Service Request. The Link KB Solution Procedure links a Knowledge Management solution with a Service Request. The Status, Severity, Problem Code, Urgency and Owner can be updated by the Service Request API. The Service Request API also launches the Work Flow based on customer profile setups.

Related documents for service request can be created and updated using the following API. The package that is used for this is cs_incidentlinks_pub. The create_lincidentlink procedure can create links between two service requests. The delete incident link API is used to delete the link.

The create_lincidentlink_ext procedure can create links between a service request and any other document. To create an external link the document must have a valid object code in JTF_OBJECTS and must also have a valid SQL stored against the object code. The Object code must be linked to the Object user code in JTF_OBJECT_USAGES. The delete incident link ext API is used to delete the link.

Example Listing

- CS_SERVICEREQUEST_PUB
- CS_INCIDENTLINKS_PUB

The following tables describe the public APIs discussed in this chapter.

APIs for CS_SERVICEREQUEST_PUB

Name	Description
Create ServiceRequest	This procedure creates a Service Request.
Update ServiceRequest	This procedure updates a Service Request with the given parameters. All the fields that are changed are updated.
Update Status	This procedure updates the status of a service request
Update Serverity	This procedure updates the severity of a service request
Update Urgency	This procedure updates the urgency of a service request
Update Owner	This procedure updates the owner of a service request
Update Problem Code	This procedure updates the problem code of a service request
Link KB Statement	This Procedure Links a Knowledge Management statement with a Service Request
Link KB Solution	This Procedure Links a Knowledge Management solution with a Service Request

APIs for CS_INCIDENTLINKS_PUB

Name	Description
Create Incident Link	This procedure creates a link between a Service Request and another.
Create Incident Link ext	This procedure creates a link between a Service Request to another externa document. (E.g Defect, Sale order)

Delete Incident Link	This procedure deletes the link between two service requests.
Delete Incident Link ext	This procedure deletes the link between the service requests and the external document.
Update Incident Link	This procedure updates the link between two Service requests.
Update Incident Link ext	This procedure is deprecated. Not to be used any more.

A.2 Package Service Request

First public package description.

- CS_SERVICEREQUEST_PUB
- CS_INCIDENTLINKS_PUB

A.2.1 CS_SERVICEREQUEST_PUB

This Public API allows for the creation and updating of a Service Request.

Procedure Specification

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    VARCHAR2,
x_msg_count                  OUT    NUMBER,
x_msg_data                   OUT    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,
  x_request_id                  OUT    NUMBER,
  x_request_number              OUT    VARCHAR2,
  x_interaction_id          OUT    NUMBER,
  x_workflow_process_id         OUT    NUMBER
);


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    VARCHAR2,
  x_msg_count                   OUT    NUMBER,
  x_msg_data                    OUT    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    NUMBER,
  x_interaction_id              OUT    NUMBER
);

PROCEDURE initialize_rec(
  p_sr_record                     IN OUT service_request_rec_type
);
```

```
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     VARCHAR2,
  x_msg_count                   OUT     NUMBER,
  x_msg_data                    OUT     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_flagIN       VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id              OUT     NUMBER
);

PROCEDURE Update_Severity(
  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     VARCHAR2,
  x_msg_count                   OUT     NUMBER,
  x_msg_data                    OUT     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_severity_id                 IN      NUMBER   := NULL,
  p_severity                    IN      VARCHAR2 := NULL,
  p_audit_comments              IN      VARCHAR2 := NULL,
  p_comments                    IN      VARCHAR2 := NULL,
  p_public_comment_flagIN       VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id              OUT     NUMBER
);

PROCEDURE Update_Urgency
( 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     VARCHAR2,
  x_msg_count                   OUT     NUMBER,
  x_msg_data                    OUT     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_urgency_id                  IN      NUMBER   := FND_API.G_MISS_NUM,
  p_urgency                     IN      VARCHAR2 := FND_API.G_MISS_CHAR,
  p_audit_comments              IN      VARCHAR2 := NULL,
  p_comments                    IN      VARCHAR2 := NULL,
  p_public_comment_flagIN       VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id              OUT     NUMBER
);

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     VARCHAR2,
  x_msg_count                     OUT     NUMBER,
  x_msg_data                      OUT     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_flagIN         VARCHAR2 := FND_API.G_FALSE,
  x_interaction_id                OUT     NUMBER
);

PROCEDURE Update_Problem_Code(
  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     VARCHAR2,
  x_msg_count                     OUT     NUMBER,
  x_msg_data                      OUT     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_problem_code                  IN      VARCHAR2,
  p_comments                      IN      VARCHAR2 := NULL,
```

```
 p_public_comment_flagIN       VARCHAR2 := FND_API.G_FALSE,
 x_interaction_id              OUT     NUMBER
);

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     VARCHAR2,
 x_msg_count                   OUT     NUMBER,
 x_msg_data                    OUT     VARCHAR2,
 p_request_id                  IN      NUMBER,
 p_statement_id                IN      NUMBER,
 p_is_statement_true      IN      VARCHAR2,
 x_statement_link_id      OUT     NUMBER
);

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     VARCHAR2,
 x_msg_count                   OUT     NUMBER,
 x_msg_data                    OUT     VARCHAR2,
 p_request_id                  IN     NUMBER,
 p_solution_id                 IN     NUMBER,
 p_is_solution_true            IN     VARCHAR2,
 x_solution_link_id            OUT     NUMBER
 );
```

Current Version

2.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

CS_SERVICEREQUEST_PUB IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
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_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_last_updated_by	NUMBER	Yes	User Id
p_last_update_login	NUMBER	Optional	Login Id

p_last_update_date	DATE	Optional	Update Date

The following table describes the OUT parameters associated with this API.

CS_ServiceRequest_PUB Out Parameters

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_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
x_workflow_process_id	NUMBER	If a new workflow gets launched during an update then this parameter will have the workflow process id. New workflow will get launched only if the service request does nit have an active workflow associated with it.

A.2.2 CS_INCIDENTLINKS_PUB

This Public API allows for the creation,updating and deletion of Related Documents for a Service Request.

Procedure Specification

```
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,
X_RETURN_STATUS                OUT    VARCHAR2,
```

```
X_MSG_COUNT          OUT    NUMBER,
X_MSG_DATA  OUT              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_ORG_ID                     IN     NUMBER        := NULL,
P_FROM_INCIDENT_IDIN     NUMBER         := NULL,
P_FROM_INCIDENT_NUMBER IN     VARCHAR2   := NULL,
P_TO_INCIDENT_ID             IN     NUMBER        := NULL,
P_TO_INCIDENT_NUMBER     IN     VARCHAR2   := NULL,
P_LINK_TYPE                  IN     VARCHAR2,
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_CONTEXT               IN     VARCHAR2      := FND_API.G_MISS_CHAR,
X_LINK_ID                    OUT    NUMBER
);

PROCEDURE CREATE_INCIDENTLINK_EXT (
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    VARCHAR2,
X_MSG_COUNT          OUT    NUMBER,
X_MSG_DATA                   OUT    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_ORG_ID                     IN     NUMBER        := NULL,
P_FROM_INCIDENT_IDIN     NUMBER         := NULL,
P_FROM_INCIDENT_NUMBERIN     NUMBER         := NULL,
```

```
P_TO_OBJECT_ID              IN    NUMBER,
P_TO_OBJECT_TYPE            IN    VARCHAR2,
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_CONTEXT              IN    VARCHAR2      := FND_API.G_MISS_CHAR,
X_LINK_ID                   OUT   NUMBER
);

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,
X_RETURN_STATUS             OUT   VARCHAR2,
X_MSG_COUNT         OUT   NUMBER,
X_MSG_DATA                  OUT   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_ORG_ID                    IN    NUMBER        := NULL,
P_LINK_ID                   IN    NUMBER
);

PROCEDURE DELETE_INCIDENTLINK_EXT (
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   VARCHAR2,
X_MSG_COUNT         OUT   NUMBER,
X_MSG_DATA                  OUT   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_ORG_ID                      IN    NUMBER        := NULL,
P_LINK_ID                     IN    NUMBER
);

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,
X_RETURN_STATUS               OUT   VARCHAR2,
X_MSG_COUNT           OUT   NUMBER,
X_MSG_DATA                    OUT   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_ORG_ID                      IN    NUMBER        := NULL,
P_LINK_ID                     IN    NUMBER,
P_FROM_INCIDENT_IDIN    NUMBER        := NULL,
P_FROM_INCIDENT_NUMBERIN    VARCHAR2  := NULL,
P_TO_INCIDENT_ID              IN    NUMBER        := NULL,
P_TO_INCIDENT_NUMBER    IN    VARCHAr2  := NULL,
P_LINK_TYPE                   IN    VARCHAR2      := NULL,
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_CONTEXT                IN    VARCHAR2      := FND_API.G_MISS_CHAR
);

PROCEDURE UPDATE_INCIDENTLINK_EXT (
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    VARCHAR2,
X_MSG_COUNT          OUT    NUMBER,
X_MSG_DATA                  OUT    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_ORG_ID                    IN     NUMBER        := NULL,
P_LINK_ID                   IN     NUMBER,
P_FROM_INCIDENT_IDIN    NUMBER        := NULL,
P_FROM_INCIDENT_NUMBERIN    VARCHAR2   := NULL,
P_TO_OBJECT_ID              IN     NUMBER        := FND_API.G_MISS_NUM,
P_TO_OBJECT_TYPE            IN     VARCHAR2       := FND_API.G_MISS_CHAR,
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_CONTEXT              IN     VARCHAR2       := FND_API.G_MISS_CHAR
);
```

Current Version

2.0

Parameter Descriptions

The following table describes the IN parameters associated with this API.

CS_INCIDENTLINKS_PUB IN Parameters

Parameter	Data Type	Required	Descriptions and Validations
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 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_from_ incident_id	NUMBER	Yes	The service request id from where the link is created.
p_from_ incident_number	VARCHAR2	Yes	The service request number from where the link is created.
p_to_incident_ id	NUMBER	Yes	The service request id to which the link is created.
p_ to_incident_ number	VARCHAR2	Yes	The service request number to which the link is created.
p_link_type	VARCHAR2	Yes	The Link type.
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_context	VARCHAR2	Yes	The context for flex fields
p_to_object_id	NUMBER	Yes	The unique id of the external document.
p_to_object_ type	VARCHAR2	Yes	The Object code that identifies the external document. This must be a valid code in JTF_OBJECTS.

The following table describes the OUT parameters associated with this API.

CS_ServiceRequest_PUB Out Parameters

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_link_id	NUMBER	The unique id that represents each link.

A.3 Data Structure Specifications

The following data structures are used in CS_SERVICEREQUEST_PUB:

- NOTES_REC - Notes Record Type holds the attributes for the Notes Table
- CONTACTS_REC - Contacts Record Type holds the attributes for the Contacts Table
- SERVICE_REQUEST_REC_TYPE - Service Request Record Type holds the attributes for the Service Request Creation

A.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. The 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
);
```

A.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
);
```

A.3.3 Service_request_rec_type

The Service Request Record Type holds the attributes for the Service Request Creation.

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(80),
     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_versionVARCHAR2(250),
     db_versionVARCHAR2(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),
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(3),
inv_component_id           NUMBER,
inv_component_version      VARCHAR2(3),
inv_subcomponent_id        NUMBER,
inv_subcomponent_version   VARCHAR2(3),
tierVARCHAR2(250),
tier_versionVARCHAR2(250),
operating_system VARCHAR2(250),
databaseVARCHAR2(250),
cust_pref_lang_idNUMBER,
category_idNUMBER,
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)
);
```

A.4 Messages and Notifications

The status messages associated with the Service Request public APIs are divided into the following categories:

- Common Messages
- CS_SERVICEREQUEST_PUB
- CS_INCIDENTLINKS_PUB

A.4.1 Common Messages

The following table describes a lists of common error messages and notifications that the Service Request API can generate. Note that not all messages are returned by every APIs.

Common ServiceRequest API Messages

Error messages for CS_SERVICEREQUEST_PUB are in the same format as Common Messages.

Num-ber	Type	Name	Text

1388		PROFILES_CANNOT_READ	Cannot read value for profile option &OPTION in routine &ROUTINE. Possible values for &OPTION CS_ID_FLEX_CODE Possible vlaues for & ROUTINE G_PKG_NAME.Create_ServiceRequest G_PKG_NAME.Update_ServiceRequest G_PKG_NAME.Update_Status G_PKG_NAME.Update_Serverity G_PKG_NAME.Update_Urgency G_PKG_NAME.Update_Owner G_PKG_NAME.Update_Problem_Code
		CS_API_ALL_MISSING_ORG_ID	API Programming Error (&API_NAME): The parameter ORG_ID is required in a multi-org environment. Possible values for &API_NAME Create_ServiceRequest Update_ServiceRequest Update_Status Update_Serverity Update_Urgency Update_Owner Update_Problem_Code
		CS_API_ALL_ORG_ID_IGNORED	API Programming Warning (&API_NAME): The parameter ORG_ID cannot be set because Multi-Org is not enabled

		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). Possible Values for &TRUNCATED_PARAM p_status p_severity p_urgency p_type_name p_status_name p_severity_name p_urgency_name p_employee_number p_cp_ref_number p_request_number &VAL_LEN is the length of the input string and &DB_LEN is the defined length of that string.

API Messages

Number	Type	Name	Text
1388		PROFILES_CANNOT_READ	Cannot read value for profile option &OPTION in routine &ROUTINE.
		CS_API_ALL_MISSING_ORG_ID	API Programming Error (&API_NAME): The parameter ORG_ID is required in a multi-org environment.
		CS_API_ALL_ORG_ID_IGNORED	API Programming Warning (&API_NAME): The parameter ORG_ID cannot be set because Multi-Org is not enabled

		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).