**Oracle® Service Contracts**

Implementation Guide

Release 11*i*

**Part No. B19364-01**

August 2005

ORACLE

Oracle Service Contracts Implementation Guide, Release 11*i*

Part No. B19364-01

# Contents

**4 Setting Up Contract Events**

**5 Setting Up Automatic Renewals**

**6 Enabling Other Setups**

**A System Profiles**

**B Lookups**

**C Concurrent Programs**

# Send Us Your Comments

**Oracle Service Contracts Implementation Guide, Release 11*i***

**Part No. B19364-01**

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. 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 about this manual?

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

- Electronic mail: appsdoc_us@oracle.com
- FAX: 650-506-7200 Attn: Oracle Supply Chain Management Documentation Manager
- Postal service:
  Oracle Supply Chain Management Documentation Manager
  Oracle Corporation
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

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

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

# Preface

## Intended Audience

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

See Related Documents on page viii for more Oracle Applications product information.

## TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## 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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ .

## Accessibility of Code Examples in Documentation

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

## Accessibility of Links to External Web Sites in Documentation

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

## Structure

1 **Overview of Implementing Oracle Service Contracts**
2 **Setting Up Oracle Service Contracts**
3 **Setting Up Service Pricing and Billing**

# Related Documents

**Related Service Contracts Guides**:

*Oracle Service Contracts User Guide*

*Oracle Contracts Implementation and Administration Guide*

**Other Related Guides**:

*Oracle Advanced Pricing User's Guide*

*Oracle Applications User's Guide*

*Oracle Bills of Material User's Guide*

*Oracle Common Application Components Implementation Guide*

*Oracle Depot Repair Implementation Guide*

*Oracle General Ledger User Guide*

*Oracle Human Resources Management Systems Implementation Guide*

*Oracle Install Base Implementation Guide*

*Oracle Inventory User's Guide*

*Oracle Knowledge Management Implementation Guide*

*Oracle Order Management User's Guide*

*Oracle TeleService Implementation Guide*

*Oracle Territory Manager Implementation Guide*

*Oracle Work in Process User's Guide*

# Do Not Use Database Tools to Modify Oracle Applications Data

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

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

Because Oracle Applications tables are interrelated, any change you make using an
Oracle Applications form can update many tables at once. But when you modify Oracle
Applications data using anything other than Oracle Applications, you may change a row
in one table without making corresponding changes in related tables. If your tables get

out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

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

# 1

# Overview of Implementing Oracle Service Contracts

This chapter covers the following topics:

- Confirming Oracle Applications Setups
- About Oracle Service Contracts Setups

## Confirming Oracle Applications Setups

Before you implement Oracle Service Contracts, confirm the following:

| Required | Setup |
| --- | --- |
| Yes | Set up System Administrator, page 1-2 |
| Yes | Define Key Flexfields, page 1-2 |
| Yes | Define Calendar, Currency, and Set of Books, page 1-3 |
| Yes | Set up Organizations, page 1-4 |
| Yes | Set up Employees, page 1-4 |
| Yes | Set up Oracle Inventory, page 1-4 |
| Optional | Set up Oracle Bills of Material, page 1-5 |
| Optional | Set up Oracle Work in Process, page 1-5 |
| Yes | Set up Oracle Order Management, page 1-5 |
| Yes | Set up Oracle Advanced Pricing, page 1-5 |
| Optional | Set up Oracle TeleService, page 1-5 |
| Optional | Set up Oracle Knowledge Management, page 1-5 |
| Optional | Set up Notes, page 1-5 |
| Yes | Set up Resources, page 1-5 |
| Yes | Set up Install Base, page 1-5 |
| Optional | Set up Oracle Depot Repair, page 1-5 |
| Optional | Set up Oracle Contracts, page 1-5 |

### Step 1: Set Up System Administrator

Perform the following setups:

- Define responsibilities. See *Oracle Applications System Administrator's Guide*.

- Set up printers (optional). See the topic about setting up printers in *Oracle Applications System Administrator's Guide*.

### Step 2: Define Key Flexfields

Coordinate the flexfield setup among other applications such as Oracle Human Resources Management Systems and Oracle Inventory, before defining the key flexfields for Oracle Service Contracts. You should not modify flexfields frequently. For more information, see *Oracle Applications Flexfields Guide.*

For each key flexfield, perform the following tasks (some are optional):

- Define the flexfield structure

- Define value sets

- Define flexfield segments

- Define flexfield segment values

- Define security rules

- Assign security rules

- Define roll–up groups

- Define cross–validation rules

Set up the following Accounting flexfield (you may not need to perform this step if you have already installed and set up Oracle General Ledger or performed a common–applications setup. For additional information, see *Oracle General Ledger User Guide.*

Set up the following Human Resources key flexfields (you may not need to set up these key flexfields if you have already installed and set up Oracle Human Resource Management Systems or performed a common–applications setup. For additional information, see *Oracle Human Resources Management Systems Implementation Guide.*

- Grade

- Job

- Position

- People Group

## Step 3: Define Calendars, Currencies, and Set of Books

If you defined your calendars, currencies, and set of books for other Oracle Applications, proceed to the next step.

> **Note:** If you are performing a Multi–Org implementation, you may optionally create more than one calendar, currency, or set of books. See:*Multiple Organizations in Oracle Applications*.

Perform the following tasks:

- Set up calendars. See Defining Calendars in *Oracle General Ledger User Guide.*

- Define period types. See Defining Period Types in *Oracle General Ledger User Guide.*

- Define accounting calendar. See Defining Calendars in *Oracle General Ledger User Guide.*

- Define transaction calendar. See Defining Transaction Calendars in *Oracle General Ledger User Guide.* (Optional)

- Define workday calendar. See Overview of Workday Calendar in *Oracle Bills of Material User's Guide.* (Optional)

- Define exception templates. See Creating a Workday Exception Template in *Oracle Bills of Material User's Guide.* (Optional)

- Define currencies. See Defining Currencies in *Oracle General Ledger User Guide.*

- Define conversion rate types. See Defining Conversion Rate Types in *Oracle General Ledger User Guide.*

- Assign your set of books to a responsibility. See Assigning Responsibility to Set of Books in *Oracle General Ledger User Guide.*

- Set up currency rates.

- Set up accounting code combinations. See Setting up Accounting Code Combinations in *Oracle General Ledger User Guide.*

- Open and close accounting periods. See Opening and Closing Accounting Periods in *Oracle General Ledger User Guide.*

## Step 4:  Set Up Organizations

You may not need to perform this step if you have installed and set up Oracle Inventory or performed an Oracle Common Application Components setup. For the following tasks relating to setting up organization, see the *Oracle Human Resources Management Systems Implementation Guide.*

- Define organization QuickCodes

- Define business groups

- Define organizations

- Define human resources organizations

- Define legal entities organizations

- Define Oracle users for operating–unit organizations

- Assign operating units to legal entities

- Set up inventory organizations. For the tasks relating to setting up inventory organizations, see *Oracle Inventory User's Guide*

- Define organization hierarchies. See *Oracle Human Resources Management Systems Implementation Guide*

- Assign business groups and operating units to responsibilities (make sure that the profile option *HR: Business Group* is set at the responsibility level to the business group for that responsibility. See *Oracle Human Resources Management Systems Implementation Guide*

- Define profile options:

  - MO: Security Profile

  - Top Reporting Level

## Step 5:  Set Up Employees

You enter and maintain employees in Oracle Human Resources Management Systems. If you do not install Oracle Human Resources Management Systems with Oracle Depot Repair, you can use the Enter Employee form to define and maintain employees in Oracle Purchasing.

## Step 6:  Confirm Setup of Oracle Inventory

Confirm the setup of Oracle Inventory. If your company sells services but does not sell goods you will still need to setup contract items in Oracle Inventory. See the *Oracle Inventory User's Guide.*

> **Note:** Prior to setting up Items in Oracle Inventory, you must set up Coverage Templates, *Oracle Service Contracts User Guide* and Subscription Templates, *Oracle Service Contracts User Guide*.

### Step 7: Confirm Setup of Oracle Bills of Material

Confirm the setup of Oracle Bills of Material. If your company sells services but does not sell goods you do not need to perform this setup. See the *Oracle Bills of Material User's Guide.*

### Step 8: Confirm Setup of Oracle Work in Process

Confirm the setup of Oracle Work in Process. If your company sells services but does not sell goods you do not need to perform this setup. See the *Oracle Work in Process User's Guide.*

### Step 9: Confirm Setup of Oracle Order Management

Confirm the setup of Oracle Order Management. See the *Oracle Order Management User's Guide.*

### Step 10: Confirm Setup of Oracle Advanced Pricing

Oracle Service Contracts uses Oracle Advanced Pricing for discounts and pricing. In addition, the APIs which Oracle Service Contracts calls for pricing covered products are part of Oracle Advanced Pricing. Oracle Advanced Pricing is required to implement that feature. Confirm the setup of Oracle Advanced Pricing. See the *Oracle Advanced Pricing User's Guide*.

> **Note:** There are limited situations where a customer would not use Oracle Advanced Pricing when implementing Oracle Service Contracts.

### Step 11: Confirm Setup of Oracle TeleService

Confirm the setup of Oracle TeleService. See the *Oracle TeleService Implementation Guide.*

### Step 12: Confirm Setup of Oracle Knowledge Management

Confirm the setup of Oracle Knowledge Management. See *Oracle Knowledge Management Implementation Guide.*

### Step 13: Set Up Notes

Set up Notes. See the *Oracle Common Application Components Implementation Guide*

### Step 14: Set Up Resources

Set up Resources. See the *Oracle Common Application Components Implementation Guide*.

### Step 15: Confirm Setup of Oracle Install Base

Confirm the setup of Oracle Install Base. See the *Oracle Install Base Implementation Guide*.

### Step 16: Confirm Setup of Oracle Depot Repair

Confirm the setup of Oracle Depot Repair. See the *Oracle Depot Repair Implementation Guide.*

### Step 17: Confirm Setup of Oracle Contracts

Confirm the setup of Oracle Contracts. See the *Oracle Contracts Implementation and Administration Guide.*

# About Oracle Service Contracts Setups

The following table provides an overview of this guide, and highlights procedures and appendices that you will use in your implementation.

**Implementation Setups**

**Setting Up Oracle Service Contracts - General Setups**

Extend lookup codes, page 2-1

Register in Oracle Quoting, page 2-1

Map time units of measure, page 2-2

Set up status and operations, page 2-2

Register a new source as a JTF object, page 2-3

Define a process, page 2-5

Define quality assurance checklists, page 2-6

Define roles and role sources, page 2-7

Set up categories, page 2-8

Define coverage types, page 2-9

Define coverage and subscription templates, page 2-9

Create contract items in Oracle Inventory, page 2-10

Define contract groups, page 2-10

Set up autonumbering, page 2-10

Define service availability, page 2-15

Define service cotermination, page 2-16

**Setting Up Service Pricing and Billing**

Create billing profiles, page 3-1

Enable invoice level loading, page 3-1

Enable invoice previewing, page 3-2

Set up transaction types, page 3-2

Set up batch transaction sources, page 3-6

Set up transaction flexfield segments, page 3-8

Price service, subscription, and usage, page 3-9

Enable multi-currency price list support, page 3-10

Enable advanced attribute sourcing, page 3-10

Run the build attribute mapping rules for pricing qualifiers and attributes, page 3-11

Define sales territories to assign vendor contacts, page 3-11

**Implementation Setups**

Execute billing through concurrent programs, page 3-13

**Setting Up Events**

Define condition templates, page 4-2

Use query conditions, page 4-3

Review errors from asynchronous processing, page 4-3

Use the events controller, page 4-4

Enable service request creation based on contract events, page 4-5

Start the event and outcome listeners, page 4-7

Start the workflow background process, page 4-9

Run the date assembler, page 4-10

**Setting Up Automatic Renewals**

Creating a process definition for renewals, page 5-2

Define independent conditions for the renewal, page 5-4

Confirm renewal defaults, page 5-6

Start the event and outcome listeners for renewals, page 5-6

Start the workflow background process for renewals, page 5-6

Run the date assembler for renewals, page 5-7

Confirm the automatic renewal, page 5-7

Create and enable templates for e-mails to customers, page 5-7

Troubleshoot the automatic renewal process, page 5-7

**Enabling Other Setups**

Enable the quick menu, page 6-1

Define the contract terms library, page 6-2

Migrate clauses to the contract terms library, page 6-2

**Appendix: System Profiles**

Set up system profile options, page A-1

**Appendix: Lookups**

Define lookup codes, page B-1

**Appendix: Concurrent Programs**

Run concurrent programs, page C-1

# 2

# Setting Up Oracle Service Contracts

This chapter covers the following topics:

- Extending Lookup Codes
- Registering in Oracle Quoting
- Mapping Time Units of Measure
- Setting Up Status and Operations
- Registering a New Source as a JTF Object
- Defining a Process
- Defining Quality Assurance Checklists
- Defining Roles and Role Sources
- Setting Up Categories
- Defining Coverage Types
- Defining Coverage and Subscription Templates
- Creating Contract Items in Oracle Inventory
- Defining Contract Groups
- Setting Up Autonumbering
- Defining Service Availability
- Defining Service Cotermination

## Extending Lookup Codes

There are lookup codes that you can define. These are listed in an Appendix within this guide. See Defining Lookups, page B-1.

## Registering in Oracle Quoting

Service Contracts must be registered in Oracle Quoting, to enable contracts to be created for any immediate services or delayed services, as well as, warranties and extended warranties originating from Oracle Order Management.

**Prerequisite:**

Define Oracle Service Contracts as an application (System Administrator).

### To register in Oracle Quoting:

1. Select the Quoting Sales Manager responsibility, then navigate to Setup and select Quick Codes.

2. Query lookup: ASO_ORDER_FEEDBACK_CRM_APPS.

3. From the toolbar select the **New** icon (plus sign).

4. Enter OKS in the Code field.

5. Enter Service Contracts in the Meaning field.

6. Optionally, enter a description.

7. Enter the effective dates.

   You do not need to enter an end date.

8. Save.

## Mapping Time Units of Measure

Oracle Service Contracts defines unit of measure conversion for time differently than other Oracle applications. This helps ensure that the scheduling is more accurate than a simple conversion such as 1 month = 30 days, which is only correct for 5 out of the 12 months. To define your own time unit conversions for extending a contract or for scheduling, you must define your own time unit conversions.

There are six internal base time units: seconds, minutes, hours, days, months, and years. Make sure to map each time unit you want to use in Oracle Service Contracts. An example of a mapping: Day (your definition) = 1 day (base definition).

**Prerequisite**:

Define new unit of measure code with class code Time in the Units of Measures form before mapping it to Service Contracts Base Time Units

### To map time units of measure:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Units of Measure > Time Units of Measure.

2. From the toolbar select the **New** icon (plus sign).

3. Select a user unit from the User Unit LOV. You should have previously defined this in Unit of Measures form.

4. Select the base unit of measure from the Base Unit LOV.

5. Enter conversion information.

6. Optionally, enter a description.

7. Save.

   **Caution:** After you save, definitions cannot be deleted or updated.

## Setting Up Status and Operations

You can control the operations such as whether a contract can be updated online. This depends upon the category assigned to the contract when it is created and the status of

the contract, such as Active or Terminated. For example, if you define a new status for the status type Active, then you must make sure that you specifically allow operations such as online update. If you create a new status without specifying any allowed operations, then you implicitly allow no operations for this contract status. In order for the concurrent program Status Change to automatically update contract status, you must define a default status for each status type.

> **Note:** There are several statuses seeded with the application, however, you can also set up new statuses.

## To define status:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Status and Operations.

2. Query a status type, such as Active.

3. Navigate to the Statuses region.

4. From the toolbar select the **New** icon (plus sign).

5. Enter a status, such as ACTIVE.

6. Enter a meaning.

7. Optionally, enter a description.

8. Confirm the start date.

9. Optionally, enter an end date.

10. To make this the default status for the status type, select the Default check box.

> **Note:** Each status type can have only one default.

11. Save.

## To define operations:

1. Select a status, such as Active.

2. In the Allowed Operations by Category region, place your cursor in the Category field and select a Category from the LOV, such as Service Agreement.

3. Select an Operation from the LOV, such as Update by Change Request.

4. Select the Allowed check box.

5. Save.

# Registering a New Source as a JTF Object

You can implement Oracle Service Contracts to pick up data from sources other than those seeded with the application. You can use these sources when defining: role sources and process definitions. For example, contacts on the contract can be sourced from your own legacy applications. The process of making any kind of data available as sources consists of two procedures:

- Defining a view.

- Integrating the view into a JTF object.

> **Note:** This setup is not required for all implementations. There are only certain situations when you must perform this setup.

## To define a view:

To define a view that can be integrated into a JTF object, the view must have the format outlined in the following table:

| Name | Null | Type |
|------|------|------|
| ID1 | NOT NULL | NUMBER |
| ID2 | NOT NULL | NUMBER |
| Name | NOT NULL | VARCHAR2(11) |
| DESCRIPTION | NOT NULL | VARCHAR2(31) |
| PRIMARY_UOM_CODE | NOT NULL | VARCHAR2(0) |
| STATUS | NOT NULL | CHAR(1) |
| START_DATE_ACTIVE | -- | DATE |
| END_DATE_ACTIVE | -- | DATE |

ID1 and ID2 ensure the uniqueness of the rows. If ID1 is selective enough to be unique, you can fill in a dummy character like '#' as ID2.

> **Note:** For line items, you need the column primary_uom_code only if you want to price the line.

## To integrate the view into a JTF Object:

This procedure gives you an example for the typical case of registering a view as a JTF object. You can also use this procedure for creating a source for process definition parameters.

1. Select the CRM Administrator responsibility, then navigate to Task and Escalation Manager > Setup > Objects Meta-data. The Task Setup: Object Types window appears.

2. Enter a name.

3. Enter a description.

4. Enter a unique object code that will be used for internal identification.

5. Enter a start date.

6. In the Select Statement Details section, enter a value (view defined previously confirming standard format) for the From field.

7. Enter a value for the Where field.

8. In the Order by field, enter the object code name and column name you want to order by.

9. In the Usage section, select an Object User from the LOV.

10. Save.

# Defining a Process

Use this section to define a system routine procedure that can be invoked by either: an outcome, a function, a quality assurance checklist, auto numbering, a change request process, or to approve or renew.

A process may be either an Oracle Workflow process or procedure, either packaged or standalone.

> **Note:** If you define your own procedures or functions, they are not supported.

**Prerequisites**:

- Create the Oracle Workflow processes or procedures and packages first.

- To use a source for a process definition, create the source first. See Registering a New Source as a JTF Object, page 2-3.

## To define a process:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Process Definition.

2. Enter a unique name.

3. Enter the description.

4. Confirm effective dates (from and to).

5. Select a purpose from the list, such as Approve

6. Select a type from the list such as, PL/SQL, Workflow, Alert, Script.

   - If you select PLSQL, then enter the names for Package and Procedure.

   - If you select Workflow, then enter the Workflow Name and Workflow Process.

7. Click **Validate Name**.

8. In the Parameter region, choose the Basic tab and enter or select the following attributes:

   - **Name**

   - **Data Type**

   - **Description**

9. In the advanced region, when you choose the purpose **outcome** or **function**, you can define the parameters as a value from the document such as contract number (Object Name: OKC_K_HEADERS and Column Name: CONTRACT_NUMBER)

   In the Parameters region, choose the Advanced tab and enter the following attributes:

   - **Object Name**: (Optional) Enter an appropriate object source. This defines the source for lists of values for Function and Outcome parameters in the Condition form.

   - **Column Name**: (Required when Object Name is used) This allows users to select a column from the specified source as the parameter values for function and outcome for a condition line.

- **Description Column**: (Optional) Allows users to add description column to the dynamically created LOV used when creating condition lines.

10. Save.

# Defining Quality Assurance Checklists

The application validates a contract before you can submit it for approval using a quality assurance (QA) checklist. Each checklist consists of one or more processes.

In addition to the seeded QA checklist, you can define additional checklists. The seeded QA checklist is executed automatically for any contract, in addition to any other checklist you may wish to run against the contract. You can modify or delete processes that contain the User access level. If you create an additional checklist and would like it to default to your contracts during authoring, set the profile option OKS: Default QA Checklist.

## To define a quality assurance checklist:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Quality Assurance.

2. Enter a name.

3. Enter a description.

4. In the Processes region:

   1. Enter a value in the Run Sequence field.

   2. Select a process to include on the QA checklist from the Name LOV.

   3. Check the Active check box.

   4. From the Severity list, select one of the following levels:

      - **Warning**: A warning appears in the QA results but the contract requires no changes to continue to the approval process.

      - **Stop**: The contract does not pass QA if this process fails. An error appears in the QA results. This error must be corrected in the contract in order to pass the QA process and continue to the approval process.

   5. Optionally, select an access level.

      - **Extensible**: Only new child records can be added in a parent-child hierarchy.

      - **System**: Data cannot be updated or deleted. In a parent-child hierarchy, the addition of child level records cannot be done.

      - **User**: All operations are possible.

   6. Confirm the start date.

   7. Optionally, enter an end date.

5. Optionally, override the default values for the Parameters region. This region passes parameters to process.

6. Save.

# Defining Roles and Role Sources

Oracle Service Contracts provides party roles such as Vendor or Customer and contact roles such as Buyer or Salesperson. You can add your own roles to track relationships on the contract such as Partner or Service Agent. You can extend the lookups for:

- Party roles
- Contact roles

You can also define the LOVs that appear as sources for your roles.

## To define party roles:

1. Select the Service Contracts Manager responsibility, then navigate to > Setup > Contract > Categories and Sources> Define Party Roles.

   The Application Object Library: CONTRACT PARTY ROLES Lookups window appears.

2. Select the **New** icon (plus sign) from the toolbar.

3. To add a new party role, enter the following fields:

   - Code
   - Meaning
   - Description
   - Effective Dates
   - Enabled check box.

4. Save.

## To define contact roles:

1. Select the Service Contracts Manager responsibility, then navigate to > Setup > Contract > Categories and Sources> Define Contact Roles.

   The Application Object Library: Contact Role Lookups window appears.

2. Select the **New** icon (plus sign) from the toolbar.

3. To add a new contact role, enter the following fields:

   - Code
   - Meaning
   - Description
   - Effective Dates
   - Enabled check box.

4. Save.

## To define role sources:

You can modify the source of seeded party or contact roles with the access level User. You can also define the source for new party or contact roles that you set up. You cannot delete a source from a role, after you define it.

1. Select the Service Contracts Manager responsibility, then navigate to > Setup > Contract > Categories and Sources> Define Roles Sources.

2. Select a party role, such as Contract Customer.

3. In the Party Source tab, select a source from the Source LOV.

4. Service Contracts are always managed from the perspective of the vendor. Select Sell from the Intent list.

5. Confirm the start date.

6. Optionally, enter an end date.

7. Select a value from the Access LOV.

8. Select the Contact Source tab, and click the **New** icon (plus sign) in the toolbar.

9. Select a contact role from the LOV.

10. Select a source.

11. Select Sell from the Intent list.

12. Confirm the start date.

13. Optionally, enter an end date.

14. Select a value from the Access LOV.

15. Save.

# Setting Up Categories

A category is a type of contract, such as a service agreement, subscription agreement, or warranty. Contract categories are seeded in Oracle Service Contracts and do not need to be set up. User defined categories are not supported for Service Contracts because certain processes in the application are dependent upon data being present that may be missing in a user defined category. In addition, new line types cannot be added to Service Contracts categories. You must assign access to a category in order for a given responsibility to read or modify contracts. Optionally, you can add new party roles that you have defined to a seeded category.

## To set up categories:

1. Select the Service Contracts Manager responsibility, then navigate to > Setup > Contract > Categories and Sources> Define Categories.

2. Query the Service Agreement class.

   > **Note:** Complete the following steps for each category you use within the Service Agreement class.

3. Select the Responsibilities tab.

4. Select the responsibility to provide access from the Name LOV.

5. Select an access level.

6. Confirm the start date.

7. Save.

> **Note:** When determining what level of access to grant a user for an individual contract, the application checks both the level of access granted in the Define Categories form and the level of access granted in the Security section of the contract and gives the highest level of access. It does not work in a hierarchical fashion. For example, if the user is given Modify access to the Warranty and Extended Warranty contract category but is only given Read Only access on a particular extended warranty contract, the user will still be able to modify that contract based on the fact the highest level of access granted is Modify.

**To add party roles to categories**

8. Select the Service Contracts Manager responsibility, then navigate to > Setup > Contract > Categories and Sources> Define Categories.

9. Query the Service Agreement class.

> **Note:** Complete the following steps for each category you use within the Service Agreement class.

10. Select the Party Roles tab.

11. Select the party role to add from the Role LOV.

12. Confirm the start date.

13. Save.

# Defining Coverage Types

You can use the Coverage Types window to define coverage types.

## To define coverage types:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Define Coverage Types.

2. Enter a unique alphanumeric code, such as G.

3. Enter a name for the coverage type, such as Gold.

4. Enter a description for the coverage type.

5. Enter the importance level (numeric value). In the Coverage window the importance level automatically appears, after the Coverage Type is selected.

6. Confirm the effective dates.

7. Select the Enabled check box.

8. Save.

# Defining Coverage and Subscription Templates

Oracle Service Contracts enables you to define a standard set of templates, to create offerings for a common set of agreements used by your organization, and modify them

to meet your customers' requirements. You must set up coverage and subscription templates, before you can define the service and subscription items to sell.

See Overview of Coverage and Subscription Templates, *Oracle Service Contracts User Guide*.

# Creating Contract Items in Oracle Inventory

You must create contract items in Oracle Inventory to sell to your customers. See About Contract Item Types, *Oracle Service Contracts User Guide*.

# Defining Contract Groups

You can set up contract groups, both Public and Private, to help you organize contracts more effectively. Public groups are available to all users, while Private groups are available to the user who creates them.

See Creating Groups, *Oracle Service Contracts User Guide*.

# Setting Up Autonumbering

Autonumbering Contracts consists of the following topics:

- Overview of Autonumbering, page 2-10
- Understanding Autonumbering Features, page 2-11
- Understanding Autonumbering Examples, page 2-12
- Understanding the Truth Table, page 2-12
- To set up autonumbering, page 2-13
- To set up a user function, page 2-14

## Overview of Autonumbering

Autonumbering functionality automatically populates the contract number when you create it. Autonumbering provides greater flexibility in defining how contracts are numbered.

Contract numbers can be either sequential numbers or a combination of defined prefix and suffix alpha-numeric characters to classify a contract based on its attributes. These attributes include: site, business group, operating unit, class, and category.

You can set up auto-number classification using a prefix and suffix with a contract number. For example, you can number a service contract originating in the United States as US-9999-SRV. Use of a prefix and suffix is optional. The numeric part of the contract number will be formatted to the length specified for that sequence. For example number 99 will be formatted as 00099 if the length is specified as five digits.

Autonumbering of contracts is helpful in a scenario when contracts are entered or imported from an external source or system, built from another document, or when contract data is entered manually. From the Contracts Launchpad, a contract number can be automatically generated when you create a new contract by:

- Launching Contracts and selecting New from the Tools drop-down menu.

- Navigating to the Contract Navigator tab, selecting a contract group from the list, highlighting a contract from the right column, and either right clicking the mouse and selecting Copy or selecting Copy from the Tools drop-down menu. In the Copy form, the New Contract check box must be selected.

You can manually override the system generated number, assuming the Allow Manual check box is selected during setup. In addition, if there is no setup defined, manual numbering is assumed and the Contract Number field is not protected from update.

When a contract is created as a result of running a concurrent program, for example creating a service contract using the Service Contracts Order Capture Integration concurrent program, the contract number is automatically generated even if the Sequential Numbering profile option is not set up. That is the profile option is set to Always Used at the session level when a contract is created from a concurrent program.

## Understanding Autonumbering Features

You can use the Autonumbering form to set up contract numbering. In the top section, you decide how to number the contracts. There is only one header level record for the entire installation. In case of a user-defined function, there will be no details required and the Sequence Details region is grayed out.

### Site

A sequence number can be by installation site. One row is active in the Sequence Details region. This selection allows entry of Prefix, Suffix, Sequence Number From, Sequence Number To, Format Length, and Allow Manual override in the Sequence Details region. Business Group, Operating Unit, Class, and Category A sequence number can be generated for any of the above attributes. These attributes are selected from an LOV. Please refer to the Truth table for possible combinations of contract attributes for generating sequence numbers automatically.

### User Function

A user defined function offers the flexibility to build in more logic for generating a sequence number. For example, you can have the contract number reflect the date and sequence when the contract becomes active, such as 03222001001). Please refer to the Truth table for possible combinations of contract attributes for generating a sequence number automatically.

### Enforcing Number Generation

The Allow Manual check box indicates whether the contract number is a protected field or not. When this check box is not selected, the number is generated automatically and this field is protected. If you select this check box the contract number is automatically generated only when the contract number field is left blank.

### Prefix and Suffix

Free form fields help provide classification coding to a contract. You can use these fields to add a separator or delimiter. These values are characters only.

### Sequence Numbers

You may choose to specify a starting and ending number for the numeric part of the contract number sequence. If the contract number exceeds the specified range, depending on the selection of the Allow Manual check box, you are prompted to extend the range or enter a unique contract number.

**Format Length**

You can specify a formatting length for the numeric part of the contract number sequence. The total contract number length (concatenated suffix, number prefix) cannot exceed 120 characters.

**Template**

When a contract is created from a template, it generates a number automatically.

**Copy** When a contract is copied to a new contract, the attributes pertinent to the contract such as class and category do not change. The attributes such as Business Group and Operating Unit are based on the responsibility from which the copy is being created and not those of the source contract. A new number is generated accordingly.

## Understanding Autonumbering Examples

The following table shows examples of the First Number based on the various combinations selected.

| Bus Unit | Op Unit | Class | Cat | Prefix | Suffix | Seq From | Seq To | Length | First Number |
|---|---|---|---|---|---|---|---|---|---|
| USA | -- | Service | -- | US- | -SRV | 99 | 10000 | 5 | US-00099-SRV |
| UK | -- | Service | -- | UK- | -SRV | 1 | 20000 | 5 | UK-00001-SRV |

## Understanding the Truth Table

Depending on which autonumbering attribute you select, the Sequence Details fields are selectively enabled. For example, if you only select the Business Group, then Operating Unit, Class, Category sequence details are disabled. The following table (Truth Table) shows the possible sequences:

| Description | User Function | Site | Business Group | Operating Unit | Class | Category |
|---|---|---|---|---|---|---|
| User, no other attributes can be selected | Y | N | N | N | N | N |
| Site, no other attributes can be selected | N | Y | N | N | N | N |
| Business Group can be selected with Class | N | N | Y | N | Y | N |
| Business Group can be selected with Category | N | N | Y | N | N | Y |
| Business Group can be selected without Class or Category | N | N | Y | N | N | N |
| Operating Unit can be selected with Class | N | N | N | Y | Y | N |
| Operating Unit can be selected with Category | N | N | N | Y | N | Y |
| Operating Unit can be selected without Class or Category | N | N | N | Y | N | N |
| Class can be selected without Business Group/ Operating Unit | N | N | N | N | Y | N |
| Category selected without Business Group/ Operating Unit | N | N | N | N | N | Y |

## To set up autonumbering:

**Prerequisites**

The OKC: Document Sequence Name profile option should be enabled. This profile option is used to define a text literal, for example, OKC_DOC to be used for naming Sequences created.

The Sequential Numbering profile option should be set to **Always Used** for Oracle Service Contracts. By default, this profile is not set up. This profile is set at the application level. This means that Service Contracts users may use sequential numbering.

If Sequential Numbering Profile is set to **Always Used** and the Allow Manual check box is not selected in the AutoNumbering setup, you cannot manually enter a contract number when copying a contract to a new contract. It disables the Contract Number and Contract Number Modifier fields for the target contract.

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Autonumbering.

2. Select the attribute type, such as Site, User Function, Business or Group. If you select the User Function, choose a Process Name from the LOV.

3. Select the Sequence Details. Depending on attribute selected see, the Truth Table for attribute combinations from the LOV in the Sequence Details region.

4. Optionally, enter Sequence Numbers From/To.

5. Optionally, enter the Format Length for the contract number.

6. Optionally, enter Prefix/Suffix values.

7. To manually override the system-generated contract number, select the Allow Manual check box.

8. Save.

9. Select the Service Contracts Manager responsibility, then navigate to Launch Contracts > Open > Tools > New.

10. Select a Contract Category from the LOV.

11. Click **Create**.

12. Verify that the contract number is correct based on your setup.

> **Note:** If you try to change any of Autonumbering attributes, for example from Class to Category attribute, a Decision window displays the following warning: *You are changing the current contract numbering setup. Existing current numbering rules will no longer be effective. Do you want to continue?* If you click Yes, another window opens and asks: *Do you want to delete existing sequence numbering rules?* If you click **Yes**, the existing sequence numbering rules are deleted and there is no way you can retrieve these sequence numbering rules. If you click **No**, the existing sequence numbering rules will be kept in the system. If later on you want to revert to the original attribute, such as Class attribute in the above example, the sequence numbering rules appear when you change the attribute from Category to Class.

## To set up a user function:

A User Defined Function is a database procedure and must have the following parameters:

- x_contract_number OUT NOCOPY Varchar2

- x_return_status OUT NOCOPY Varchar2

> **Note:** Parameter x_contract_number should return the generated unique contract number. Parameter x_return_status should be: S if the program succeeds and return a contract number, E in case of error, and U for unexpected errors.

Use of NOCOPY is recommended for performance.

These two parameters should not be defined in the Process Definition but instead should be included in the user-defined procedure.

1. A User Defined Function must be registered in the Process Definition form. Select the Service Contracts Manager responsibility, choose Setup > Contract > Process Definition

2. Enter Auto Numbering for Purpose, PLSQL for Type. Then enter the Package and Procedure you want to define.

3. Enter the following parameters:

   **Parameters**

   Site

   Business Group

   Operating Unit

   Class

   Category

   Contract Currency

   Contract Amount

   Contract Party Information

4. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Autonumbering.

5. Check the User Function check box and select the user-defined process in the Process Name field.

   If you select the User Function check box in the Autonumbering form to generate contract number, you cannot select any sequence details in the Autonumbering form. The associated procedure takes care of autonumbering using any prefix, suffix, and formatting.

6. To allow manual override, select the Allow Manual check box

7. Save.

8. Select the Service Contracts Manager responsibility, choose Open > Contract Navigator > Tools > New.

9. Select a Contract Category from the LOV.

10. Click **Create**.

11. Verify contract number is correct based on your setup.

# Defining Service Availability

Service availability are services that can be sold to certain parties or to cover certain products. As well as defining those services that are available. Exceptions can be listed

to identify those services that are not available for parties or for products. The Generally Available check box determines whether the entries in Product and Party tabs are inclusions or exclusions. By selecting the Generally Available check box for a service, all products and parties listed are excluded from receiving that service. By leaving the Generally Available check box unselected, the products and parties listed are the only ones eligible to receive that service.

**Prerequisites:**

- Define service items and serviceable products in Oracle Inventory.

- Define parties in Oracle Receivables.

## To define service availability:

1. Select the Service Contracts Manager responsibility, then navigate to Setup > Contract > Service Availability.

2. Select a service item from the Name LOV.

3. From the Party tab, select the Generally Available check box if applicable.

4. Enter the effective dates.

5. Select a Party from the LOV.

6. Enter the start and end dates.

7. Select the Product tab and choose the Generally Available check box if applicable.

8. Enter the effective dates.

9. Select a Product from the LOV.

10. Enter the low and high item revision numbers if applicable.

    > **Note:** Oracle Service Contracts does not check revision level. The application checks whether service is available for that product or not.

11. Enter the start and end dates.

12. Save.

# Defining Service Cotermination

You can set any or all of a customer's service lines to a end on a predetermined day of the year. For example, some customers may require that any services that are sold to them at any time during the year all end on December 31st, so that they can review and negotiate their renewals at a single point each year.

**Party Level Cotermination**: For a coterminate day and month for services sold to a specific party, you must set up from Oracle Service Contracts. When you sell a service to that party, either in Service Contracts or in Order Management, the end date of the services can optionally be set to the established cotermination date. You can do this in Oracle Service Contracts by clicking the Coterminate or Coterminate All buttons on the Lines tab and Effectivities subtab. See *Oracle Order Management User's Guide* for guidance on how to coterminate service sold on sales orders.

**Prerequisite:**

Define parties and their customer accounts.

## To define service cotermination for a party :

1. Select the Service Contracts Manager responsibility and navigate to Setup > Contract > Service Cotermination.

2. Select a Party from the LOV.

   The corresponding customer accounts for the party appears in the Cotermination region.

3. Enter the cotermination day.

4. Enter the cotermination month.

5. Save.

# 3

# Setting Up Service Pricing and Billing

This chapter covers the following topics:

- Creating Billing Profiles
- Enabling Invoice Level Loading
- Enabling Invoice Previewing
- Setting Up Transaction Types
- Setting Up Batch Transaction Sources
- Setting Up Transaction Flexfield Segments
- Pricing Service, Subscription, and Usage Items
- Enabling Multi-Currency Price List Support
- Enabling Advanced Attribute Sourcing
- Running the Build Attribute Mapping Rules for Pricing Qualifiers and Attributes
- Defining Sales Territories to Assign Vendor Contacts
- Executing Billing Through Concurrent Programs

## Creating Billing Profiles

You can set up multiple, ad hoc billing profiles in Oracle Service Contracts. Billing profiles include information about accounting and invoicing rules, type of billing, and its frequency. Invoices are generated according to the billing profile attributes.

You can create billing profiles for a specific customer, a specific customer bill to site, or a generic one for all customers and sites.

For more information see Creating Billing Profiles, *Oracle Service Contracts User Guide*.

## Enabling Invoice Level Loading

The Service Contracts Main Billing process utilizes Invoice Level Loading, a parallel program to assign groups of invoices to different concurrent programs. The program first determines the count for all of the contract lines that qualify for the master request submission. If the count is greater than the threshold value (currently set to 500) the program splits up the master request into sub requests. There is no setup for this functionality other than to enable the profile option. For more details, see the OKS: Parallel Worker profile option, page A-1.

# Enabling Invoice Previewing

Before sending billing transactions directly to Oracle Receivables, you can preview the invoice information to ensure accuracy. When initiating the billing process, you can set up the application to place the transactions into a separate set of Service Contracts tables. These tables mirror the billing transaction tables used during the normal billing process.

**Guidelines:**

If necessary, you can create a simple report to access the required information in these tables.

The tables used for Invoice Preview are:

oks_bcl_pr

oks_bsl_pr

oks_btn_pr

oks_bsd_pr

oks_btl_pr

The preview process runs in much the same way as the normal billing process. It includes only those contracts that meet the same criteria that must be included in the actual contract billing process. For example, status is eligible for invoicing, bill on date is due.

If potential billing errors are found during the preview, you can make adjustments manually in the appropriate application, for example, Oracle Service Contracts or Oracle Install Base, and re-run the pre-invoice report. This iterative process would continue until the pre-invoice process gives the expected results, at which point you may choose to run the actual billing process.

The billing process uses the latest data in calculating the invoices, so it should be recognized that the actual invoices created may be different from those included in the last pre-invoice process, if there is sufficient delay between running the process and the actual billing process. Changes can occur such as counters updated, contracts terminated, new products added to coverage, which affect the actual invoices created from the Contracts application.

The process includes the same information normally sent to Oracle Receivables during the actual billing process. Service Contracts sends either detailed or summary information to Oracle Receivables, depending on the value of the OKS: Summary Transactions profile option.

# Setting Up Transaction Types

Transaction types define the accounting for the debit memos, charge backs, commitments, and invoices you create in Oracle Receivables. Transaction types also determine whether your transaction entries update your customer's balances and whether Oracle Receivables posts these transactions to your general ledger. To bill from Service Contracts, transaction types must be defined for invoices and credit memos.

## To set up invoice transaction types:

1. Select the Receivables responsibility, then navigate to Setup > Transactions > Transaction Types.

2. Enter a transaction name: Invoice-OKS (this OKS transaction type is case sensitive and must be entered as: Invoice-OKS). Optionally enter a description.

3. Select Invoice as transaction from the Class LOV.

4. Select the Open Receivable check box.

   This updates your customer balances each time you create a complete debit memo, chargeback, or on–account credit with this transaction type. Oracle Receivables also includes these transactions in the standard aging and collection processes.

5. Select the Post To GL check box to allow posting to general ledger.

6. Choose a default Printing Option for transactions with this transaction type. Select Print or Do Not Print. You can override this value when entering transactions.

7. Choose a Transaction Status of Open, Closed, Pending, or Void. Use these statuses to implement your own invoice approval system.

8. Optionally, select the Allow Freight check box to allow freight to be entered for transactions with this transaction type.

9. Select the Tax Calculation check box to let Oracle Receivables calculate tax for transactions with this transaction type.

10. Choose a Creation Sign. If you are using the Cash Basis accounting method, your transaction's creation sign must be either Positive or Negative. You cannot update this field after you enter transactions with this type.

11. To restrict the direction in which items with this transaction type can be updated by applications entered against them select the Natural Application Only check box. If you select this box, Oracle Receivables sets Allow Overapplication to No. You cannot update this option after you save this transaction type.

12. Optionally, select an Application Rule Set for this transaction type from the LOV. An Application Rule Set determines the default payment steps when you use the Applications window or AutoLockbox to apply receipts to transactions using this type. If you do not enter a rule set, Oracle Receivables uses the rule set in the System Options window as the default.

13. If you did not select the Natural Application Only check box, choose whether to Allow Overapplication against items with this transaction type by selecting or deselecting this box. If you select this check box, Oracle Receivables sets Natural Application to No and you cannot update it after you save this transaction type. If you use the Cash Basis accounting method, the default value is No and you cannot change it.

14. Enter the Credit Memo Type to use when crediting items with this transaction type (optional). When you enter a credit memo against an invoice with this transaction type, the value you enter here is the default credit memo transaction type.

15. Enter the Receivable Account for transactions with this transaction type. Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the receivable accounts for transactions with these types. Oracle Receivables creates a transaction record using this account so you can transfer to your general ledger and create a journal entry if the Post To GL check box is selected for this transaction type.

16. Enter a Freight Account for transactions with this transaction type. Oracle Receivables uses this information, along with your AutoAccounting definition to determine the freight account for transactions with this transaction type. Oracle Receivables skips this field if the Allow Freight check box is not selected.

17. Enter a Revenue Account for transactions with this transaction type. Oracle Receivables skips this field if the Allow Freight is not selected. Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the revenue account for transactions with this transaction type.

18. Enter a Clearing Account for transactions with this transaction type. Oracle Receivables uses this account to hold any difference between the revenue amount specified for the Revenue Account and the selling price times the quantity for imported invoice lines. Oracle Receivables only uses the Clearing Account if you have enabled this feature for transaction sources that you use for your imported transactions.

19. Enter an Unbilled Receivable Account. When you use the Bill In Arrears invoicing rule, Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the Unbilled Receivable account for transactions with this transaction type.

20. Enter an Unearned Revenue Account. Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the unearned revenue account for transactions with this transaction type. Oracle Receivables only uses this account when your transaction's invoicing rule is Bill In Advance.

21. Enter a Tax Account. Oracle Receivables uses this information along with your AutoAccounting definition to determine the tax account for transactions with this transaction type.

22. Enter the range of dates that this transaction type will be active. The default Start Date is today's date, but you can change it. If you do not enter an End Date, this transaction type will be active indefinitely.

23. Save.

## To set up credit memo transaction types:

1. Select the Receivables responsibility, then navigate to Setup > Transactions > Transaction Types.

2. Enter a transaction name: Credit-OKS (this OKS transaction type is case sensitive and must be entered as: Credit-OKS). Optionally enter a description.

3. Select Credit Memo from the Class LOV.

4. Select the Open Receivable check box. This updates your customer balances each time you create a complete debit memo, chargeback, or on-account credit with this transaction type. Oracle Receivables also includes these transactions in the standard aging and collection processes.

5. Select the Post To GL check box to be able to post transactions with this type to your general ledger.

6. Choose a default Printing Option for transactions with this transaction type. Select Print or Do Not Print. You can override this value when entering transactions.

7. Choose a Transaction Status of Open, Closed, Pending, or Void. Use these statuses to implement your own invoice approval system.

8. Optionally select the Allow Freight check box to allow freight to be entered for transactions with this transaction type

9. Optionally select the Tax Calculation check box to let Oracle Receivables calculate tax for transactions with this transaction type.

10. Choose a Creation Sign. If you are using the Cash Basis accounting method, your transaction's creation sign must be either Positive or Negative. You cannot update this field after you enter transactions with this type.

11. If you want to restrict the direction in which items with this transaction type can be updated by applications entered against them, select the Natural Application Only check box. If you select this box, Oracle Receivables sets Allow Overapplication to No. You cannot update this option after you save this transaction type.

12. Enter an Application Rule Set for this transaction type or select one from the list of values (optional). An Application Rule Set determines the default payment steps when you use the Applications window or AutoLockbox to apply receipts to transactions using this type. If you do not enter a rule set, Oracle Receivables uses the rule set in the System Options window as the default.

13. If you did not select the Natural Application Only check box, choose whether to Allow Overapplication against items with this transaction type by selecting or deselecting this box. If you select this check box, Oracle Receivables sets Natural Application to No and you cannot update it after you save this transaction type. If you use the Cash Basis accounting method, the default value is No and you cannot change it.

14. Enter the range of dates that this transaction type will be active. The default Start Date is today's date, but you can change it. If you do not enter an End Date, this transaction type will be active indefinitely.

15. Enter the Receivable Account for transactions with this transaction type. Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the receivable accounts for transactions with these types. Oracle Receivables creates a transaction record using this account so you can transfer to your general ledger and create a journal entry, if the Post To GL check box is selected for this transaction type.

16. Enter a Freight Account for transactions with this transaction type. Oracle Receivables uses this information, along with your AutoAccounting definition to determine the freight account for transactions with this transaction type. Oracle Receivables skips this field if the Allow Freight check box is not selected.

17. Enter a Revenue Account for transactions with this transaction type. Oracle Receivables skips this field if the Allow Freight is not selected. Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the revenue account for transactions with this transaction type.

18. Enter an Unbilled Receivable Account. When you use the Bill In Arrears invoicing rule, Oracle Receivables uses this information, along with your AutoAccounting definition, to determine the Unbilled Receivable account for transactions with this transaction type.

19. Enter an Unearned Revenue Account. Receivables uses this information, along with your AutoAccounting definition, to determine the unearned revenue account for transactions with this transaction type. Receivables only uses this account when your transaction's invoicing rule is Bill In Advance.

20. Enter a Tax Account. Receivables uses this information along with your AutoAccounting definition to determine the tax account for transactions with this transaction type.

21. Save.

# Setting Up Batch Transaction Sources

Batch sources control the standard transaction type assigned to a transaction and determine whether Oracle Receivables automatically numbers your transactions and transaction batches. Active transaction batch sources appear as list of values choices in the Transactions, Transactions Summary, and Credit Transactions windows.

You can define two types of transaction batch sources:

- **Manual**: Use manual batch sources with transactions that you enter manually in the Transaction and Transactions Summary windows.

- **Imported**: Use imported batch sources to import transactions into Oracle Receivables using AutoInvoice. Batches are automatically numbered with the batch source name - request ID.

You can make a batch source inactive by deselecting the Active check box and saving your work. Oracle Receivables does not display inactive transaction batch sources as list of values choices or let you assign them to your transactions.

**Suggestion**: If you have installed multiple organization support (multi–org), define an imported batch source with the same name in each organization These sources can have the same or different settings. This enables you to import order lines that belong to different organizations in Oracle Order Management into Oracle Receivables.

For more information on setting up transaction batch source, transaction types, refer *Oracle Receivables User Guide*.

## To set up batch transaction sources:

1. Select the Receivables responsibility, then navigate to Setup > Transactions > Sources.

2. Enter OKS_CONTRACTS as the name.

3. Select Imported as the type.

4. Select the Batch Source tab.

5. Enter a description.

6. Enter the range of effective dates for this source. The Start date is the current date, but you can change it. If you do not enter an end date, this transaction batch source will be active indefinitely.

7. Select the Automatic Transaction Numbering check box and enter a Last Number to automatically number new transactions you create using this source. You can use automatic transaction numbering with both Imported and Manual sources.

8. Optionally, select the Copy Document Number to Transaction Number check box to use the same value for both the document number and the transaction number for transactions assigned to this source.

9. Select Invoice-OKS as the Standard Transaction Type for this batch source. When you choose a batch source during transaction entry, this is the default transaction type. You can define new transaction types in the Transaction Types window.

10. Select the AutoInvoice Options tab.

11. Specify how AutoInvoice handles imported transactions that have Invalid Tax Rates. An invalid tax rate is one in which the imported transaction's tax rate does not match its tax code. Select Correct from the Invalid Tax Rate LOV, for AutoInvoice to automatically update the tax rate that you supplied to the one that you defined previously for the tax code. Select Reject if you want AutoInvoice to reject the transaction.

12. Specify how AutoInvoice handles imported transactions with Invalid Lines by selecting either Reject Invoice or Create Invoice from the Invalid Line LOV.

13. Specify how AutoInvoice handles imported transactions that have lines in the Interface Lines table that are in a closed period. To have AutoInvoice automatically adjust the GL dates to the first GL date of the next open or future enterable period, select Adjust from the GL Date in a Closed Period LOV. Alternatively, select Reject to reject these transactions.

14. Optionally, select a Grouping Rule to use for a transaction line from the Grouping LOV. If you do not enter a grouping rule, AutoInvoice uses the following hierarchy to determine which rule to use:

    • The grouping rule specified in the Transaction Sources window for the batch source of the transaction line.

    • The grouping rule specified in the Customer Profile Classes window for the bill–to customer and bill–to site of the transaction line.

    • The grouping rule specified in the Customer Profile Classes window for the bill–to customer of the transaction line.

    • The default grouping rule specified in the System Options window.

15. For AutoInvoice to require that the revenue amount for each transaction line is equal to the selling price times the quantity specified for that line, select the Create Clearing check box. Use this option to distribute revenue on a transaction in an amount that is not equal to the transaction line amount. If you select this check box, AutoInvoice puts any difference between the revenue amount and the selling price times the quantity for a transaction into the AutoInvoice Clearing account that you have defined. Otherwise, AutoInvoice requires that the revenue amount be equal to the selling price times the quantity for all of the transactions it is processing. Define your clearing account in the Automatic Accounting window.

16. Indicate whether sales credits can be entered for transactions using this source by selecting or deselecting the Allow Sales Credit check box. This option and the Require Salesreps option in the System Options window determine whether sales credits are optional or required.

17. Select the Customer Information tab.

18. Select Id for each option to indicate that AutoInvoice validates your customer information for this batch source using an identifier. Choose Value if you use this source to import data from a non–Oracle system.

19. Select the Accounting Information tab.

20. Select Id to indicate how AutoInvoice validates your Invoice and Accounting Rule data for this batch source.

21. Select Id to indicate whether AutoInvoice validates the identifier for this batch source.

22. Select the Derive Date check box to derive the default rule start date and default GL date from the ship date, rule start date, order date and the default date that you supply when you submit AutoInvoice. If Oracle Inventory is installed, this must be selected.

23. Select Id to indicate that AutoInvoice validates your Payment Terms for this batch source using identifiers.

24. Select Percent to indicate that AutoInvoice validates your Revenue Account Allocation data for this batch source.

25. Select the Other Information tab.

26. Select Id to validate other data except for Agreement, Sales Territory, and Related Document.

27. Select the Sales Credit Validation tab.

28. Select Id for the first two options to validate information using identifiers for this batch source.

29. Select Percent to validate sales credits based on percent.

30. Save.

# Setting Up Transaction Flexfield Segments

Transaction flexfields are descriptive flexfields that AutoInvoice uses to uniquely identify transaction lines. Oracle Receivables lets you determine how to build your transaction flexfield structure and what information you want to capture. To define the line-level Transaction Flexfield, query "Line transaction Flexfield" in the title field of the Descriptive Flexfield Segments window and enter the text and segments associated with this transaction flexfield.

The following table shows the values for the transaction flexfields.

| Column Name | Segment Name |
| --- | --- |
| INTERFACE_LINE_ATTRIBUTE1 | CONTRACT_NUMBER |
| INTERFACE_LINE_ATTRIBUTE2 | CONTRACT_MODIFIER |
| INTERFACE_LINE_ATTRIBUTE3 | INSTANCE_NO |
| INTERFACE_LINE_ATTRIBUTE4 | BILLED_FROM |
| INTERFACE_LINE_ATTRIBUTE5 | BILLED_TO |
| INTERFACE_LINE_ATTRIBUTE6 | AMOUNT |
| INTERFACE_LINE_ATTRIBUTE7 | BILLED_FROM_DATE |
| INTERFACE_LINE_ATTRIBUTE8 | START_DATE |
| INTERFACE_LINE_ATTRIBUTE9 | LINE_TYPE |
| INTERFACE_LINE_ATTRIBUTE10 | BILL_INSTALLMENT_NUMBER |

### To set up transaction flexfield segments:

1. Select the System Administrator responsibility, then navigate to Application > FlexField > Descriptive > Segments.

2. Enter/Query Oracle Receivables as the application and Line Transaction Flexfield for the title.

3. Deselect the Freeze Flexfield Definition check box (otherwise you cannot create a new record).

4. Select OKS CONTRACTS in the Context Field Values region.

5. Click Segments to edit the definition.

6. Enter the values listed in the previous table.

> **Note:** These must be entered in upper case.

7. Compile the flexfield by selecting Compile.

8. Select the Freeze Flexfield Definition check box to freeze the definition.

## Pricing Service, Subscription, and Usage Items

There are a number of setups you may perform related to pricing.

- **Adding Items to a Price List**

  For any items that you create, you must add them to a price list. See the Adding Items to a Price List procedure in the *Oracle Advanced Pricing User's Guide*.

- **Creating Price Breaks for Usage Items**

  You can set special price breaks for usage items.

  See Creating Special Price Breaks, *Oracle Service Contracts User Guide*.

- **Setting Up Price Adjustments**

You can adjust prices of service, subscription, and usage lines in your contract by applying adjustments set up by the application administrator. See Entering Price Adjustments, *Oracle Service Contracts User Guide* and Getting Help in Picking Adjustments, *Oracle Service Contracts User Guide*.

# Enabling Multi-Currency Price List Support

You can set up multi-currency price lists. For example, a customer that operates in 40 countries may want three price lists, each with a different currency. Rather than administering 120 price lists (three price lists in each currency), multi-currency price lists allow the customer to maintain three price lists that handle all of the currency conversions.

**Prerequisite**:

Set up multi-currency price lists as specified in the *Multi-Currency Conversion Lists* chapter of the *Oracle Advanced Pricing User's Guide*.

## To enable multi-currency price list support:

1. Set the system profile QP:Multi Currency Installed to **Yes**.

2. Run the concurrent program Update Price Lists with Multi-Currency Conversion Criteria.

3. Set the system profile QP: Multi Currency Usage to **Yes** at Application Level.

    **Note:** After you run the concurrent program, you cannot revert your price lists back to single-currency price lists. Oracle does not support changing the profile QP: Multi Currency Installed back to **No**.

# Enabling Advanced Attribute Sourcing

Oracle Service Contracts integrates with Oracle Advanced Pricing to price services. A Service Contracts request type is seeded in Advanced Pricing to allow Service Contracts to leverage Attribute Management capabilities in Advanced Pricing. See the *Oracle Advanced Pricing User's Guide* for more information on Attribute Management). You can map pricing attributes and qualifiers to the Service Contracts request type in Advanced Pricing. After you complete the mapping, pricing calls from Service Contracts can automatically source the attributes that are used for determining price. For instance, a service provider that employs a price list with special pricing for the company's top five customers can now automatically pass the customer name from Service Contracts to Advanced Pricing as a pricing qualifier.

## To enable advanced attribute sourcing:

1. Select the Pricing Manager responsibility, then navigate to Setup > Attribute Management > Pricing Transactions Entity Associations.

    The Pricing Transaction Entity window appears.

2. Select ORDFUL from the Name LOV.

3. Confirm that the Enabled check box associated with OKS is selected.

4. Save and close.

5. Navigate to Setup > Attribute Management > Context and Attributes.

   The Context Setup window appears.

6. Query for CUSTOMER in Code field.

7. Verify that PARTY_ID Code is available and enabled.

8. Save and close.

9. Navigate to Setup > Attribute Management > Attribute Linking and Mapping.

   The Pricing Transaction - Attribute Linking window appears.

10. Select Order Fulfillment (ORDFUL) from the Pricing Transaction Entity LOV.

11. Select Qualifier Context from the Context Type LOV.

12. Highlight CUSTOMER in the Code field.

13. Click **Link Attributes**.

14. Highlight Party ID in the Code field.

15. Click **Attribute Mapping**.

16. In Application Name field, select Service Contracts.

17. In User Source Type field, select PL/SQL API.

18. In User Value String, enter OKS_QPATTRIB_PVT.G_CONTRACT_HDRREC.PARTY_ ID for Header and OKS_QPATTRIB_PVT.G_CONTRACT_LINREC.PARTY_ID for Line.

19. Save and close.

## Running the Build Attribute Mapping Rules for Pricing Qualifiers and Attributes

After creating pricing qualifiers and attributes you must run the Build Attribute Mapping Rules concurrent program to ensure that the pricing engine uses only the qualifier and pricing attributes used in the price lists, modifiers, and formulas that are relevant to each call, rather than all attribute mapping rules.

You can run this program using the Oracle Pricing Manager responsibility.

For information see the *Oracle Advanced Pricing User's Guide*.

## Defining Sales Territories to Assign Vendor Contacts

You can set up sales territories using Oracle Territory Manager to automatically:

- Assign sales agents as vendor contacts to new contracts created from Oracle Order Management orders.

  You can use sales territories to assign sales agents to a contract. This is useful if your company uses different organizations for selling and renewing service contracts.

- Assign new sales agents to contracts after personnel changes and reorganizations.

  For contracts in the active, signed, and hold statuses, there is a concurrent program that makes the new sales agent assignments and end-dates any existing sales agents

in the contract. For contracts in the entered status, the concurrent program simply replaces the existing sales agents with those derived from the territories.

When the application cannot assign a resource from the territory, for example for new or dormant customers that have no territories set up for them, the application looks for the individual to notify in OKS: Contract Administrator and, if none is set, then in OKC: Contract Approver

**Prerequisite:**

Set up sales territories for Oracle Service Contracts as described in the *Oracle Territory Manager Implementation Guide*. Use the following guidelines during your setup:

## To define sales territories to assign vendor contacts:

1. Select the CRM Administrator responsibility, then navigate to Territory Manager > Territory Administration.

2. Enable Oracle Service Contracts qualifiers:

    1. Choose **Setup Qualifiers** from the Administration menu.

    2. Choose Oracle Service Contracts From the Usage LOV.

    3. Click Find.

        The application displays the list of qualifiers available for use in Oracle Service Contracts:

        - Country

        - Customer Name

        - Customer Range

        - State

    4. Select the respective Enabled check box to enable each qualifier.

    5. Click **Update Qualifiers**.

    6. Close the window.

3. Navigate to the Oracle Service Contracts folder in the territory manager Navigator and create your territories. For each territory:

    1. From the Overview tab, choose Contract Renewal as the Transaction Type.

    2. From the Transaction Qualifiers tab, choose the qualifiers using the Name LOV.

    3. Enter the qualifier values for the territory.

    4. From the Resources Tab, choose the resource using the Name LOV. Enter the name of one resource.

4. Run the concurrent program Generate Territory Packages with the following parameters:

    1. Usage: Oracle Service Contracts

    2. Transaction Type: Contract Renewal

5. To have the application use the sales territories to automatically assign a resource as a vendor contact in contracts created through Oracle Order Management orders:

1. Set the system profile OKS: Territory Sales Person for First Year Contracts to **Derive** at the site level. (By default this system profile is set to Retain, causing the application to copy the sales person on the order to be the vendor contact or type Salesperson on the contract).

2. Confirm the following system profiles are set to the resource to notify if the application cannot assign a sales person to the contract based on the territory setup.

   In this case, the following hierarchy is used to send the notification when the application cannot assign a resource from the territory:

   1. OKS: Contract Administrator

   2. OKC: Contract Approver

6. To reassign sales agents to contracts based on the sales territories, for example, as a result of personnel changes or reorganizations, run the concurrent program Service Contracts Concurrent Program for Reassigning Resources. The profile option value of OKS: Use Territories to Default Sales Person should be set to Yes either at the user level or site level.

   This program reassigns sales agents to in Entered, Active, Signed and Hold statuses. For contracts in the Entered status, the application replaces all vendor contacts of role Salesperson with the resource supplied by the territory. For contracts in the Active, Signed, and Hold statuses, the application end-dates (system date -1) the current vendor contact(s) of role Salesperson and appends the resource supplied by the territory. The application sets the new sales person's start date to the system date. You can restrict the scope of the concurrent program's assignment by entering any of the following parameters:

   • Contract Number

   • Contract Status

   • Organization

   • Sales Person (sales person within that organization)

        **Note:** If you do not enter any of these parameters, the program reassigns sales agents to all contracts.

# Executing Billing Through Concurrent Programs

The procedures for executing service contract billing involve the following concurrent requests:

   • **Service Contracts Main Billing**: According to pricing and billing schedule in the contract, running this request generates transactions in the Service Contracts billing tables and the Oracle Receivables interface table.

   • **Autoinvoice Import Program**: Running this request picks records from the Oracle Receivables interface table and process billing.

   • **Service Contracts Fetch Receivables Info for Billing**: Running this request fetches the invoice number and tax from Oracle Receivables and updates the contract billing history.

These processes should be set to run on a periodic basis, the frequency of which is dependent on your billing policy. You may find it useful to run these requests as a

request set. You may also consider running the QA process within this set to ensure that information in the contract is still accurate prior to running the billing process. See Executing Billing Through Concurrent Programs, *Oracle Service Contracts User Guide*.

# 4

# Setting Up Contract Events

This chapter covers the following topics:

- Overview of Events
- Defining Condition Templates
- Using Query Conditions
- Reviewing Errors from Asynchronous Processing
- Using the Events Controller
- Enabling Service Request Creation based on Contract Events
- Starting the Event and Outcome Listeners
- Starting the Workflow Background Process
- Running the Date Assembler

## Overview of Events

You can set up events to execute a specific outcome if certain conditions are met. For example, you could set up contracts of a specific type to renew 90 days prior to expiration. An event can be action based, such as contract signed, or date based such as renew contract 90 days prior to expiration.

In order to set up an event you must have the following:

- **Action** (an action can be action-based or date-based).
- **Condition**
- **Outcome**

After you define an action the following occurs:

- If the condition is true, the system executes the outcome. If the outcome is successful, the system sends a success notification. If the outcome fails, the system sends failure notification.
- If the condition is false, the outcome is not executed, and a notification is not sent.

   **Note:** This group of procedures covers the Contract Events menu assigned to the Service Contracts Manager responsibility, except for the following:

   The procedure for creating processes is covered in Defining a Process, page 2-5.

The procedure for creating independent conditions is covered in Creating Independent Conditions for the Renewal, page 5-4

> **Caution:** Custom events and actions are not supported. If you customize events, you are doing so at your own risk. Oracle Support cannot help you create or debug actions, functions, or outcomes. If you encounter problems with customizations and wish to obtain support, you must recreate the problem using standard Oracle objects.

# Defining Condition Templates

You can use condition templates to define multiple samples of conditions. The templates can then be used to define independent conditions or conditions attached to a contract.

**Prerequisite**

Create an outcome.

## To define a condition template:

1. Select the Service Contracts Manager responsibility, then navigate to Contract Events > Define Condition Template.

2. Enter a name and a description.

3. Check the Create a Task check box, to create a task for the Schedule tab in the execution overview.

4. Enter the task owner.

5. Select Action or Date.

6. To create an action condition, select an action.

7. To create a date condition, enter the date information.

8. Build your condition lines.

    > **Note:** When using the LIKE operator, do not use quotes. For example, when building a condition, where contract_number like %-2000%, your right value must be %-2000 and not "%-2000".

9. To enter fixed values for parameters for a function, click **Parameters** and enter the information

10. Click **Show Condition** to display all condition lines and check their syntactical validity.

    If the condition has validated successfully, then Condition Valid is automatically selected.

11. Select **Outcomes**.

12. To assign fixed values or action attribute values to the outcome parameters, click **Parameters** and enter the information.

13. Select **Notifications**.

14. Select a Success Notifier from the LOV.

15. Select a Failure Notifier from the LOV.

16. Save.

## Using Query Conditions

You can use Query Conditions to search all conditions, including independent conditions, condition templates, and instances created by condition templates.

### To use query conditions:

1. Select the Service Contracts responsibility, then navigate to Contract Events > Query Condition.

2. From the View menu, select **Query By Example** and choose **Enter**.

   You can query any of the fields that are blue. These are the only fields that are active.

3. Enter or select your criteria. You can use wildcards such as **%**.

4. From the View menu, select **Query By Example** and choose **Run**.

   The window is populated with the conditions that match the criteria that you entered.

5. Select the **Page Down** button on your keyboard to see additional conditions that match your search criteria.

## Reviewing Errors from Asynchronous Processing

The Events Component is dependent on asynchronous processing. It uses Advanced Queuing System. This system is transparent to the user. The messages processed may encounter exceptions and as a result the messages are rolled back and will be reprocessed again at a specified time. There is a need to record any exceptions encountered. Hence whenever an exception is encountered, the Evaluator writes the details of the exceptions to a table OKC_AQERRORS and the error stack to OKC_AQMSGSTACKS tables. The Asynchronous Errors window displays all error messages of advanced queueing processes that have failed. This window displays Queue Name, Message ID, and the Message Text. This window is generally used by your support representative to resolve issues.

### To review errors from asynchronous processing:

1. Select the Service Contracts Manager responsibility, then navigate to Contract Events > Query Asynchronous Error Messages.

   The following provides a description of the attributes:

   - **Source Name**: Provides source name.

   - **Begin Date**: Provides begin date.

   - **Queue Name**: Identifies name of the Advanced Queue.

   - **Message ID**: Provides message ID.

   - **Retry Count**: Displays number of times the queue tried to dequeue the message.

   - **Queue Contents**: Displays queue content.

- **Message Text**: Displays error message text.

# Using the Events Controller

The Events Controller is a tool for the system administrator to debug any environment problems related to the queues. This is a query only form which gives the system administrator the provision to check common problems related to advanced queues, for example whether the listeners are running, queue objects are valid, or if there are any errors thrown by the queues. This form provides a more convenient method of viewing more information from one location. The Events Controller has a start and stop option for queues and concurrent programs. The queues must always be enabled (started). The listeners will be running as scheduled and will show an error if the queues are disabled (stopped). The start and stop option is an extra DBA provision provided to the system administrator and can also be used when debugging certain queue problems. If the Events Controller is used to stop queues, it will automatically terminate any Listener for Events/Outcome concurrent programs. You can start the queues again and restart the listeners. This will not cause any data loss from the queues.

> **Note:** The Asynchronous window form only displays queue errors.

## To use the events controller:

1. Select the Service Contracts Manager responsibility, then navigate to > Contract Events > Events Controller.

2. Select the Background tab.

   This tab displays the events and outcome listeners that are running in the background and the workflow engine process. In the event of an error, this tab provides you with the ability to access the Errors window which describes the error and the cause.

   The following provides a description of the attributes:

   - **Request ID**: Provides ID of the submitted concurrent program.

   - **Listener/Program**: Identifies the name of the event, outcome, or workflow background process.

   - **Phase**: Provides update, such as Pending, Running, or Completed.

   - **Status**: Displays Normal or Error. When a program has been completed with an Error status, you can click Diagnostics to view the error details returned by the concurrent program.

   - **Request Date**: Provides date submitted.

   - **Requestor**: Identifies user who submitted the request.

3. Select the Queue Content tab.

   This tab displays the Queue Messages and Queue Errors that have occurred. If an event is launched (such as contract signed) you can expect to see the queue content within this tab. If there is nothing in the queue, then the Queue Status tab is used to check the queue status and also to check if all queue objects are valid.

   The following provides a description of the attributes:

   - **Queue Name**: Identifies name of the Advanced Queue.

   - **Correlation**: Identifies the correlation Id such as KSIGN and KEXTEND.

- **Consumer**: Describes the program which parses and consumes the message in the queue.

- **Enqueue Time**: Shows when message was placed in queue.

- **Queue Content**: Displays queue content.

- **Source Name**: Shows source of error.

- **Retrys**: Displays number of times the queue tried to dequeue the message.

- **Begin Date**: Shows date when error occurred.

- **Error Details** (Button): Enables you to view the details of queue errors.

4. Select the Queue Status tab.

   This tab displays the queue names, queue objects, and their status (valid or invalid).

   The following provides a description of the attributes:

   - **Queue Name**: Identifies name of the Advanced Queue.

   - **Queue Table**: Shows table name.

   - **Queue Type**: Displays the queue type such as Normal and Exception.

   - **Enqueue Enabled**: Yes/No.

   - **Dequeue Enabled**: Yes/No.

   - **Start Queue/Stop Queue** (Buttons): Enables you to stop or start queue.

   - **Queue Objects**: Names all the advance queue objects.

   - **Object Name**: Provides name of object.

   - **Object Type**: Displays type of object such as Table and View.

   - **Status Type**: Shows whether object is valid or invalid.

   - **Last Modified**: Displays date last modified.

   - **Details** (Button): Shows details for rules and subscriber objects.

# Enabling Service Request Creation based on Contract Events

You can set up Oracle Contracts Service to automatically create a service request. This occurs when an event which is attached to a service line, satisfies the condition. When a condition template is created for a service, the OKSEVENT-CREATE_SR process can be used as an outcome. After you capture a counter for the contract line and the condition is satisfied, the system creates a service request and notifies the user defined in profile OKS: Service Request Creator.

For example, if Vision Corporation has a contract with Business World to service a copier, they can set up an event for the service, where once the counter for the copier reaches 10000, a service request is created and a notification is sent to the service request creator.

**Prerequisites:**

- Set the following profile options:

  - Service: Default Service Request Type

  - Service: Default Service Request Status

- Service: Default Service Request Severity

- OKS: Service Request Creator (set it at site level)

    **Note:** If the profile Service: Use SR Type - Responsibility Mapping is set to **Yes**, then the responsibility mapped to default service request type must also be added to the application user as setup in profile OKS: Service Request Creator. The default service request type is setup in the profile Service: Default Service Request Type. For Service application related setup, refer to *Oracle TeleService Implementation Guide*.

- Define a contact for the Customer Party type of the Contract. Refer to Adding Contact Points for a Customer and Adding Contact Points for a Customer site in the *Oracle TeleService User Guide*.

- Schedule the following concurrent programs:

    - Listener for Events Queue

    - Listener for Outcome Queue

    - Workflow Background process for workflow item type Contract Alert

- Define a Condition for the Service item

    When defining the Condition for the service item, make sure to select Counter group updated as an Action and attach the OKSEVENT-CREATE_SR process as an outcome. Also, select the K_LINE_ID as the parameter.

    **Note:** If you requery the parameter, it appears as SOURCE OBJECT ID, not K_LINE_ID.

## To enable service request creation based on contract events:

1. Select the Service Contracts responsibility, then navigate to Contract Events > Define Condition Template.

2. Enter a name.

3. In the condition type region, select Counter group Updated from the Action LOV.

4. Select the Counter tab, and select Service from the Item/Product list.

5. From the Description LOV, select a service.

6. In the Left Counter Value field, select a counter.

7. Select an operator and enter a value in the Right Counter Value field.

8. From the Outcomes secondary tab, select OKSEVENT-CREATE_SR from the Outcomes LOV.

9. Select the Notifications secondary tab and choose the individual(s) to send the Success Notifiers and the Failure Notifiers.

10. Select **Parameters**.

    The Parameters window appears, the value K_LINE_ID is the default.

    Click **OK**.

11. Select an action attribute for K_LINE_ID and click **OK**.

12. Save.

13. From the Navigator, select Launch Contracts and create a Service Agreement.

    Make sure the party role is Customer and that you select the contact you setup in the Contact Center. Enter a service line with the service you selected in the condition template.

14. Select the Lines tab and Event subtab.

    You should see the condition that you created in the Events region.

15. Select the Lines tab and Counters subtab.

16. From the Actions menu select **Counter Capture**. Enter the counter reading to trigger the event.

    After the counter is captured for the contract line, and if the condition evaluates to true, the condition creates a service request. Subsequently, the service request creator gets the notification of the service request creation that can be viewed from the Inbox the Launch Contracts window.

## Starting the Event and Outcome Listeners

For automatic renewals and any other events to work there two concurrent programs that must be scheduled to run. These are:

- Listener for Events Queue
- Listener for Outcome Queue

Use the concurrent request set to start both programs. After submitting the Listener request set, each concurrent program should end with a normal completion status.

These listeners impact automatic renewals as follows:

When the Date Assembler runs a message is written to the event queue, OKC_AQ_ EV_QUEUE. The Event Listener picks up the message, evaluates the condition, and for contracts that meet the condition, a message is written to the outcome queue, OKC_AQ_QUEUE, which tells the listener to perform the appropriate outcome, such as renew the contract.

If the database or the concurrent managers have been restarted, verify these two processes are restarted.

There are two profile options to set the work load of the listeners: OKC: Event Listener Iterations and OKC: Outcome Listener Iterations. These profiles control the number of messages the listeners will dequeue for every run. The profiles can be set at the site and application levels.

If the above profiles are not set, OKC Event and Outcome listeners will keep checking the queue messages all every time even if there is no message in the queue. This check is CPU extensive and uses memory resources.

If these profiles are set, the listeners stop checking the queue if there are no messages and resume checking the next time listeners are scheduled to run (based on the schedule parameter you set when running the concurrent request for Event and Outcome Listeners).

The default load value is set to 100, which is recommended. This number can be changed to suit the Advanced Queue load. The minimum should be greater than 1 and the maximum should not exceed 1,000.

This number should be set to a higher value for systems that have larger work loads. Setting a higher value for the profile options, results in longer running and more resource CPU intensive listener processes.

Submit requests to start both listeners using a Periodic schedule. The suggested period is every three to five minutes. Failure to run the Listeners on a regular schedule will severely limit functionality in Contracts and other applications using the Contracts Events System. It will also cause Events system Advanced Queue to become back logged.

Listeners run and complete rather than running indefinitely. In order to process messages in the queue, you must set the listeners to run on a regular schedule. Increase or decrease the interval/period at which the Listener process run according to your system load. The sleep parameter seen when submitting a request is now redundant. You should balance these processes with the iterations and request schedule. Use both these methods to fine tune your listener processes.

> **Note:** You should stop these concurrent requests if you are applying product upgrades or performing system maintenance. Terminating the concurrent request will not stop the listener process. For a detailed description on stopping these requests see Doc ID 183558.1 on Oracle*Metalink*.

> The following procedure describes the Events Controller, which has a role in starting and stopping listeners. See Using the Events Controller, page 4-4.

## To start the event listener:

1.  Select the Service Contracts Manager responsibility, then navigate to Control > Request > Run.

    The Submit a New Request window appears.

2.  Select Single Request and click **OK**.

    The Submit Request window appears.

3.  From the Name LOV select **Listener for Events Queue**.

    The Parameters window appears.

4.  Enter a value for the following parameters:

    *   Wait

    Click **OK**.

5.  Click **Schedule**.

    The Schedule window appears.

6.  Select Periodically from Run the Job block.

7.  From the Re-run Every field, enter 5 and select Minutes.

8.  Click **OK**.

9.  Click **Submit**.

Make a note of the Request ID.

## To start the outcome listener:

1.  Select the Service Contracts Manager responsibility, then navigate to Control > Request > Run.

    The Submit a New Request window appears.

2.  Select Single Request and click **OK**.

    The Submit Request window appears.

3.  From the Name LOV select **Listener for Outcome Queue**.

    The Parameters window appears.

4.  Enter a value for the following parameters:

    *   Wait

    Click **OK**.

5.  Click **Schedule**.

    The Schedule window appears.

6.  Select Periodically from Run the Job block.

7.  From the Re-run Every field, enter 5 and select Minutes.

8.  Click **OK**.

9.  Click **Submit**.

    Make a note of the Request ID.

# Starting the Workflow Background Process

If you are using Oracle Workflow with other applications, the program Workflow Background Processes is most likely running in recurring intervals. If the Workflow Background Processes is not scheduled to run, the system administrator must start this process to run regularly, such as every five minutes.

## To start the Workflow Background Process:

1.  Select the Service Contracts Manager responsibility, then navigate to Control > Request > Run.

    The Submit a New Request window appears.

2.  Select Single Request and click **OK**.

    The Submit Request window appears.

3.  From the Name LOV select **Workflow Background Process**.

    The Parameters window appears.

4.  Select a value for the following parameters:

    *   Contract Alert from the Item Type LOV.

    *   Yes from the Process Deferred LOV.

- Yes from the Process Timeout LOV.

Click **OK**.

5. Click **Schedule**.

The Schedule window appears.

6. Select Periodically from Run the Job block.

7. From the Re-run Every field, enter 5 and select Minutes.

8. Click **OK**.

9. Click **Submit**.

Make a note of the Request ID.

# Running the Date Assembler

You should run the Date Assembler concurrent program once every day, preferably later in the evening. This program automatically renews contracts that are eligible for renewal.

The Date Assembler checks if the end date of the contract is between the last_rundate + variance and sysdate + variance. So, if the last_rundate is 4-Jan-2005 and sysdate is 5-Jan-2005 and variance is 90 days, then Date Assembler renews all contracts expiring on 5-Apr-2005.

## To run the Date Assembler:

1. Select the Service Contracts Manager responsibility, then navigate to Control > Request > Run.

The Submit a New Request window appears.

2. Select Single Request and click **OK**.

The Submit Request window appears.

3. From the Name LOV select **Date Assembler**.

4. Select **Submit**.

# 5

# Setting Up Automatic Renewals

This chapter covers the following topics:

- Overview of Automatic Renewals Setup
- Creating Process Definitions for Renewals (Outcomes List of Values)
- Creating Independent Conditions for the Renewal
- Confirming Renewal Defaults
- Starting the Event and Outcome Listeners for Renewals
- Starting the Workflow Background Process for Renewals
- Running the Date Assembler for Renewals
- Confirming the Automatic Renewal
- Creating and Enabling Templates for E-Mails to Customers
- Troubleshooting the Automatic Renewal Process

## Overview of Automatic Renewals Setup

You can set up the application so that users can specify a period of time prior to expiration for renewing a contract. You can define actions that are related to a certain point in time, such as the date when contracts are about to expire, or a customer's anniversary date within the specified period.

Users can specify how a contract is renewed. For information on contract renewals, see Understanding Contract Renewals, *Oracle Service Contracts User Guide*.

To set up automatic renewals you must:

- Create a process definition
- Create an independent condition
- Start event and outcome listeners
- Start Workflow Background Process
- Run Date Assembler
- Confirm the event listener picked up the message
- Confirm the outcome listener renews the contract

If the contract is renewed successfully, the application notifies the resource attached to the contract, such as the sale representative when the Workflow Background Process next runs.

> **Note:** Service Contracts customers can review and accept their renewal contracts from a self-service web page. The Electronic Renewal process in Service Contracts places renewed contracts in Entered status and sends a notification to both the sales representative and the customer. The customer receives an e-mail notification with a username and password, as well as a link to a web page. Customers can review the quote and accept, reject, or request changes to the contract renewal from this web page.

> To support this process, an FND_USER record is created using the customer's Quote-To Contact e-mail address and a randomly generated password. Service Contracts Electronic Renewals responsibility is assigned to the user. This is a required step to allow access to the external facing Customer Acceptance Java Server Pages. The user is associated with STANDARD security group and a profile option Self Service Personal Home Page mode is set to Personal Home Page for this user.

> The following profile options must be set up in order to use the e-mail Quote functionality:
>
> - OKS: Wallet Path
> - OKS SMTP Host
> - OKS: SMTP Port

# Creating Process Definitions for Renewals (Outcomes List of Values)

You can set up Process Definitions to define the seeded or custom processes that should be used for certain situations. For Contract Renewals, the following stored PL/SQL procedure can be used to renew contracts: OKS_EVTREN_PUB.RENEW.

The following procedure provides an example of how you can create a process definition for the Outcomes LOV within the Condition form.

> **Note:** You can also refer to the generic procedure, Defining a Process, page 2-5.

## To create a process definition for Renewals (Outcomes List of Values):

1. Select the Service Contracts Responsibility, then navigate to Setup > Contract > Process Definition.

2. Enter a name for your renewal process.

3. Select Outcome from the Purpose LOV.

4. Select PLSQL from the Type LOV.

5. Enter OKS_EVTREN_PUB in the Package field.

6. Enter RENEW in the Procedure field.

7. Enter the values in Parameter region.

| Name | Data Type | Default Value | Description |
|---|---|---|---|
| P_API_VERSION | NUMBER | 1.0 | API version for OKS_EVTREN_PUB. RENEW |
| P_CGP_NEW_ID | NUMBER | OKC_API.G_MISS_ NUM | CGP_NEW_ID |
| P_CGP_RENEW_ID | NUMBER | OKC_API.G_MISS_ NUM | CGP_RENEW_ID |
| P_CONTRACT_ID | NUMBER | -- | Contract ID |
| P_CONTRACT_ MODIFIER | CHAR | OKC_API.G_MISS_ CHAR | Contract Modifier |
| P_CONTRACT_ NUMBER | CHAR | OKC_API.G_MISS_ CHAR | Contract Number |
| P_CONTRACT_ VERSION | CHAR | OKC_API.G_MISS_ CHAR | Contract Version |
| P_DURATION | NUMBER | -- | Duration |
| P_END_DATE | DATE | OKC_API.G_MISS_ DATE | End Date |
| P_FUNCTION_ NAME | CHAR | OKC_API.G_MISS_ CHAR | Function Name |
| P_MARKUP_ PERCENT | NUMBER | OKC_API.G_MISS_ NUM | Markup Percent |
| P_NEW_ CONTRACT_ NUMBER | CHAR | OKC_API.G_MISS_ CHAR | New Contract Number |
| P_NEW_ CONTRACT_MODIF IER | CHAR | OKC_API.G_MISS_ CHAR | New Contract Modifier |
| P_OBJECT_VERS ION_NUMBER | NUMBER | OKC_API.G_MISS_ NUM | Object Version Number |
| P_ORIG_START_ DATE | DATE | OKC_API.G_MISS_ DATE | Original Start Date |
| P_ORIG_END_DATE | DATE | OKC_API.G_MISS_ DATE | Original End Date |
| P_PDF_ID | NUMBER | OKC_API.G_MISS_ NUM | PDF_ID |
| P_PRICE_LIST_ID1 | CHAR | OKC_API.G_MISS_ CHAR | Price List ID1 |
| P_PRICE_LIST_ID2 | CHAR | OKC_API.G_MISS_ CHAR | Price List ID2 |
| P_QCL_ID | NUMBER | OKC_API.G_MISS_ NUM | QCL_ID |

| | | | |
|---|---|---|---|
| P_PO_REQUIRED_YN | CHAR | OKC_API.G_MISS_CHAR | PO Required (Y/N) |
| P_RENEWAL_TYPE | CHAR | OKC_API.G_MISS_CHAR | Renewal Type |
| P_RENEWAL_PRICING_TYPE | CHAR | OKC_API.G_MISS_CHAR | Renewal Pricing Type |
| P_RLE_CODE | CHAR | OKC_API.G_MISS_CHAR | RLE CODE |
| P_START_DATE | DATE | OKC_API.G_MISS_DATE | Start Date |
| P_UOM_CODE | CHAR | -- | Unit Of Measure Code |

8. Select the Required check box for only the following parameters:
   - P_CONTRACT_ID
   - P_CONTRACT_NUMBER
   - P_CONTRACT_MODIFIER
   - P_DURATION
   - P_UOM_CODE
9. Save.

# Creating Independent Conditions for the Renewal

You can set up independent conditions that must be met before an outcome is processed for an event.

The Last Occurrence field of the Independent Condition is updated when the Date Assembler successfully runs, which means that not only does Date Assembler find eligible contracts, but it also completes the event, such as a renewal. For example, if you define a condition specifying that a renewed contract should be created 90 days before contract expiration, then the Last Occurrence should be the current date + 90.

The following procedure provides an example of how you can set up an independent condition that you can used to autorenew contracts.

### To create an independent condition for the renewal:

1. Select the Service Contracts Manager responsibility, then navigate to Contract Events > Define Independent Condition.
2. Enter a name for the independent condition, such as Auto Renewal for Contracts.
3. Enter the effective dates.
4. In the Condition Type region:
   1. Select the Date radio button
   2. Enter a value in the Number of Days field, such as 90.

3. Select Before from the Before/After list.

4. Select a value from the Date LOV, such as Contract Expiration.

5. From the Expression tab enter the expressions.

    For example:

    **Example**

    10 (Contract Class = Service Agreement) AND

    20 (Contract Status Code = Active) AND

    30 (Contract Category = Service Agreement)

6. From the Outcomes tab, select the process definition for renewals from the Outcomes LOV.

7. Verify the Enabled check box is selected.

8. Select Parameters.

    The Parameter window appears.

9. Enter actions for all parameters that require an action or value:

    1. Highlight the P_CONTRACT_ID, P_UOM_CODE, P_DURATION, P_ CONTRACT_MODIFIER, P_CONTRACT_NUMBER by holding down the Ctrl key and left clicking on the values.

    2. After all of the values are highlighted, right click and choose **Populate selected rows in PARAMETERS** from the menu.

    3. In the Parameters window enter the following values:

    | Parameter | Data Type | Account Attribute | Value |
    |---|---|---|---|
    | P_CONTRACT_ID | NUMBER | CONTRACT_ID | -- |
    | P_CONTRACT_ MODIFIER | CHAR | CONTRACT NBR MODIFIER | -- |
    | P_CONTRACT_ NUMBER | CHAR | CONTRACT NUMBER | -- |
    | P_DURATION | NUMBER | -- | 12 |
    | P_UOM_CODE | CHAR | -- | MTH |

    > **Note:** The P_UOM_CODE value must equal the UOM code as defined in Oracle Inventory. From the Service Contracts Manager responsibility, navigate to Setup > Units of Measure > UOM Classes. Query Class = Time. Select the Units of Measure button. Each entry has a name, a UOM and a description. Place the value found in UOM in this field.

    4. Click **OK**.

10. From the Notifications tab:

    - Select a name from the Success Notifier LOV.

    - Select a name from the Failure Notification LOV.

11. Save.

# Confirming Renewal Defaults

Renewal and pricing attributes are maintained in the individual contracts and apply to the contract categories Warranties and Extended Warranties, Service Agreements and Subscriptions. Renewal Defaults for Warranties and Extended Warranties contracts are set at the contract level in the Contract Details form within Order Management. See Specifying Renewal and Other Details for a New Contract Created for Your Order, *Oracle Service Contracts User Guide*. If renewal options are not specified in contract, the renewal and pricing attributes are retrieved from the renewal event or from the global defaults. Attributes are retrieved from the renewal event or from the global defaults in the following order of precedence:

**Event**: Events can be used for Contract Renewal. By entering values for the parameters in either the Outcome or the Independent Condition window you can default contract information when using Events to automatically renew contracts.

**Party**: Defaults can be set for a Party in the Renewal Rule Defaults window.

**Organization**: Defaults can be set for organizations in your company. Organizations defined as Operating Units are eligible to be selected for renewal defaults.

**Global**: Global Defaults are located at the top of the Renewal Rule Defaults window. If default values have not been specified in earlier checks (Contract, Event, Party or Organization), the defaults defined here are used.

You can set up contracts to renew using different renewal types. For an understanding of renewal types, see Understanding Contract Renewals, *Oracle Service Contracts User Guide*.

The following procedures explain how to specify renewal rules:

**Default Renewal Rules using Global Contract Defaults**

You can specify default renewal rules using the Global Contracts Default window. See Entering Default Renewal Rules, *Oracle Service Contracts User Guide*.

**Default Renewal Rules at the Contract Header Level**

You can setup automatic defaults for contract renewals from the Renewals subtab of the header. See Specifying How a Contract Will Be Renewed., *Oracle Service Contracts User Guide*

**Default Renewal Rules at the Line and Subline Level**

You can also set renewal defaults at the line and subline level. See Specifying How a Contract Line is to Be Renewed, *Oracle Service Contracts User Guide*.

# Starting the Event and Outcome Listeners for Renewals

This procedure is covered in the Events group of topics.

See Starting the Event and Outcome Listeners, page 4-7.

# Starting the Workflow Background Process for Renewals

This procedure is covered in the Events group of topics.

See Starting the Workflow Background Process, page 4-9.

## Running the Date Assembler for Renewals

This procedure is covered in the Events group of topics.

See Running the Date Assembler, page 4-10.

## Confirming the Automatic Renewal

To find out if the contracts were renewed successfully you can check for renewed contracts in Contract Navigator.

You can also check the date in the Last Occurrence field of the Independent Condition window. This date should be updated whenever the Date Assembler runs and is successful in picking up and renewing contracts. For example, if you defined an independent condition specifying that a contract should be picked up and renewed if its expiration date is 45 days before contract expiration and this is successful, then the Last Occurrence is the current date + 45.

## Creating and Enabling Templates for E-Mails to Customers

This topic is covered in About Templates for E-Mails to Customers, *Oracle Service Contracts User Guide*

## Troubleshooting the Automatic Renewal Process

Verify that you have confirmed the procedures covered in the preceding group of procedures. If you are still experiencing problems, the following may help you to troubleshoot issues with the automatic renewal of contracts.

**Validate that Date Assembler Found Eligible Contracts**

The Date Assembler evaluates each Independent Condition. Contracts that satisfy the criteria defined in the condition appear in the output log file.

From the following sample log, we the CREATE RENEWAL EVENT condition was executed. Two contracts met the conditions defined in this condition.

- Contracts Nominated for Expiry Date Action Assembler

- Condition -CREATE RENEWAL EVENT is processed

- Contract -10096 is processed

- Contract -10097 is processed

Possible causes for the Date Assembler to fail to find eligible contracts include:

- Independent Condition has an end-date prior to the current date

- Independent Condition is invalid (verify that the Condition Valid check box is enabled)

- Desired contract does not meet defined condition

**Confirm Error Messages**

From the Service Contracts Manager responsibility, you can navigate to Contract Events > Query Asynchonous Error Message. If this does not return errors, you can use the following details to help you troubleshoot:

- Messages are placed in two tables OKC_AQERRORS and OKC_AQMSGSTACKS.

  You can find the contract id from OKC_AQERRORS and specific error messages from OKC_AQMSGSTACKS.

- The following conditions prevent records from being placed in the error tables:

  - The Listener for Outcomes did not run.

  - The Independent Condition did not fire.

  - A problem with the (Success or Failure) notification prevented processing from completing correctly.

- You can use the following query to confirm that errors were detected by the Listener for Outcomes or Listener for Events:

  ```
  SELECT SOURCE_NAME, Q_NAME, RETRY_COUNT, QUEUE_CONTENTS, CREATION
  _DATE FROM OKC_AQERRORS
  ```

- You can use the following query to view the specific errors detected for each contract:

  ```
  SELECT AQE_ID, MSG_SEQ_NO, MESSAGE_TEXT FROM OKC_AQMSGSTACKS
  ```

- You can write additional queries by matching the column ID from OKC_AQERRORS to the column AQE_ID from OKC_AQMSGSTACKS. For example:

  ```
  SELECT QUEUE_CONTENTS, MESSAGE_TEXT FROM OKC_AQMSGSTACKS MSG, OKC
  _AQERRORS ERR WHERE ERR.ID = MSG.AQE_ID
  ```
  If the query returns values, you can conclude that the Advanced Queuing worked at least one time.

**Validate Queues Exist**

The Date Assembler places messages in the queue. The Listener for Outcomes and Listener for Events dequeues these messages.

You can use the following queries to establish that the queues are defined correctly and that the Advanced Queuing process is functioning correctly:

1. ```
   SELECT NAME, QUEUE_TABLE, QID, QUEUE_TYPE, ENQUEUE_ENABLED, DEQUE
   UE_ENABLED FROM DBA_QUEUES WHERE QUEUE_TABLE = 'OKC_AQ_EV_TAB'
   ```
   This query should return three rows. Take note of the values for QID as they are needed in the second query.

   Expected queue names and queue types are as follows:

   OKC_AQ_EV_QUEUE (NORMAL_QUEUE) OKC_AQ_OC_QUEUE (NORMAL_QUEUE) AQ$_OKC_AQ_EV_TAB_E (EXCEPTION_QUEUE)

   The results of the query indicate whether records are in the queue (column ENQUEUE = YES), and whether records have been processed by the Listener for Outcomes and Listener for Events, (column ENQUEUE = YES).

   ```
   NAME           QUEUE_TABLE   QID    QUEUE_TYPE      ENQUEUE   DEQUEUE
   OKC_AQ_EV_QUEUE   OKC_AQ_EV_TAB 108816 NORMAL_QUEUE   YES       YE
   S
   OKC_AQ_OC_QUEUE   OKC_AQ_EV_TAB 108817 NORMAL_QUEUE   YES       YE
   S
   AQ$_OKC_AQ_EV_TAB_E OKC_AQ_EV_TAB 108815 EXCEPTION_QUEUE NO
     YES
   ```

2. ```
   SELECT * FROM V$AQ WHERE QID IN (queue id #1, queue id # 2, queue
    id #3)
   ```

> **Note:** Replace queue id #1, queue id #2, and queue id #3 with the QID values returned in the first query.

**Check Status of Messages in Queue**

You can use this query to find the number of non-expired messages in the queues:

```
SELECT MSG_STATE,CORR_ID,CONSUMER_NAME,COUNT(*) FROM OKC.AQ$OKC_A
Q_EV_TAB WHERE MSG_STATE <> 'EXPIRED' GROUP BY MSG_STATE,CORR_ID,C
ONSUMER_NAME ORDER BY 1,2
```

You can use this query to find the number of times the messages were retried:

```
SELECT CORR_ID,RETRY_COUNT,COUNT(*) FROM OKC.AQ$OKC_AQ_EV_TAB WHE
RE RETRY_COUNT >=1 AND MSG_STATE = 'READY' GROUP BY CORR_ID,RETRY_
COUNT ORDER BY 1,2
```

**Check Status of Queue**

You can run this query to verify that Advanced Queuing is working for the queue OKC_AQ_EV_TAB:

```
SELECT STATE FROM OKC_AQ_EV_TAB
```

*AQ Message States*:

Queue tables are created using the procedure DBMS_AQADM.CREATE_QUEUE_TABLE and can be listed using the dictionary view DBA_QUEUE_TABLES. <queue_table>.state is a NUMBER column containing value in {0,1,2,3}.

- **0 'READY'**: = Message is ready for dequeue.

- **1 'WAITING'**: = The delay specified by message_properties_t.delay while executing dbms_aq.enqueue has not been reached.

- **2 'RETAINED'**: = The message has been successfully processed (dequeued) but will remain in the queue until the retention_time specified for the queue while executing dbms_aqadm.create_queue has been reached.

- **3 'EXPIRED'**: = The message was not successfully processed (dequeued) in either: the time specified by message_properties_t.expiration while executing dbms_aq.enqueue or; the maximum number of dequeue attempts (max_retries) specified for the queue while executing dbms_aqadm.create_queue.

  > **Note:** Consult with a qualified DBA if the query returns no rows, or if the result of the query is 3. If the query returns no rows, this may indicate that your Success and Failure Notifiers have not been defined for the Independent Condition. Define these values and retest.

**Check Contract Alert Workflow**

Use the System Administrator login and responsibility and navigate to Workflow > Find Processes.

1. Enter a Query for Item Type, Contract Alert.

2. Find one of your processes (probably one of the more recent processes).

3. Select View Diagram to see where the process failed.

   The workflow process branches in five directions from the initial process, Version. If it branches through leg 5, Generic Function, and the Loop Prompt and Loop Counter are highlighted in green, this is an indication that the notification process

failed. This failure could prevent error messages from being placed in the table OKC_AQERRORS.

4. Verify that the Success and Failure notifiers are identified in the Independent Condition.

5. More detailed information can be obtained by selecting the option Advanced Options before selecting View Diagram. Enable all of the Activity Type check boxes, then select the Filter Activities button and select View Diagram.

**Confirm Packages and Procedures**

The following packages are used by the Date Assembler, Listener for Outcomes and Queuing. You should be prepared to provide the package version numbers if you continue to experience problems and contact Oracle Support.

| Package | Filename | Function |
| --- | --- | --- |
| OKC_AQ_PVT | OKCRAQB.pls | Listener for Outcomes executable |
| | | Listener for Events executable |
| OKC_AQ_PUB | OKCPAQB.pls | Standard API for advanced queuing |
| OKC_DATE_ASSEMBLER_PUB | OKCPDASB.pls | Date Assembler program |
| OKC_CONDITIONS_PUB | OKCPCNHB.pls | Public API for inserting, updating, validating and deleting records for Condition Headers and others |
| OKC_EXP_DATE_ASMBLR_PVT | OKCREDAB.pls | Contracts expiry date assembler |
| NA | OKCCINDX.sql | Creates composite index on queue table |
| NA | OKCCQUET.sql | Creates queue table,queues and subscribers |
| NA | OKCQGRNT.sql | Create grants/synonyms on queue objects (tables and sequences) |
| NA | OKCSTQUE.sql | Adds rule based subscribers and starts queues |

# 6

# Enabling Other Setups

This chapter covers the following topics:

- Enabling the Quick Menu
- Defining the Contract Terms Library
- Migrating Clauses to the Contract Terms Library

## Enabling the Quick Menu

You can set up the Quick Menu to navigate to windows in other applications without having to change responsibility and requery the customer. The Quick Menu is available as a selection on the Tools menu.

For example, an agent working on a contract for Business World can access the Contact Center from the Service Contracts Authoring window, by selecting Quick Menu from the Tools menu, and choosing to open the Contact Center. The agent can modify customer information, including addresses, accounts, and other details, and then switch back to the contract without switching responsibilities.

### To enable the quick menu:

1. Select the Application Developer responsibility, then navigate to Application > Menu .

2. Query the submenu OKS_QUICKMENU and add the forms to use in the menu.

   You can add forms from any application, but only forms that are Quick Menu-enabled pass the customer context from the contract. For a complete list, query the seeded Quick Menu for Customer Support CSX_CUSTOMER_SUPPORT_QM.

3. Query the OKS_MANAGER_TOP_MENU

4. Verify that the Quick Menu for Service Contracts is added to OKS_MANAGER_TO P_MENU as a submenu.

5. Set the system profile Customer Care: Start Menu for Quick menu to **Quick Menu for Service Contracts** at the Responsibility level.

   > **Note:** By default, this system profile is set to Quick Menu for Customer Support at the Site level.

## Defining the Contract Terms Library

This topic is covered in the *Oracle Contracts Implementation and Administration Guide*. See *Setting Up Contract Terms Library* in the *Oracle Contracts Implementation and Administration Guide*.

## Migrating Clauses to the Contract Terms Library

This topic is covered in the Migrating Clauses to Contract Terms Library, *Oracle Contracts Implementation and Administration Guide*.

# A

# System Profiles

This appendix covers the following topics:

- Setting Up System Profile Options

## Setting Up System Profile Options

Use this list to identify profile options you need to change for your implementation. You can set profile options in any order and at the following levels, as indicated in the table:

- Site (S)

- Application (A)

- Responsibility (R)

- User (U)

You can access the Find System Profile Values window by logging into the System Administrator Responsibility and navigating to Profile > System. To change profile options, follow the standard procedure outlined in the *Oracle Applications User's Guide*.

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| Sequential Numbering | Always Used | S,A,R | Y | Assigns numbers to documents created by Oracle Financial products and provides a method of checking whether documents have been posted or lost. Set this option at the application level to Always Used. Do not set this option at the Responsibility level. |
| OKC: Change Request Approver | SYSADMIN | S,A,R,U | N | Identifies the default change request approver, which overrides the workflow approver. |
| OKC: Contract Approver | SYSADMIN | S,A,R,U | N | Identifies the default contract approver, which overrides the workflow approver. |

| Option | Sample Values | Level | Required | Description |
| --- | --- | --- | --- | --- |
| OKC: Error Message Recipient | Ronda Abbot | S | N | Identifies the resource (person) to notify in case the outcome execution has failed. This value can be overwritten in case the user selects another person to be notified when entering the outcome condition. |
| OKC : Event Listener Iterations | 100 | S, A | N | Allows System Administrators to set the work load of the listeners. The default load (listener its/iterations) is set to 100. |
| OKC: Global Update Privilege | Yes | S | N | If set to **Yes**, an active contract will have the status QA Hold when you open the contract. The Open for Update button will not appear if the profile is set to **Yes**. |
| OKC: Number of days to retain a message on the exception queue | 30 | S | N | In case of any exceptions encountered while running events, messages are thrown into the exception queues. This profile determines the retention period (in days) of such messages. |
| OKC: Outcome Listener Iterations | 100 | S, A | N | Allows System Administrators to set the work load of the listeners. The default load (listener its/iterations) is set to 100. |
| OKC: Public Group Creator | -- | R,U | N | Set this profile option to **Yes** at the user level, to enable users to create public groups in Oracle Service Contracts. |
| OKC: Renewed Contract Identifier | R | S,A,R,U | N | Provides prefix identifier, which is attached to the system date and shown as a modifier on the renewed contract. |
| OKC: Schedule Rule Alert Window | -- | R,U | N | Determines the number of days before a task is due when the user is notified of an upcoming task. |
| OKC: Schedule Rule Escalate | 1 | R,U | N | Determines the number of days after an incomplete task's due date when escalation begins. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKC: Time UOM Class | Time | S | Y | Limits the units in the Map Time Units window. Note: This is case sensitive and **Time** should be entered using the exact case. |
| OKC: View Contracts By Organization | No | S,R | Required for multi-org installation | Set this profile option to **Yes** at the user level, to provide users visibility to contracts authored in the operating unit associated with their responsibility. |
| QP: Time UOM Conversion | Oracle Contracts | S,A,R,U | N | Allows users to price service based on the calendar, as used by Service Contracts, or the standard Oracle Applications conversion. If set to **Oracle Contracts**, pricing takes the calendar into account, including irregular length months or leap years. |
| OKS: Billing Schedule Level | Top Level | S,A,R,U | N | Defines the default level for billing schedules. You can: <br><br> • Define billing periods with complete control of dates and amount (**Equal Amount**). <br><br> • Define billing periods so that Service Contracts calculates the amount (**Top Level**). <br><br> • Define billing dates and amounts to be charged for each period for each covered level (**Covered Level**). |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Category for Order Management Originated Contracts | Service | S | N | Determines the contract category for Order Management originated contracts. |
| | | | | If set to **Service**, the service ordered interfaces to Contracts with category as Service Agreement and the line type Service. |
| | | | | If set to **Warranty**, the contract created is of the Warranty/ Extended Warranty category with the line type Extended Warranty. |
| | | | | If set to **Subscription**, the Contract category created for the service ordered is Subscription Agreement and the line type Service. |
| OKS: Check Coverage Match | No | S,A,R,U | N | Set this profile option to **Yes**, to force the renewal consolidation process to preserve differences in coverage between source and target contracts. |
| OKS: Consolidate Warranty for Multiple Orders | Yes | S | Y | Determines if an order for products with warranties, sold in Order Management, should be consolidated when the service contract is created. Similar warranties are grouped on a single contract rather than creating separate contracts. |
| OKS: Contact Center Date Range | -- | S,A,R,U | Y (for viewing contracts from the Contact Center) | Controls the date range of contracts that can be viewed from the Contact Center. Enter the number of days prior to the current date. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Contracts Validation Source | Install Base | S,A,R,U | N | Allows the user to define the organization information that should be referenced when automatically creating a contract. You can set the profile as follows: <br><br> • **Ship From Org - Order Management** <br><br> • **Sold From Org - Order Management** <br><br> • **Install Base** <br><br> • **Master Org** <br><br> The organization that sells the product may be different from the organization that ships the product. If the selling organization has the relationship with the customer, and has agreed to make the sale, it may be that the selling organization should be the one to honor any warranty associated with that product, rather than the organization that shipped the product. In this case, the profile option could be set to **Sold From Org - Order Management**. The organization information is obtained via the sales order line information. |
| OKS: Copy Notes | Yes | S,A,R,U | -- | Allows notes to be copied. |
| OKS: Counter Validate | NO | S,A,R,U | N | Set to **Yes**, to consider only those counter readings marked as valid in Install Base for billing purposes. If set to **No**, all counter readings are considered for billing. |
| OKS: Credit Card Display Privileges | -- | S,A,R,U | N | Determines how much of a customer's credit card number is displayed on screen. Valid values are: <br><br> • **All**: Complete credit card number is displayed. <br><br> • **Limited**: The last four digits are displayed. <br><br> • **None**: The card number is not displayed. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Credit Card Minimum Authorized Amount | 50 | S,A,R,U | N | Determines the minimum amount that is authorized against a credit card. |
| OKS: Credit Card Validation Level | Validate | S,A,R,U | N | Determines whether to validate the entered credit card number, by calculating the checksum or obtaining an authorization. This profile is used during the QA process. |
| OKS: Debug Error Log | No | S,A,R,U | N | Set to **Yes** to generate a debug file. Should remain set to **No**, unless directed by a support analyst. |
| OKS: Default Contract Status for Transfers | Active | S | N | Determines the status of the new contract created for the new party. |
| OKS: Default Line Style | Service | S,A,R,U | Y | Defines the default line style of a service contract. Only Service, Extended Warranty, Warranty, Subscription, and Usage line styles are supported for Service Contracts. |
| OKS: Default Order Type for Subscriptions | Mixed | S,A,R,U | Y (if using tangible subscriptions) | Defines order type for subscription line orders. |
| OKS: Default Pricing Date | Sysdate | S,A,R,U | N | Defines the date used to determine the correct price list and pricing effectivity. |
| OKS: Default QA Checklist | Default Quality Assurance Check List | S,A,R,U | Y | Specifies the QA default when authoring new contracts. This profile option applies only to new contracts that are created during authoring. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Default Sales Person for Renewal | -- | S,A,R,U | N | Sets the default sales person for sales credits on a renewed contract. It also populates this name in the Vendor Contact in the Party Contact Role. |
| | | | | If the profile option OKS: Use Territories to Default Sales Person is set to **Yes**, the sales representative is retrieved from resource management instead of this profile option. |
| | | | | If OKS: Enable Sales Credits profile option is set to **Drop**, this profile option is ignored. |
| OKS: Default Time Zone | Pacific/Pitcairn | S,A,R,U | N | Determines the default time zone for business process during Coverage template definition. |
| OKS: Electronic Renewal URL | http://www. vision.com/ renewals | S,A,R,U | N | Determines the URL for the web page provided to customers for Electronic Renewals. |
| | | | | Customers can use the web page to review and respond to renewal notifications. |
| OKS: Enable Grace Period | No | S | N | Indicates whether users should be able set up a Grace Period, when authoring or updating a contract. This site level profile option defaults to **No**, which causes Grace Period fields on the Contracts Authoring form to be grayed out. If set to **Yes**, users can populate the Grace Duration and Grace Period fields. In addition, entitlements processing honors the grace period. |
| OKS: Enable Install Base Integration Notifications | Yes | S,A,R,U | N | Determines if notifications are to be sent for any contract creation or updates based on Install Base operations. |
| OKS: Enable Negative Invoicing | NO | S,A,R,U | N | Set to **Yes**, to enable negative amounts on invoices. If set to **No**, negative values are invoiced at zero value. |

| Option | Sample Values | Level | Required | Description |
|--------|---------------|-------|----------|-------------|
| OKS: Enable Sales Credits | Retain | S,A,R,U | N | Determines Sales Person, Revenue Type Distribution, Revenue Type, and Vendor Contact during contract renewal |
| | | | | If set to **Derive**, this profile option refers to the following other profile options to obtain information to populate the new contract: |
| | | | | • Sales Person (unless OKS: Use Territories to Default Sales Representative is set to Yes) |
| | | | | • Revenue Type Distribution |
| | | | | • Revenue Type |
| | | | | • Vendor Contact. |
| | | | | If set to **Retain**, existing information is copied into the renewed contract. |
| | | | | If set to **Drop**, none of this information is populated in the new contract. |
| OKS: Full Credit for Product Return | Y | S | N | Determines if a full or partial credit should be issued when a subline is terminated as a result of a customer product return. |
| OKS: Intangible Subscription Pricing Method | Subscription Based | S | N | Determines the pricing method for Intangible Subscription items. Pricing method can be set to Effectivity based or Subscription based. |
| OKS: Item Display Preference | Name | S,A,R,U | Y | Specifies the display order in the list of values for Inventory items when selecting Covered Products or Covered Items, while creating a subline in the Authoring form. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Layout Template Source | XML Publisher | S | N | Specifies the method you wish to use to create the e-mail renewal notices sent by the electronic renewals process:<br><br>• **XML Publisher**: Permits you to create the text of the e-mail communications to customers using software capable of producing an RTF file, such as Microsoft Word, and use Oracle Service Contract's Integration with Oracle XML Publisher to create the templates.<br><br>• **Standard**: You must use the XML reporting capability supplied by Oracle Contracts. This is the default setting. |
| OKS: Mass change security level | Advanced | S,A,R,U | Y | Defines list of update levels available to the user when making mass change. If set to **Basic**, then only the Contract and Contract Group level can be selected for updates. If set to **Advanced**, users can perform mass change updates at all levels. |
| OKS:Merge System Transfers | Yes | S | N | Determines if contracts covering multiple products in a system are merged into a single new contract when an instance is transferred. |
| OKS: Minimum Service Duration | 1 | S,A,R,U | N | Defines the minimum duration of a service for cotermination. If a service program sets its cotermination date for duration less than the minimum service duration, then the service program honors the minimum service duration to determine the cotermination date. |
| OKS: Minimum Service Period | Months | S,A,R,U | N | Defines the minimum period of a service for cotermination. If a service program sets its cotermination date for duration less than the minimum service duration, then the service program honors the minimum service duration to determine the cotermination date. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Notify Contract Administrator | CONMGR | S,A,R,U | Required for reas- signment of sales reps | Identifies the administrator to be notified if unable to reassign the sales representative on the contract. |
| OKS: Notify Sales Administrator | CONMGR | S,A,R,U | Required for reas- signment of sales reps | Identifies the administrator to be notified if the sales representative setup is incomplete. |
| OKS: Notify Territory Administrator | CONMGR | S,A,R,U | Required for reas- signment of sales reps | Identifies the administrator to be notified if the territory setup is incomplete. |
| OKS: Notify User of Install Base Integration Notifications | CONMGR | S,A,R,U | N | Identifies the user to be notified whenever a warranty or extended warranty is created or updated in Install Base. |
| OKS: Parallel Worker | NO | S,A,R,U | N | Defines how the Service Contracts Main Billing concurrent program is run for efficient performance. Set to **Yes** to spawn multiple concurrent requests for Service Contracts Main Billing program, when the billing load exceeds 500 lines. |
| OKS: Payment Method for AR Interface | Credit Card | S | N | Captures the payment method for a customer for integration with iPayment Server. |
| OKS: Raise Credit Memo for Install Base Transactions | YES | S | N | Determines if a credit memo is raised when a contract subline is terminated due to transfer, return, or termination of a customer product in Install Base. |
| OKS: Reprice Warning Message | YES | S,A,R,U | N | Suppresses the warning message while repricing or manually overriding the final price. If set to **No**, the warning message "This action is irreversible. Do you wish to Continue?" is suppressed. |
| OKS: Revenue Type Distribution for Sales Credit | 100 | S,A,R,U | N | Sets the credit percentage on a renewed contract for a sales representative. In order to pass QA, revenue distribution for a renewal contract must be 100%. |
| | | | | If OKS: Enable Sales Credits is set to **No**, this profile option is ignored. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Revenue Type for Sales Credits | Quota Sales Credit | S,A,R,U | N | Sets the Sales Credit type, for example quota or non-quota, for creating the sales credits during renewal. |
| | | | | If OKS: Enable Sales Credits is set to Drop, this profile option is ignored. |
| OKS:Service Request Creator | EBUSINESS | S,A,R,U | Y (if using automated service request creation) | Sets default username for automated creation of service requests. |
| OKS: SMTP HOST | vissmtp02. visioncorp.com | S,A,R,U | Required for e-mail quote | Identifies name of the outgoing mail server used to deliver external e-mail notifications. |
| OKS: SMTP PORT | 25 | S,A,R,U | Required for e-mail quote | Identifies SMTP port number used by the SMTP server defined in OKS: SMTP Host profile option. |
| OKS: Subscription Item Filter | Subscription | S | N | Determines which items the user can select in a subscription line under a subscription agreement. If the value is set to **Subscription**, only subscription items can be selected. If the value is set to **All**, the user can select any subscription or products (except service warranty and usage items) at the subscription line. |
| OKS:Summary Transactions | NO | S,A,R,U | Y | If summary transaction is unchecked at the contract header level, determines if the summary or detailed transactions are sent to Oracle Receivables. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Territory Sales Person for First Year Contracts | Derive | S | N | Determines whether the application uses sales territories to assign sales agents (vendor contacts of role Salesperson) to new contracts created from orders in Oracle Order Management. |
| | | | | A setting of **Retain**, causes the application to copy the sales person on the order. |
| | | | | A setting of **Derive** uses sales territories to assigns one resource as a vendor contact of type Salesperson to each contract. |
| OKS: Transfer Party Relationship | CUSTOMER/ SELLER | S | Y | Specifies the relationship that would identify related parties during the transfer management process. If this relationship does not exist between the parties, the parties are considered unrelated. |
| OKS: UOM Code | -- | S,A,R,U | N | Sets the Unit of Measure (UOM) code for covered party, site, system, and customer. |
| OKS: Update Contract with Install Base Quantity | Yes | S | Y | Determines if the contract covered product quantity is updated as a result of the update to item instance quantity in Install Base. This does not apply to other transactions, such as a split quantity. |
| OKS: Usage Billing Calculation | Counter | S | Y | Determines readings to be considered in Usage billing computation. |
| OKS: Use Advanced Pricing for Manual Adjustment | YES | S,A,R,U | N | Controls integration with Advanced Pricing for manual override of the final price. If set to **No**, Advanced Pricing is not used to create or derive price adjustments. If set to **Yes**, Advanced Pricing is used to derive price adjustments. All adjustments is stored in Oracle Service Contracts and can be viewed via the pricing adjustment screen. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Use Territories to Default Sales Person | YES | S,A,R,U | N | Determines whether to use Territories to default the sales representative for a renewed contract. If set to **No**, the sales representative is retrieved from OKS: Default Sales Person for Renewal. If set to **Yes**, the application retrieves the sales person assigned by territory. Territories must be setup if this profile is set to **Yes**. This profile must be set to **Yes** if using Reassign Resources concurrent request. If OKS: Enable Sales Credits is set to **Drop**, this profile option is not used to populate the sales person renewal agreement. |

| Option | Sample Values | Level | Required | Description |
|---|---|---|---|---|
| OKS: Vendor Contact Role | SALESPERSON | S,A,R,U | N | Sets the role for creating contacts during contract renewal as well as during extended warranty creation. Select any role from the list of values to be added to the party role contacts, which will be copied from the expiring contract. This profile option populates only the role. The person assigned to this role is named in the profile option OKS: Default Sales Person for Renewal. The Concurrent Process for Reassigning Resources looks to the Vendor Contact Role of SalesPerson to validate the name of the sales rep indicated in the territory assignment. Therefore, consider setting this profile option to SalesPerson. If OKS: Use Territories to Default Sales Person is set to **Yes** and the vendor contact role is SalesPerson, this party role contact name may be changed to match the territory assignment. If OKS: Enable Sales Credit is set to **No**, this profile option will be ignored. |
| OKS: Wallet Path | vision.com | S | N | When the environment is set up for HTTP communications over SSL, a wallet is required to make a HTTPS request. This profile specifies the directory path where the wallet is saved and that wallet will be used in the E-mail Quote functionality. Wallet password is appended to this path with a $$ prefix in the format path$$pswd. |

# B

# Lookups

This appendix covers the following topics:

- Defining Lookups

## Defining Lookups

The following table identifies lookups that you can define for your implementation. These lookups have one of the following access levels:

- **User**: Allows you to change the lookup code.

- **Extensible**: Allows you to add new lookup codes. However, you cannot modify seeded lookup codes.

> **Note:** Lookups with the System access level cannot be modified and are not included in this appendix.

You can enter lookups in any order by navigating to the Application Object Library Lookup window. For lookups beginning with the:

- OKC prefix, log into the Contracts Manager responsibility.

- OKS prefix, log into the Service Contracts Manager responsibility.

Navigate to Setup > Others > System > Lookups.

Follow the standard procedure outlined in the *Oracle Applications User's Guide*.

| Lookup Types | Description | Lookup Meaning Example | Access Code User: U Extensible: E |
|---|---|---|---|
| OKC_CONTINGENT_EVENTS | Defines contingent events | Acts of God | E |
| OKC_PRICE_TYPE | Defines pricing types | Firm Fixed | E |
| OKC_PROCESS_USAGE_TYPES | Defines process usage type | Approve Change Request | E |
| OKC_ROLE | Defines party contact roles | Customer | E |

| Lookup Types | Description | Lookup Meaning Example | Access Code User: U Extensible: E |
|---|---|---|---|
| OKC_SUBJECT | Standard article subject types | Administration | E |
| OKS_AUTO_RENEW_STATUS | Defines status for autorenewal | Quote Accepted | U |
| OKS_BILLING_RATE | Defines types of billing rates | Overtime rate | U |
| OKS_BILL_ACTIONS | Defines billing action rules | Usage Averaging | U |
| OKS_CONTRACT_CATEGORY | Defines contract category from OM | Service | U |
| OKS_COVERAGE_SEARCH_TYPE_WEB | Defines Simple Search list in Coverage Template. | Name | U |
| OKS_COVERAGE_TEMPLATE_NAV_WEB | Defines list for Navigation pages in Coverage Template module. | Coverage Times | U |
| OKS_CREDIT_AMOUNT | Defines credit amount types for service termination | Full | U |
| OKS_GLOBAL_WEB | Defines defaults global main | Organization Defaults Overview | U |
| OKS_MEDIA_DEF | Defines invoice media definitions | Fax | U |
| OKS_MED_PRTY_TYPE | Defines subscription media/property type | Microfiche | U |
| OKS_ORG_WEB | Defines organization overview navigation | Global Defaults | U |
| OKS_PARTY_WEB | Defines party overview navigation | Global Defaults | U |
| OKS_PBL_PRORATE | Defines OKS locked price breaks prorate value | All | U |
| OKS_REPORT_TYPE | Defines report types | Reminder Notice | U |

| Lookup Types | Description | Lookup Meaning Example | Access Code User: U Extensible: E |
|---|---|---|---|
| OKS_REPROCESS_OPTIONS | Defines reprocess options | All | U |
| OKS_RESOURCE_CLASS | Defines resource classes | Excluded | U |
| OKS_SALES_CREDITS | Defines sales credits | Derive | U |
| OKS_SALES_PERSON | Defines how to derive salesperson for territory | Retain | U |
| OKS_SC_DISTRIBUTION | Defines sales credit definition | Full | U |
| OKS_SC_YES_NO | Defines Service Contracts yes/no | Y | U |
| OKS_SUBS_ITEM_FLTR_VALUES | Defines subscription item filter values | Subscription | U |
| OKS_SUB_TYPE | Defines subscription type | Membership | U |
| OKS_SVC_PERIOD | Defines service periods | Day | U |
| OKS_TRANSFER_OPTIONS | Defines service contracts transfer option | Transfer If Not Related | U |
| OKS_TRANSFER_STATUS | Defines transfer status | Active | U |
| OKS_USAGE_BILLING_BASED_ ON | Defines usage billing calculation methodology | Counter | U |
| OKS_USAGE_SEARCH_TYPE_ WEB | Defines list box in Simple Search drop down within the Usage Template | Description | U |
| OKS_USAGE_TYPE | Defines billing types for usage items | Fixed Rate | U |
| OKS_YES_NO | Defines service contracts lookup code | Yes | U |

# C

# Concurrent Programs

This appendix covers the following topics:

- Running Concurrent Programs

## Running Concurrent Programs

The following table provides a listing of the reports that you will run during your implementation of Oracle Service Contracts. These programs are described within the procedures covered in this guide and the *Oracle Service Contracts User Guide*.

| Concurrent Program | Responsibility | Description |
|---|---|---|
| Autoinvoice Import Program | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Build Attribute Mapping Rules | Oracle Pricing Manager | Generates the attribute mapping rules for all attributes that have the Attribute Mapping Enabled box selected. See the *Oracle Advanced Pricing Implementation Manual.* |
| Contracts Status Change by Contract | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Contracts Status Change by Range | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Date Assembler | Service Contracts Manager | See Running the Date Assembler for Renewals, page 4-10. |
| Fetch Receivables Info for Billing | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Generate Territory Packages | CRM Administrator | Updates changes made to territories. See the *Oracle Common Application Components Implementation Guide*. |
| Listener for Events Queue | Service Contracts Manager | See Starting the Event and Outcome Listeners, page 4-7. |
| Listener for Outcome Queues | Service Contracts Manager | See Starting the Event and Outcome Listeners, page 4-7. |
| Service Contracts Concurrent Program for Reassigning Resources | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Service Contracts Main Billing | Service Contracts Manager | See About Concurrent Programs, *Oracle Service Contracts User Guide*. |
| Service Contracts Order Capture Integration | Service Contracts Manager | See Running Concurrent Programs to Generate Contracts Automatically, *Oracle Service Contracts User Guide*. |
| Service Contracts Setup Report | Service Contracts Manager | See About Reports, *Oracle Service Contracts User Guide*. |
| Update Price Lists with Multi-Currency Conversion Criteria | Pricing Manager | Enables price list windows for multi-currency conversions. See the *Oracle Advanced Pricing Implementation Manual*. |
| Workflow Background Process | Service Contracts Manager | See Starting the Workflow Background Process, page 4-9. |