

Oracle® Contracts Core

Concepts and Procedures

Release 11*i*

April 2000

Part No. A83649-01

ORACLE®

Part No. A83649-01

Copyright © 2000, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Contracts Core is a trademark of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Intended Audience.....	ix
Structure	ix
Conventions	x
Understanding Contracts Core	
Overview of Contracts Core	1
Oracle Contracts Integration Points	2
Self-Serve Functionality	2
Understanding the Contract Launchpad.....	2
Understanding Contract Groups	3
The Two Main Components of Contract Groups.....	3
Understanding Contract Articles.....	4
What Are Contract Articles?.....	4
Different Types of Articles in a Contract.....	4
Article Subjects	5
Article Sets.....	5
Understanding Contract Rules.....	6
What is a Rule?	6
Difference Between Rules and Articles.....	6
Tracking Deliverables with Rules.....	6
Rule Groups	7

Where Rules Can Be Used.....	7
Quality Assurance and Rules	8
Understanding the Contract Index	8
What is the Contract Index?.....	8
How the Contract Index Works	9
How the Contract Index Can Help You.....	9
Understanding Subcontracts.....	9
Understanding Status and Operations.....	10
Access to Contracts and Security	11
Understanding Line Styles and Sources.....	14
Understanding Roles and Contact Sources	16
Understanding Events	17
Types of Events.....	18
Understanding Order Capture Integration	20

Implementing Contracts Core

Standard Articles	23
Process Definitions	25
Maintain Time Units of Measure.....	26
Setting up Status and Operations.....	27
Defining Quality Standards	27
Defining Termination Reasons	28
Defining a Category	28
Setting up an Approval Process	30
Defining Line Styles	30
Defining Party Role or Contact Role.....	31
Defining Rule Groups	32
Setting up Events	32
Defining Sources.....	35
Customizing Events	36

Using Contracts Core

Working with the Launchpad and the Contract Navigator	49
Creating a New Contract.....	50
Creating a Contract from a Template	50

Opening a Contract	51
Creating a New Version	51
Viewing the History of a Contract	52
Creating a Change Request.....	52
Closing the Change Request for a Contract.....	53
Copying a Contract	54
Creating a Subcontract.....	55
Renewing a Contract.....	56
Changing the Status of a Contract	56
Extending a Contract	57
Extending a Contract Line	57
Terminating a contract.....	58
Terminating Contract Lines	58
Viewing the Schedule	59
Communications	59
Authoring Contracts	60
Entering Summary Information.....	61
Entering Contract Parties	62
Specifying Contract Line Item Information.....	62
Entering Article Information in a Contract	64
Entering a Standard Article	64
Entering a Non-Standard Article by Copying a Standard Article.....	65
Entering a Non-Standard Article	65
Entering Rules.....	66
Checking the Contract	66
Submitting for Approval	66
Using Conditions.....	67
Copying from a Condition Template	67
Creating a Contract Specific Condition	67
Reviewing Errors from Asynchronous Processes	68

Send Us Your Comments

Oracle Contracts Core Concepts and Procedures, Release 11*i*

Part No. A83649-01

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

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

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

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

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

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

Preface

Welcome to the Oracle Customer Relationship Management, Release 11*i*, suite of applications.

This Concepts and Procedures manual provides information and instructions to help you work effectively with Oracle Number Portability.

This preface explains how Concepts and Procedures is organized and introduces other sources of information that can help you.

Intended Audience

This guide is aimed at the following users:

- Technical Service Representatives
- Customer Service Representatives
- Service Administrators

Structure

This manual contains the following chapters:

“Understanding Contracts Core” provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application’s relationships to other Oracle or third-party applications.

“Using Contracts Core” provides process-oriented, task-based procedures for using the application to perform essential business tasks.

“Administering Contracts Core” provides task-based procedures for required for ongoing system maintenance and includes information on administration tools and utilities.

Conventions

The following conventions are also used in this manual:

Convention	Meaning
·	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
·	
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
< >	Angle brackets enclose user-supplied names.
[]	Brackets enclose optional clauses from which you can choose one or none.

Understanding Contracts Core

This topic group provides overviews of the application and its components, explanations of key concepts, features, and functions, as well as the application's relationships to other Oracle or third-party applications.

Overview of Contracts Core

Oracle Contracts automates the process of creating and managing contracts. With Contracts, you can customize functions and guidelines to make it easier to author contracts and manage your contract repository.

A contract is a legally binding agreement between parties. Almost every type of organization inevitably contracts for some kind of good or service. With Oracle Contracts you can record, group, query, and use contract data for professional decisions. By using the automated processes in Contracts, you can author and track contracts and the specific terms and conditions of the individual contracts.

The main entry point to contracts is the Launchpad. It uses the concept of Contract groups helps you organizing your contracts. For additional information, see [Understanding the Contract Launchpad](#) and [Understanding Contract Groups](#).

When you create a new contract, you will use articles. Articles are the textual presentation of rules, which are the same contract terms and conditions with the addition of computer-readable formatting that cause Oracle Contracts to take action based on the information. To get a quick overview of all components of a contract, you can use the Contract Index. You can use Oracle Core Contracts to create and manage subcontracts.

See [Understanding Contract Articles](#).

See [Understanding Contract Rules](#).

See [Understanding the Contract Index](#).

See [Understanding Subcontracts](#).

Oracle Contracts Setup

Several concepts are important for understanding Oracle Contracts. Operations permitted on a contract are controlled with contract statuses.

See [Understanding Status and Operations](#).

For an overview of how users or group access to contracts, see [Access to Contracts and Security](#). The concept of individual contract lines is explained in [Understanding Line Styles and Sources](#).

Roles and parties setup is explained in [Understanding Roles and Contact Sources](#). The automation of contracts is supported through events. See [Understanding Events](#).

Oracle Contracts Integration Points

Contracts integrates with Oracle Order Capture, Order Entry, Accounts Receivable, Accounts Payable, Purchasing, Inventory, Task Manager, and Human Resources. It also integrates with the Service Core applications, which include Installed Base.

Self-Serve Functionality

The self-serve functionality in Contracts is designed to make it easy for people without special training to enter and access information over the Internet.

Contract Browse lets you call up the contract by contract number and lets you select the start and end dates, the type of contract, a brief description, and the amount of the contract.

Understanding the Contract Launchpad

The Launchpad has functional areas: the Contract Launchpad and the Contract Navigator.

The Launchpad Inbox displays contracts that require action. The Find button brings up a pop-up called Inbox Find. By entering information in either the Select Status or Select Notification Type fields, you can search for a particular contract. You can use the Respond button to invoke workflow functions such as approving or rejecting a contract.

Another region on the Contract Launchpad is the Recent Documents area. The most recently modified documents can be accessed from here by double-clicking the contract, or by selecting the contract and clicking the Open button.

Bookmarked documents is a short list of contracts you maintain independent of any other criteria. You can access bookmarked contracts more quickly. Double-click on a contract to open it, or click the Open button. You can open a selected contract or remove the bookmark using the buttons.

You can use the Contract Navigator to get a more structured access to contracts, organized by contract groups. On the right side you see the contracts in a grid that belong to the group selected on the left side.

From the grid you can right-click to invoke several contract execution functions, or you can double-click the contract to get a quick overview of the contract. In addition to the overview, double-clicking gives you access to the contract history, schedule, and communications.

Understanding Contract Groups

Contract groups are important in the contract creation process. Groups do not permit or restrict access to contracts. They are a convenient way to organize contracts.

You can group contracts by random criteria. You can create groups and associate or connect contracts to any number of other groups, making it easier to track, report, and follow-up on procedures.

Create as many contract groups as you need to help you and other users find related information regarding contracts and contract parties.

You can create private groups which are accessible only to the user who created that particular group. Private groups can contain individual contracts as well as public groups.

If the system administrator gave you the privilege, you can also create public groups that are accessible to all users. Public groups can contain individual contracts as well as other public groups.

The Two Main Components of Contract Groups

Contracts grouping functionality is made up of two parts: defining the group and maintaining the group.

- Contract group definition. Create groups to collect references to contracts by any criteria. Enter a name and a description. If you have the public group creator privilege, you can define public groups by checking the public flag.
- Contract group member maintenance. You can manage and oversee the groups that were created by the group definition. Use the window to maintain group hierarchies or to assign contracts to groups.

In each of the tabs, Contracts and Groups, you can choose between two views. One view gives you access to all unassigned contracts or groups. The other expands the candidates to add to a contract group or to all groups or contracts.

Use the left arrow in the middle to select a group or contract from the right and add them to the group on the left. To remove a contract or group, select that contract or group on the left side and press the right arrow.

Understanding Contract Articles

This topic explains what contract articles are and how they are used when authoring a contract. The following subjects are covered:

[What are Contract Articles?](#)

[The Library of Articles](#)

[Different Types of Articles](#)

[Article Subjects](#)

[Article Sets](#)

[The Difference Between a Contract Rule and a Contract Article](#)

What Are Contract Articles?

A Contract Article is the text that describes and details the terms and conditions that are attached to a contract. Also see [Understanding Contract Rules](#).

Different Types of Articles in a Contract

There are two different types of articles: standard and nonstandard.

- Standard articles are referenced from a library of articles. Standard articles cannot be changed. You still may add variation to an article in a contract.

- Nonstandard articles are specific to the individual contract. They are either written by the author of the contract, or they might be taken from the Library of Articles and modified.

Article Subjects

Articles have a subject to classify it. The article subjects are defined as lookup codes with the lookup type: OKC SUBJECT.

Article Sets

The article set in the Standard Articles Library Window has two tabs. On one tab you record the pricing type. On the other tab you add a category.

When you assign a category to an article set, then this article set appears in the List of Values when you author a contract. You can then drill down to articles from that set. However, the authoring window does not limit the articles you select to sets that are assigned to a category. You can still access articles directly when authoring, even though their article set is not assigned to the current category.

Create new article sets by defining a new lookup code with the lookup type: OKC ARTICLE_SET.

Incompatibility

Articles may be defined to be incompatible with each other. For example, you could make a payment term of 30 days net incompatible with a 10% advance payment.

You cannot define an incompatibility before both incompatible articles are defined.

Release

Creating new releases of articles is a way to maintain different versions of an article. Creating a new release does not impact articles in other contracts. To change an article already in use, you must create a new release, which automatically becomes the most current version after the previous release expires.

Used in Contracts

The Used in Contracts button in the Release tab is only active after you have used this article in at least one contract. Then you can click Used in Contracts to see in which contract the article is used.

Understanding Contract Rules

This section defines what contract rules are and explains how they are used in Oracle Contracts. Because rules are what trigger many functional processes within Contracts, it is essential that they be clearly understood. This section covers the following subjects:

[What is a Rule?](#)

[The Difference Between Rules and Articles](#)

[The Types of Rules that are in Contracts](#)

[Rule Groups](#)

[Where Rules Need to be Created](#)

[Quality Assurance and Rules](#)

What is a Rule?

Rules are the terms and conditions of a contract. Contracts can automate data and trigger other application processes by using the computer-readable, formatted rules. Rules are created so that you can control different processes to manage a contract. You can use billing rules, for example, to determine when and how a bill should be sent. Contracts has a preset list of rules.

Difference Between Rules and Articles

In Contracts the terms or conditions of a contract are referred to as rules and articles. Articles are the textual presentation of rules, which are the same contract terms and conditions with the addition of computer-readable formatting that cause Oracle Contracts to take action based on the information.

When you put an article into Contracts, that article represents text. With a rule, however, you can initiate functions within Contracts. For example, an article might state that Company A agrees to sell a product to Company B. When this information is entered as a rule, it is stored in a formatted way that allows the system to handle billing or prepayments.

Tracking Deliverables with Rules

Rules capture additional information you want to store for a contract. You can group rules into rule groups that help to enforce business processes and company standards.

Schedule rules capture date information to help you manage deliverables. Schedule rules store point-in-time specification that can be used with events. An example could be “drug-free certification must be provided within 30 days of contract signing.” In other words, rules can capture information about dates or future dates that are not known at the time a contract is authored. Schedule rules can represent date driven, event, or periodic phrases such as these:

- Date driven: 1st Monday of every month
- Event: after the contract is signed
- Periodic: every 6 months, every week

Rule Groups

A rule group contains one or many rules. Rules collect one or many related pieces of information. Rules are stored in a formatted way as opposed to free text like articles.

You company may choose to assemble rule groups and make some rules optional. Once you have selected a rule group in the rule editor, you can see if the rules contained are optional.

Assign rule groups to your contracts according to your business procedures.

You can save incomplete data, but all non optional rules have to be complete to pass Quality Assurance.

Rules are grouped so that rules with similar characteristics so you can easily reference and access them. For example, you could create a rule group for billing which would group together rule types such as Bill-To, Schedule Billing, Payment Terms.

Use public groups to group your contracts in a structured way that follows your company or department organization structure or business procedures. Use private groups to allow users to group contracts by their personal preferences.

Where Rules Can Be Used

Rules can be put into use at three different places in Contracts:

- Contract header
- Contract line
- Article level

Quality Assurance and Rules

Each Rule may consist of one or more fields. One or more fields may be required entries. Since you can only add rule groups to a contract, but not single rules, the Quality Assurance procedure accounts for this.

The optional flag in the rule groups impacts the required fields of a rule.

If you make a rule in a rule group mandatory, the required fields of a rule are enforced to be entered.

If you make a rule in a rule group mandatory, but there are no required fields, then quality assurance always passes.

If you make a rule in a rule group optional, with all fields being non-required, quality assurance always passes.

Note: If you make a rule in a rule group optional, the required item is only enforced if at least one field of the rule was entered. If you don't enter anything for a rule, the mandatory field is not enforced.

In addition to rules with non-required fields, quality assurance can enforce dependencies or alternatives. For example, in a prepayment rule, you either have to enter an amount or you have to enter percentage, but neither of the fields are mandatory. In this case, the quality assurance check will result in an explanation, why the rule did not pass.

Understanding the Contract Index

You can find and examine various components of a contract by using the Contract Index. This section discusses the following:

[What is the Contract Index?](#)

[How the Contract Index Works](#)

[How the Contract Index Can Help You](#)

What is the Contract Index?

The Contract Index is a tree-style structured navigator. With the navigator, you can highlight specified component parts of the contract. You can look at contract parts from the starting point that you choose.

How the Contract Index Works

The Contract Index has five display options, or Views. With these views of the various contract component parts, you can see some of the hierarchical relationships and certain characteristics of the contract. default, parties, line items, articles, and rules.

For example, you could choose to view the contract rules as the root node. Contracts would then show all of the rules of that particular contract. You could then choose to see the association that the rule has to the parties, line items, terms, or events of the contract.

If, for example, you wanted to find an article that was at the line item level and that dealt with X, you could find either any of the line item rules, or the any of the rules, making it much easier and quicker to locate the information needed.

How the Contract Index Can Help You

By using the Contract Index, you can follow, access, and understand the relationship of contract parts and better understand the context for the contract as a whole.

You can click on any part of the Contract Index tree structure and navigate to any appropriate tab, row, or column that has the information specifics that you are looking for.

Understanding Subcontracts

In order to fulfill obligations of an existing contract, a party may need to establish a new contract with third parties to provide some or all of the items (or materials, or deliverables, etc.) required in the original contract. This is called creating a subcontract. These subcontracts need to maintain relationships between the components (terms, lines, etc.) of the parent agreement and those of the new contract. You can create a subcontract and further specify those articles or rules from the prime contract or master agreement that are to flow-down to, and be incorporated into, a subcontract or task order.

Also, organizations may wish to create task orders from a master agreement, with the terms and conditions of each task order linked from the master agreement.

Alternatively, organizations may wish to simply reference other contracts, (and by inference, the terms and conditions of those other contracts), without specifying any particular details.

With the subcontracting feature, you can:

- Subcontract the components into an existing contract or a new contract.
- Maintain data consistency for the subcontracted contract. For example, a subcontract would not contain your customer as a customer in the subcontract.
- Select multiple lines or sublines for subcontracting.
- Inherit terms and conditions from higher level in the hierarchy for the subcontracted lines or sub lines.
- Copy or reference terms and conditions.
- Modify the copied components.

You can select the Lines or sublines and also the components like parties, articles, rules at the contract header level for subcontracting. After the components are selected, click on right arrow button to move them to the right navigator window.

Understanding Status and Operations

Oracle Contracts allows you to control the status and operations allowed per contract status and category. You can have a status on the contract header level and on the line level. For example, you can expire a single contract line, while the rest of the contract is still active.

You can define as many statuses per status type as you need. Oracle Contracts currently supports seven status types.

- **Entered:** Contract is currently being edited and is neither complete nor approved.
- **Signed:** Contract is approved, but not yet effective.
- **Active:** Contract is approved and effective.
- **Hold:** Contract was set on hold from either: Signed, Active or another hold type.
- **Expired:** Contract was active, but is not effective anymore.
- **Terminated:** Contract was active, but was terminated from either side before the contract expired.
- **Canceled:** Contract never became active and is not planned to become active.

For an overview of allowed status changes see Table of allowed status changes.

The status of a contract changes by two ways. From the Launchpad, you can initiate a status change. See [Changing the status of a contract](#). Some status changes have to

be done through a concurrent program. For example, the program Status Change automatically changes a signed contract to Active, when the start date is reached.

For each of those status types, you can define a default status. This is needed when the concurrent program Status Change automatically updates the contract status. For example, when a signed contract is waiting to reach the start date, the **Status Change** program changes the contract status to the default status for the type Active when the contract reaches the start date.

For each status type, the allowed operations determine which operations are supported. Use contract status to fine tune contracts access. For example, define three statuses for the type active to distinguish which contracts can be updated on-line, which contracts can only be updated through a change request, and which contracts must not be updated once they are active.

To set up new statuses, see [Setting up Status and Operations](#).

Allowed Status Changes

In the following table, auto means when the conditions for a status change are met, the concurrent program Status Change automatically changes the status.

Status To From	Entered	Signed	Active	Hold	Expired	Terminated	Canceled
Entered	Yes	Yes/ auto	Yes/ auto	Yes	No	No	Yes
Signed	No	Yes	Yes	Yes	Yes/ auto	Yes	No
Active	No	No	Yes	Yes	Yes/ auto	Yes	No
Hold	No	Yes	Yes	Yes	yes	Yes	Yes
Expired	No	Yes	Yes	Yes	Yes	Yes	No
Terminated	No	No	No	No	No	Yes	No
Canceled	No	No	No	No	No	No	Yes

Access to Contracts and Security

Oracle Contracts allow you to fine-tune access to contracts. For example, a guideline a specific user may be able to see a contract, but not be able to modify it.

Contract Groups

You can assign one or more contract groups to each contract. There are two kind of contract groups: Public and Private. A private group is created by a user and only the user creating the group has visibility and access to that group. A public group is visible to all users.

Public groups can only be created by users that have the system profile option: OKC Public group creator enabled. If this system profile option is not set or disabled, the user can only create private contract groups.

Everybody with access to a contract group assigned to the contract group has full read and write access to the contract, as defined in the status and operations window.

Contract groups allow you to navigate to contracts through different navigation paths. A contract is referenced in contract groups, not copied. Assign contract groups in the authoring window in the Details subtab of the Summary tab. As a contract administrator, you can use the Assign Contracts Group window to assign and unassign contracts to groups.

Status and Operations

The status and operations window defines by category and status, which operations can be executed against a contract. It is important to understand the allowed operations are defined by status and not by status type.

In other words, if you define a new status for the status type Active, then you have to make sure that you specifically allow operations such as on-line update. If you create a new status without specifying any allowed operations, you implicitly allow no operations for this contract status.

Access

You grant access to contracts through the category. The category definition includes granting access levels to responsibilities. For each category you can define per responsibility, if you grant read-only or modify access.

Within a contract, you can give access to one or many contract groups to a contract.

If you need to give access to contracts across organizations, see Multi-Org Considerations. That section addresses the use of the Access in the Administration subtab in the Summary tab of the authoring form.

There are a couple of operations and statuses that prevent you from changing a contract. For example, while a contract is being changed by somebody else, you

cannot access it. While a change request is pending approval, you can submit another change request, but your change request has to wait for the outcome of the pending change request.

You are restricted in the status changes you can perform on a contract. For example, you cannot terminate a contract that was never active. You have to cancel it. You cannot cancel a contract that was active.

For a complete list of allowed status change operations, see Status and Operations.

Multi-Org Considerations

Ordinarily, a contract is only visible within an organization. Oracle Contracts enforces organization security by default. For some business uses, where contracts may need to be accessed 24 hours a day from any organization around the world, contracts may need to be accessible outside your organization.

If you need to give access to contracts across organizations, you can use the Administration subtab in the Summary tab of the authoring form.

When you log on to Oracle Applications, you automatically sign on within your organization. Therefore, all your actions are restricted to access and modify data for your organization only.

Oracle Contracts allows for reading and updating contracts, by temporarily switching your current organization in the authoring window, if so authorized.

You grant access for read-only or modify to resources, independent of the organization the user is in.

To create a resource, you must choose the CRM Resource Manager responsibility to define a resource.

If you want to use persons or roles already declared somewhere else in the application setup, you must import these sources using the Import Resources function in the CRM Resource Manager responsibility. Make sure you assign contract execution rights in the Define Resource menu.

Access troubleshooting Checklist

1. User does not have access to contract group.
Check if contract group is public or private.
2. User cannot see the contract in the contract group

Check if the user is in the same organization and has access granted. Check if the user is using a responsibility with access to the category

3. User can see the contract, but cannot update the contract.
 - User was not given access
 - User is using a responsibility with read-only privilege instead of modify
 - The category does not allow for the specific operation, because the 'Status and'
 - Operations Window excludes that operation or this operation is not included, thus implicitly disallows that operation.
 - The contract may be in a status change, that does not allow any operations.

Understanding Line Styles and Sources

Line Styles and Sources:

Oracle Contracts provides you with a powerful concept to structure goods and services lines in your contracts. You can use hierarchies to drill up and down between different level of details or description of contract lines. Use line styles to clearly define at which level you price and to define if a contract line is eligible for entitlement and/or invoicing.

Top Line

Each contract may have one or more contract items. Each such item is represented by a contract line, which may be a simple priced item line or may consist of several layers of services, items and exceptions.

The most general and least detailed line in a contract is called the top line.

Only at the top line level you define an article line to be eligible for entitlement and/or invoicing.

To access a top line style from the authoring window, you must include that Top Line Style in the category definition.

Pricing

Pricing allows you to enter a quantity, unit of measure and a price. In the authoring form, the line then calculates the price, which you can overwrite in the negotiated price column to account for contractual agreements.

For each hierarchical path in a line style, you can only price once.

Example: Assume a line style hierarchy such as the following:

1st Level Top Level	2nd Level	3rd Level	4th Level
Item 1	Item 1.1	Item 1.1.1	Item 1.1.1.1
			Item 1.1.1.2
			Item 1.1.1.3
		Item 1.1.2	
	Item 1.2		
	Item 1.3	Item 1.3.1	
		Item 1.3.2	

You can mark the Items 1.1.13, Item 1.1.2, Item 1.2 and item 1.3.2 to be priced items.

When you price the Item 1.3.1, then the Item 1.3.2 is automatically marked on nonpriced.

When you mark the Item 1 at the top line as a priced item, all other dependent lines have their priced tag removed.

Line Types and Item Source:

For each level in the line style hierarchy, you can define your line type and the item source.

Define new line types in the lookup codes.

One special lookup code is the free format line type. This type allows you to enter any description. For any other line type, you can determine the source, from which the line style will present a list of values. Select from the predefined objects where your goods and service line will feed the List of Values from.

If the line is not to be priced and you have selected the system items as the source, then no List of Values will be displayed for the line. In that case, you have to go to the items tab to enter items. This concept support the business requirement that you can enter more than one item per a single contract line while pricing it.

Entering New Items

The Inventory items List of Values is restricted by the organization that is available to the contract. When a contract is entered, the organization ID is obtained from the multi-org profile option value. The inventory organization value is obtained from the Order Entry system parameters.

The contracts List of Values is fed by the Master organization and not the individual Inventory organizations. Therefor, as common for item handling the item is entered into the item master and then copied over to the individual inventory organizations.

Understanding Roles and Contact Sources

A contract is an agreement between at least two parties. Oracle Contracts allows you to add two or more parties when authoring a contract. Oracle Contracts provides a powerful setup to define the party roles and the source for validating parties, entered into the system elsewhere.

The setup of roles differentiates between sell and buy contracts.

Example

If you sell goods or services, then you are the vendor and your customer may already exist in your customer database as a regular customer.

If you buy goods and services, then you are the customer and your vendor may already exist in your vendor database. Even though you are the customer in this contract, you don't want to register as a customer in your customer database just to be available as the customer in the contract.

In both contracts you have a customer and a vendor each. In one contract you are the customer, in the other you are the vendor.

In order to allow for such different roles, you can define from which data source a role is fed.

In addition, Oracle Contracts allows you to define your own roles and to define the source for your customized roles.

Creating a New Role and Modifying Role Sources

Create a new role in the contracts setup menu for categories. You can assign the sources to the new roles in the setup menu for contracts under the menu item Sources. Depending on the intent of the contract, sell or buy, you can assign sources to your roles and the contacts for such roles.

If you want to update an existing role source, you must expire the current assignment for the role and intent. You cannot delete existing role sources to keep the role history of contracts created with a prior role assignment.

The sources you can choose from are predefined objects delivered with Oracle Contracts.

New Roles

In a contract, a role can only be selected if both of the following conditions are met:

- The category you chose allows permits the use of that role.
- The role has a valid source. Make sure that the source has a valid date range, the intent, sell or buy, matches your contract, and that the source has values.

After you have selected a role, you can tab to the following field, where the List of Values automatically refers to the right data.

Note: You cannot enter new customers from the Authoring window. If you enter a contract and you need to add a new customer or a one-time vendor, then you must first define the new customer or create a one-time vendor.

You can either use the setup menu in your contracts responsibility, or you can enter that data where it originates, that is, enter your new customer in Oracle Receivables before you can use it in your new contract.

The Concept of Object and Subject

You need this concept for Service Contracts only.

When you define a category, the category setup allows you to assign rules to roles. Use the subject and object to identify the party role you want to identify in your rules. For example, you may want to determine that your billing rules identifies the seller as the subject, in other words, the seller would send the invoice. The rule edit would then choose the right List of Values when you enter rule information.

Understanding Events

Events provides an infrastructure to ensure that certain processes are initiated in response to events that satisfy user-defined criteria. Events functionality is available not only to Oracle Contracts but also to all Oracle Service application. This topic explains Events and how you can use it within Contracts.

Events allows you to automatically initiate actions such as creating a service request, sending a notification, and creating a task as a result of something happening or before an upcoming event.

In order words, it acts upon phrases like:

When this happens, do something.

At a certain point in time, something happens.

The “something happens” or “at a certain point in time” are referred to as *action*, and “do something” is referred to as *outcome*.

The action can be qualified by conditions. For example, it may not be whenever something happens, but “whenever something happens and some conditions are met.”

In summary, Events comprises the following components: **action, conditions, and outcomes**.

Events allows synchronous and asynchronous processing. The asynchronous processing takes advantage of the advanced queuing method. That is, users do not have to wait until an event is evaluated and all the outcomes are executed to continue with their activities. On the other hands, due to the type of business, some applications such as Relationship Plan might require an immediate event evaluation while users are interacting with the application.

This section covers the following subjects:

[Types of Events](#)

[What are conditions?](#)

[Types of condition lines](#)

[Different Types of Conditions](#)

[What are Outcomes?](#)

[Customizing Events](#)

Types of Events

With Events, you can handle action-based events and date-based events.

Action-based events are predefined actions that occur in the application. Examples of action-based events are contract signed, contract terminated, counter updated, change request initiated.

Date-based events are dates of significance within the application. For example contract expiration date.

You define action-based events and date-based events using the Actions form. In this form you also define action attributes that can be used further in conditions and outcomes.

What are Conditions?

You use conditions to define additional criteria to an action-based event or to a date-based event. The conditions also specify the outcomes that need to be initiated when the condition evaluates to true.

Conditions are composed of header and lines.

In the condition header you specify the event type by choosing either an action-based event or date-based event.

You use condition lines to qualify action and date based events. They are used to specify that an outcome is to be initiated not only when the action happens, but when the action happens and certain conditions are satisfied. For example: "when the contract is signed and the amount of the contract is greater than \$100,000," or "30 days before contract expiration and when the contract category is network license agreement." It is not required to define condition lines for a condition.

Types of Condition Lines

There are three types of condition lines: **expressions**, **counters**, and **functions**.

Using expressions, you can define simple comparisons using the operators such as <, >, =. Expressions are always based on action attributes.

Counter condition lines are applicable for Service Contracts and Installed Base. It allows you to define conditions for a particular counter reaching a value. Counters can be defined for a product or for a service. For example "when the number of photocopies for a photocopier is greater than 1,000," or "when the number of calls made by a customer in a gold service agreement is over 10."

Functions can be used for more sophisticated criteria that can be addressed using PL/SQL procedures. A function accepts one or more parameters and returns true or false after its evaluation. Functions don't need to be based on action attributes.

Different Types of Conditions

There are three types of conditions:

- Condition templates

- „ Independent conditions
- „ Conditions attached to contracts

Using condition templates, you can define multiple samples of conditions. From a template you can define either an independent condition or a condition to be attached to a contract.

Independent conditions are not owned by any object. Whenever an action occurs, regardless the contract, the outcome is initiated if the condition evaluates to true. For example, “whenever any contract is signed create a task to notify the contract administrator”

Conditions can be specific to contracts. Depending on the action, a condition can be related to a contract line or to a contract as a whole. For example “when the contract is signed” relates to the entire contract, and “when the contract line is terminated” relates to a line of the contract.

What are Outcomes?

An outcome is an automated process that performs a task such as initiating a service request workflow or sending a notification. When the condition is true, the outcomes associated with that condition are performed.

Outcomes need to be defined using the Process Definition form. There are four types of outcomes: workflow, PL/SQL procedure, alert, and script. Workflow and PL/SQL procedure are outcomes. Alerts are synchronous outcomes.

You cannot use script for outcomes. Script is not referring to an SQL script, but is referring to an internal scripting tool used for Oracle Relationship Planner.

Customizing Events

Oracle Contracts Core allows you customize Events. For examples on how to create a new Action, see [Coding Action Assemblers](#). To understand how to code functions, see [Coding Functions for Function Expressions in Condition Lines](#). To understand how to code an outcome, see [Coding Outcomes for Conditions](#).

Understanding Order Capture Integration

Creating quotes, contracts and orders

Oracle Order Capture is the CRM interface to Order Management. Use Order Capture to create quotes; generate orders from a contract; and contracts from quotes.

You can use existing contracts as reference to create quotes in Order Capture. In addition, you can update the quote, for example; to reflect a new price list.

You can use contract templates to create contracts from a quote. The contract created is in status Entered, which allows you to modify the contract before it is submitted for approval. Submit the concurrent program "Create Order from a Contract" to create an Order from a contract.

Renewing contracts

You can use Oracle Contracts to automatically create quote for contracts to be renewed.

Using the events component, you define a condition for the seeded date based action "Contract expiration". Then you specify when you want to create a quote. For example: Define "30 days before the contract expiration" as the condition. The seeded outcome for this condition is "Create a Quote from a contract" for contract renewal.

After the quote is created, you still can make changes before creating the order. Once the order is fulfilled the service contract is renewed.

Implementing Contracts Core

This section describes the setup forms and gives references to section that help you understand underlying dependencies.

The Setup documentation consists of the following topics:

[Standard Articles](#)

[Process Definitions](#)

[Maintain Time Units of Measure](#)

[Setting up Status and Operations](#)

[Defining Quality Standards](#)

[Defining Termination Reasons](#)

[Defining a Category](#)

[Setting up an Approval Process](#)

[Defining Line Styles](#)

[Defining Party Role or Contact Role](#)

[Defining Rule Groups](#)

[Setting up Events](#)

[Defining Sources](#)

Standard Articles

The Library of Articles is a reference library that is a small database of previously written and established articles. These articles can be referenced and included in a contract. Part of the process of creating and using articles includes the options of referencing the Library of Articles. Oracle Contracts allows to handle multiple releases of articles, making sure that the most current article is automatically used when authoring new contracts.

For more information on articles, see [Understanding Contract Articles](#).

The article group Name and Subject line appears on the Standard Library of Articles window. There are three main tabs within the Standard Library of Articles window. From the Library of Articles, you can do the following:

- Defining Article Subject
- Defining a New Article Set
- Adding an Article to the Library
- Managing Article Releases

Defining an Article Set

Steps

1. Enter a new Lookup code. Navigate to **Setup > Other > System > Lookup**.
2. Query the lookup code: **OKC SUBJECT**
3. Enter an internal code and a meaning, which will be displayed in the article subject.

Defining a new Article Set

Steps

1. Enter a new Lookup code. Navigate to **Setup > Other > System > Lookup**.
2. Query the lookup code: **OKC ARTICLE_SET**
3. Enter an internal code and a meaning, which will be displayed in the article subject.

Add an Article to the Library

Steps

1. Navigate to **Setup > Contracts > Standard Articles**.
2. Select an article Set. To create a new article Set, See: [Define a New Article Set](#)
3. From the **Actions Menu**, select **New Article**.
Note: This automatically assigns the article set you selected to your new article. You can at any time overwrite this assignment.
4. Enter a Name of the article.

5. Select an article subject. To create new subjects, see: [Define Article Subject](#)
6. Enter the release name for your first release or version of the text.
7. To declare incompatibilities with other articles, enter the incompatible articles in the incompatibilities tab.
8. To add the article to other article sets, enter the article sets in the articles tab.
9. Click **Release Text** to edit the text.

Manage Article Releases

Steps

1. Press the **New** button from the Toolbar.
2. Enter a release name or number.
3. Define the new active date. The end date of a release is automatically determined by the start date of the next release.
4. Edit the text using the **Release Text** button.

Used in Contracts

The **Used in Contracts** button in the release sub tab is only active, once you have used this article in at least one contracts. Then you can click on **Used in Contracts** to see in which contract the release is in use.

Process Definitions

Use the Process Definitions Window to give your processes user-friendly names, so that the non-technical end user is able to use the processes you have defined when authoring or modifying a contract.

This form is used to combines the registering of six different usages:

- „ API (Automated Programming Interface)
- „ Contract approval process
- „ Change request process
- „ Function; used for Condition evaluations
- „ Outcome in the context of Events
- „ Quality Assurance workflow

Steps

- a. Navigate to **Setup > Contract > Process Definition**
- b. Enter a unique name and a description that is displayed to the user in the List of Values.
- c. Enter the usage
- d. Depending on which process you choose, the form will allow to select an allowed type and possibly parameters. When entering workflow or package and procedure names, make sure the spelling is accurate. The Process Definitions form does not validate the existence or the valid status of the referred objects.

Note: When defining outcomes for events, then you cannot use the type script. Script is not referring to an SQL script, but is referring to an internal scripting tool used for Oracle Relationship Planner.

- e. Adding parameters to pass, make sure that they are entered in the correct order.

Maintain Time Units of Measure

Oracle Contracts defines Unit of Measure conversion for Time different from the way other Oracle Applications do. This helps ensuring that the scheduling is more accurate than a simple conversion like: 1 month = 30 days, which is only correct for 5 out of the 12 months of a year.

There are six internal time units for seconds, minutes, hours, days, months and a year.

If you want to define your own time unit conversions for extending a contract or for scheduling, then you must define your own time unit conversions.

Use the Units of Measure window to define your personalized time units, like: A maintenance quarter using a class to handle time units.

This class must be defined in the system profile option: OKC Time UOM class. The profile option is needed to tell the Time Map window which time units should be mapped against the internal time units.

Use the Time Map window to map your time units to the six internal time units. As a one time setup you map your time units like:

Day (your definition) = 1 day (internal definition)

Make sure to map each time unit you want to use in the Time Map window.

Setting up Status and Operations

For more information on status and operations, see [Understanding Status and Operations](#)

Steps

1. Navigate to **Setup > Contract > Status and Operations**
2. Query the status type for which you want to add or modify a status type.
3. Enter a status code and the meaning that is displayed to the contracts user.
4. Check if the batch program **Status Change** should default to this status, when changing the status type of a contract.
5. Navigate to the **Allowed Operations by Category**.
6. Enter the Category you want to control access for.
7. Select the Operation including the level. The operations you can select here are pre-seeded and cannot be changed. Not all operations are available for all status type.
8. Check if you want to allow or prohibit access. If you category and operation is not listed for this status, then the operation is prohibited.
9. Continue with step 6 for other categories or operations on this category.

Defining Quality Standards

Oracle Contracts validates a contract before you can submit it for approval. The default Quality Assurance control (QA) is an internal Process, that will be executed for any contract, even if the user chooses another QA for the contract. You cannot modify or update the default Quality Assurance List.

Each Quality Check consists of one or many Processes. To define a new process, see [Process Definition](#).

Define a new Quality Assurance Checklist

Steps

1. Navigate to **Setup > Contract > Quality Assurance**.
2. Enter a name and a description.
3. In the Processes region, enter the processes. Select the process from the List of Values that should be part of the QA list.
4. Uncheck the Active flag if you want to temporarily exclude the process.
5. Enter the severity.
Ignore: The contract passes Approval, same as Warning.
Warning: The contract passes Approval, same as Ignore.
Stop: The contract does not pass quality assurance if this process fails.
6. In the Parameters region, the parameters defined for the process are displayed. You can override the default value for the parameter.
7. To add more processes, goto step 3.

Defining Termination Reasons

When terminating a contract, you must specify a reason for termination. Define your termination reasons in the Lookup codes. Query for the lookup type OKC_TERMINATION_REASON and enter a new code with a meaning and a description.

Steps

1. Enter a new Lookup code. Navigate to **Setup > Other > System > Lookup**.
2. Query the lookup code: OKC_TERMINATION_REASON.
3. Enter an internal code and a meaning.

Defining a Category

Category definitions allow the user to define which components may be part of a contract category. Historically, they have evolved from subclasses and were renamed to a more user-friendly name.

A contract class drives functionality and is hidden from the user. For example, the class, hidden in the category, determines which authoring form is called from the launchpad.

For example: Choose the class of Service Agreements to create a service category.

There are four tabs in the category definition: Party Roles, Rules, Line Styles and Responsibilities. In each of the tabs you enter the components you want to allow the user to add into a contract with that category. That does not make the entry of such a component mandatory. You cannot delete a component from a category. To remove a component from the category when authoring, you have to expire it.

Add a party to include by selecting a new row and selecting a role from the list of values.

Add rules you want to have access to on any of the three levels: Contract, Lines and Articles. If needed, you can add additional restrictions for the usage of a role in a contract. For example: Qualify the customer role in the billing rule as the object of the billing role, i.e. as the party to which the bill is sent. By selecting a subject you are qualifying the billing role.

The optional flag determines, if the role qualified may be omitted for a rule to pass quality assurance.

When you add a line style, the tree structure of the entire line style is displayed. For every line of a line style, the roles entered in the Party Roles tab of the category are shown in the Party Roles sub tab of the Line Style tab.

The select flag controls if a Party Role can be selected on the contract line level.

For every line of a line style, the rules entered in the Rules tab of the category are shown in the Rules sub tab of the Line Style tab.

The select flag controls if a Rule can be selected on the contract line level.

Each category should be assigned to at least one responsibility and determine if you allow modify access or grant read-only access. A responsibility not included in the list will implicitly prohibits access to the contract.

For more information about how to create Party Roles or Linestyles, see [Party Roles and Roles Contracts](#), and [Line Styles](#). For more information about the role of responsibilities, see: Topical Essay: [Access to Contracts and Security](#).

Setting up an Approval Process

You can define the approver for both, change requests and contract approvals with the setup of workflow.

For a quick work around of setting up an approver for change requests, you use SQL to define your change request approver until your workflow setup is complete. To make the System administrator responsibility your approver, you could use the following SQL statement:

```
UPDATE wf_resources
SET text = 'SYSADMIN'
WHERE name = 'WF_ADMIN_ROLE';
```

For a convenient and quick setup of contract approvers, you can use the system profile option: OKC: Contract Approver. It will override a workflow setup and use the approver, if the workflow is not yet defined.

Defining Line Styles

For an introduction of how line styles work, you can look at the Topical Essay: [Line Styles and Sources](#).

Oracle Contracts delivers a set of line styles you may want to edit or use as is.

To add a new top line style, click on the top item of the tree: Line Style. Add a new line style by clicking on the NEW icon in the toolbar.

A new node will appear in the line style tree with a generated name. Filling in a name will update that generated name. Choose a Line Type to classify your line styles.

You define line types in lookup code. Query for the lookup type OKC_LINE_TYPES and enter new top line styles. You can find the lookup menu in the setup system menu for other applications.

There is one special line type: Free Format. This line type allows you to enter ad-hoc definitions, that do not rely on existing sources like inventory items.

You can determine to price an item line at any level. However, you can only price one 'line' in the hierarchy chain. This enables you to enter quantities and price information.

Add sublines at any time by marking a line in the tree navigator on the left. Click on the NEW icon in toolbar and a node will appear in the tree to be edited just like a top line.

For all line styles, top lines and sub lines, you can determine the source of the line style. Note: In the authoring form, lines with the price flag disabled will not show a list of values if you select system items as the source. For this special case the item source list of values is retrieved in the item sub tab of the line.

OKS: Entitlement and Invoicing Eligibility

Only for Service contracts the tab Valid Operations is of interest. Top lines only have this tab. Oracle Service Contracts allows you to fine tune where you can entitle or invoice lines or contracts.

You ultimately use the [Status and Operations](#) Window to determine if a status entitles a party to a service or product or if the party can invoice, but there are some dependencies to watch out for.

How to set up entitlement and invoicing

1. Define a top line style to be eligible for entitlement or invoicing.
2. You then include that top line in the line styles for the service category you want to use. Make sure that you do not define a line style's effective date outside the categories effective date. Otherwise, you will not be able to define the eligibility for that category.
3. Assign the eligibility in the status and operations window. Make sure that you have no date conflicts with the top line style effectiveness, the category effective dates and the status effectiveness. If you run into such a conflict, the eligibility will not be a valid operation you can control.

Defining Party Role or Contact Role

Use the Source window to assign sources to existing roles. Define new lookup codes to define new roles or contact roles. For new roles, open the Define Party Roles or Define Contact Roles function in the Category Setup menu.

Once you have defined a new role or contact role, they will automatically appear in the [Sources](#) window, where you can assign the sources.

Note: A new party role or contact role will only appear in your contract, if the following conditions are all met:

1. The role or contact role is defined as a lookup code with a current effective date
2. The role or contact role have a current effective date in the sources window.

3. The role or contact role have a source assigned with a current effective date and the right intent, - buy or sell.
4. The source for the role and intent contains at least one value.

Defining Rule Groups

A rule group is a collection of rules. Rule groups allow you to control which rules are used in a contract and what additional information you may want to capture. You can use rule groups to make certain rules optional entries for one rule group by checking the optional flag, while leaving it a mandatory rule for another rule group.

When you select a rule group in your contract, all rules of that group are automatically copied into your contract. Each rule may be optional or mandatory. See [Quality Assurance and Rules](#).

Rule Type Source

For some rules defined, you need to provide sources for each intent, - buy or sell. You can modify the rule type source. You must not remove existing source definition. Adding source definitions here does not have an effect.

Categories:

To access a rule from the authoring window, you must include the rule group in the category definition. You can include more than one rule group to a category.

Setting up Events

Events comprise the following components: Action, Conditions and Outcomes. When you define conditions in a contract or in an independent condition, you combine Action and Outcomes to the condition.

Setting up Events therefor is structured into three setup steps:

[Setting up Actions](#)

[Defining Outcomes](#)

[Conditions](#)

Note: There are two concurrent programs that need to run continuously. Make sure that the following two processes are restarted if the database or the concurrent managers have been restarted:

1. Listener for Events Queue
2. Listener for Outcome Queue

The concurrent program Date Assembler should run once every, preferably later in the evening. This program is used for tasks like automatically renewing contracts that are eligible according to the contract definition.

To understand how to create your own actions, functions or outcomes, see [Customizing Events](#).

Setting up Actions

Use this window to update or define actions like: Contract Signed, Counter Group X updated. For more information of Events and Actions, see [Understanding Events](#).

Steps

1. Navigate to **Contract Events > Define Action**
2. Enter a unique name
3. Choose an Action Type, Action based or Date Based.
4. Enter a description that will be seen in the conditions window.
5. Enter the Correlation. Correlation is a term used in Advanced Queueing.
6. Check if you want to enable the Counter tab for this action when defining Conditions.
7. Check Synchronous (on-line) or asynchronous (batch-mode) action handling.
8. For each attribute you declare here, you must enter the name, element name or column and a data type in the basic tab. These attributes can be accessed in the condition definition.
9. In the Advanced tab you can enter the following:
Min/Max Value: validates the range of entries in conditions
Object Name: Defines the source for the List of Values in the conditions window. The current release supports two objects: Contract Classes and Categories.
Column name: Derived from the Object Name.

Date of Interest: Can be used for Action Based Actions only. Check an attribute when you want a schedule rule to use the attribute value, like the signing date entered. If you leave the Date of Interest unchecked, then the signing date would refer to the date, when the contract signing date was entered, ignoring the value you enter.

Multiple Values: Denote if the action may return more than one value. Then the values are concatenated with a separator in between.

Used in Conditions: Uncheck this flag if the attribute should not be used in conditions, i.e. when you need an internal ID for outcomes or functions.

Defining Outcomes

An outcome is an automated process that performs a task. Defining an outcome is defined in the Process definition window. See [Process Definitions](#).

Conditions

You may choose to define conditions as templates. Defining condition is similar for creating a condition in a contract, creating an independent condition and a condition template.

A template allows you to save incomplete entries because it will not run. A template can be used in a contract where it will be evaluated after it was completed. An independent condition is not tied to a contract and can run independent of a contract. Therefore, an independent condition has to validate before you can save it.

For more information, see [Understanding Conditions](#).

By choosing the menu, you determine if you enter an independent condition or a template.

Steps

1. Navigate to **Contract Events**. Select if you want to define an independent condition or template.
2. Enter a name and a description.
3. Check **Create a Task** if you want to create a task for the CRM Task Manager. Creating a task requires entering a task owner.
4. Check the condition type. For Action you must enter an action, for Date you must enter the number of days, qualify before or after, and you must enter the triggering date.

5. For condition of type action choose if the event is evaluated only once or each time the condition is met. For example: Check the flag if you want to evaluate the condition to true only if the counter exceeds 1,000 for the first time. Leave the flag unchecked if you want the condition to evaluate to true each time the counter reaches 1,000.
6. Build your condition lines.

Each condition may consist of one or many condition lines, with the sequence determining the order in which the lines are processed. You can enter Expressions, Counter or Function lines.

Expressions:

The left value only allows you to enter attributes defined in the action attributes.

Counter:

Select Product or Service and choose a description of the service or product.

If you select a counter group, then your left value's List of Values lists counter group types. If you choose a counter group master, then your left value's List of Values shows the related counter groups.

Functions:

Select Functions declared in the Process Definitions Window. To enter fix values for parameters, click the **Parameter** button.

7. Display the condition lines and check validity.
Click on **Show Condition** to display all condition lines and check their syntactical validity on-line. The checkbox valid is display-only and gives you a quick overview if the condition has validated successfully.
8. Enter **Outcomes** defined in the Process Definitions Window.
9. Click on the **Parameter** button to assign fix values or action attribute values to the outcome parameters.

Defining Sources

The sources window allows you to assign different sources to your roles. You can define your own [Party Roles and Roles Contracts](#). Depending on the intent of the contract, - buy or sell -, you can determine which list of values will show for a role.

For example: In a contract where you are buying services, you, - the customer -, should show in the list of values for the role of the customer, though you have not defined yourself as a customer in your Oracle Receivables customer list. In this case, you define the customer for a buy contract to select an organization.

To define the source, you refer to standard objects delivered with your standard application.

You cannot delete a source from a role, once you have defined it. To change your source assignment, you expire the current source or contact source assignment and add the new source assignment with a start date following the expiring date of the prior assignment.

Steps

1. Navigate to **Setup > Contract > Source**.
2. Query the Party role for which you want to enter the source. To create a new role, see [Creating a new role](#)
3. In the Party Source Tab, select a source for the party from the List of Values.
4. Select the intent, buy or sell.

Creating a new role

Steps

1. Enter a new Lookup code. Navigate to **Setup > Other > System > Lookup**.
2. Query the lookup code: OKC_CONTACT_ROLE
3. Enter an internal code and a meaning.

Customizing Events

When you customize Events, you are doing so at your own risk. Oracle support will not help you creating or debugging Actions, Functions or Outcomes you have created. If you encounter problems with customizations, you should make sure that the problem can be recreated using the standard Oracle objects.

Coding Action Assemblers

In Contracts business process whenever a defined action occurs, the necessary information about the attributes needs to be assembled and placed into the Events queue. You need to code an action assembler procedure for new actions.

Following are two examples to code an action assembler :

Example1 - When all the attribute information is readily available

This example accepts all attribute values as parameters. It builds the message table with the attribute name (element_name) and value (element_value). It also creates the record containing correlation for the action.

Then it calls the contracts advanced queuing procedure 'SEND_MESSAGE' to enqueue the message.

This example fits when the action occurs in a form where the form contains all the required information.

Pseudo Code

```
/*=====
Copyright (c) 1999 Oracle Corporation
Redwood Shores, California,USA
All rights reserved.

=====
SET VERIFY OFF

WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

CREATE OR REPLACE PACKAGE OKC_MY_ASMBLR_PVT AS

g_pkg_name CONSTANT varchar2(100) := 'OKC_MY_ASMBLR_PVT';

PROCEDURE acn_assemble(
/* mandatory parameters do not change */
p_api_version      IN NUMBER,
p_init_msg_list    IN VARCHAR2 DEFAULT OKC_API.G_FALSE,
x_return_status     OUT NOCOPY VARCHAR2,
x_msg_count         OUT NOCOPY NUMBER,
x_msg_data          OUT NOCOPY VARCHAR2,

/* parameters specific to this action assembler to be added here */
p_attribute1_value  IN DATATYPE,
p_attribute2_value  IN DATATYPE,
. . .
. . .
);


```

```
END OKC_MY_ASMBLR_PVT;
/

COMMIT;
EXIT;

/*
=====
Copyright (c) 1999 Oracle Corporation
Redwood Shores, California,USA
All rights reserved.

=====
*/
SET VERIFY OFF

WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

CREATE OR REPLACE PACKAGE BODY OKC_MY_ASMBLR_PVT AS

PROCEDURE acn_assemble(
    p_api_version      IN NUMBER,
    p_init_msg_list    IN VARCHAR2 DEFAULT OKC_API.G_FALSE,
    x_return_status     OUT NOCOPY VARCHAR2,
    x_msg_count        OUT NOCOPY NUMBER,
    x_msg_data         OUT NOCOPY VARCHAR2,

/* parameters specific to this action assembler to be added here */
    p_attribute1_value  IN/OUT DATATYPE,
    p_attribute2_value  IN/OUT DATATYPE,
    p_attribute3_value  IN/OUT DATATYPE,
    . . .
    . . .                  )  IS

    l_api_name      CONSTANT VARCHAR2(30)  := 'acn_assemble';
    l_api_version   CONSTANT NUMBER := 1.0;
    i               NUMBER := 1;
    l_corrid_rec    okc_aq_pvt.corrid_rec_typ;
    l_msg_tbl       okc_aq_pvt.msg_tab_typ;
    l_msg_count     number;
    l_msg_data      varchar2(1000);
    l_return_status  varchar2(1);
```

```

/* 'MY_ACTION_CORRELATION' is the correlation of the new action for
which this assembler is being coded */

CURSOR cur_corr_csr IS
SELECT aae.element_name,
       aae.format_mask format_mask
  FROM okc_actions_b acn,
       okc_action_attributes_b aae
 WHERE acn.id = aae.acn_id
   AND acn.correlation = 'MY_ACTION_CORRELATION' ;

BEGIN

  l_return_status := OKC_API.START_ACTIVITY
    (l_api_name
     ,p_init_msg_list
     ,'_PROCESS'
     ,x_return_status);

  IF l_return_status = OKC_API.G_RET_STS_UNEXP_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_UNEXPECTED_ERROR;
  ELSIF l_return_status = OKC_API.G_RET_STS_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_ERROR;
  END IF;

  l_msg_tbl := okc_aq_pvt.msg_tab_typ();
  FOR corr_rec IN cur_corr_csr
  LOOP
    /* 'ATTRIBUTE1_ELEMENT_NAME' is the 'element name' of the action attribute
for the newly defined action */
    IF corr_rec.element_name = 'ATTRIBUTE1_ELEMENT_NAME' THEN
      l_msg_tbl.extend;
      l_msg_tbl(i).element_name      := corr_rec.element_name;
      l_msg_tbl(i).element_value    := p_attribute1_value;
    ELSIF corr_rec.element_name = 'ATTRIBUTE2_ELEMENT_NAME' THEN
      l_msg_tbl.extend;
      l_msg_tbl(i).element_name      := corr_rec.element_name;
      l_msg_tbl(i).element_value    := p_attribute2_value;

    /* If the attribute is of date or number datatype and the datatype is specified
while defining the attribute then do the following */

    ELSIF corr_rec.element_name = 'ATTRIBUTE3_ELEMENT_NAME' THEN
      l_msg_tbl.extend;
      l_msg_tbl(i).element_name := corr_rec.element_name;

```

```
IF corr_rec.format_mask IS NOT NULL THEN
    l_msg_tbl(i).element_value := to_char(p_attribute3_value,
                                            corr_rec.format_mask);
ELSE
    l_msg_tbl(i).element_value := p_attribute3_value;
END IF;

ELSIF . . .
. . .
. . .
END IF;

i := i + 1;
END LOOP;

l_corrid_rec.corrid := 'MY_ACTION_CORRELATION' ;

/* This procedure puts the message on the events queue */
OKC_AQ_PUB.send_message(p_api_version      =>'1.0'
                        ,x_msg_count      =>l_msg_count
                        ,x_msg_data       =>l_msg_data
                        ,x_return_status  =>l_return_status
                        ,p_corrid_rec    =>l_corrid_rec
                        ,p_msg_tab        =>l_msg_tbl
                        ,p_queue_name     =>
                        okc_aq_pvt.g_event_queue_name);

IF l_return_status = OKC_API.G_RET_STS_UNEXP_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_UNEXPECTED_ERROR;
ELSIF l_return_status = OKC_API.G_RET_STS_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_ERROR;
END IF;

OKC_API.END_ACTIVITY(x_msg_count, x_msg_data);

EXCEPTION
    WHEN OKC_API.G_EXCEPTION_ERROR THEN
        x_return_status := OKC_API.HANDLE_EXCEPTIONS
            (l_api_name,
             G_PKG_NAME,
             'OKC_API.G_RET_STS_ERROR',
             x_msg_count,
             x_msg_data,
             '_PROCESS');
    WHEN OKC_API.G_EXCEPTION_UNEXPECTED_ERROR THEN
```

```

x_return_status := OKC_API.HANDLE_EXCEPTIONS
    (l_api_name,
     G_PKG_NAME,
     'OKC_API.G_RET_STS_UNEXP_ERROR',
     x_msg_count,
     x_msg_data,
     '_PROCESS');

WHEN OTHERS THEN
    x_return_status := OKC_API.HANDLE_EXCEPTIONS
        (l_api_name,
         G_PKG_NAME,
         'OTHERS',
         x_msg_count,
         x_msg_data,
         '_PROCESS');

END acn_assemble;

END OKC_MY_ACTION_ASMLR_PVT;
/
COMMIT;
EXIT;

```

Example2 - When attribute information needs to be assembled

This example accepts key information about the action for example like contract_id and collects the values for all the attributes from the database. It builds the message table with the attribute's element name (element_name) and value (element_value). It also creates the record containing correlation for the action.

Then it calls the contracts advanced queuing procedure SEND_MESSAGE to enqueue the message.

This example fits where the action occurs in the middle of a workflow process or in a concurrent program where information is not readily available.

Pseudo Code

```
/*=====
 | Copyright (c) 1999 Oracle Corporation
 | Redwood Shores, California, USA
 | All rights reserved
+=====*/
SET VERIFY OFF
WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

CREATE OR REPLACE PACKAGE OKC_MY_ACTION_ASMLR_PVT AS
g_pkg_name      CONSTANT varchar2(100) := 'OKC_MY_ACTION_ASMLR_PVT';

-----
-- PROCEDURE acn_assemble
-----
PROCEDURE acn_assemble(
/* mandatory parameters do not change */
    p_api_version          IN NUMBER,
    p_init_msg_list        IN VARCHAR2 DEFAULT OKC_API.G_FALSE,
    x_return_status         OUT NOCOPY VARCHAR2,
    x_msg_count             OUT NOCOPY NUMBER,
    x_msg_data              OUT NOCOPY VARCHAR2,
/* parameters essential to gather attribute values for this action to be added
here */
    p_my_param              IN/OUT DATATYPE,
    . . . );
END OKC_MY_ACTION_ASMLR_PVT;
/
commit;
exit;

/*=====
 | Copyright (c) 1999 Oracle Corporation
 | Redwood Shores, California, USA
 | All rights reserved.
+=====*/
SET VERIFY OFF
WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

CREATE OR REPLACE PACKAGE BODY OKC_MY_ACTION_ASMLR_PVT AS
g_pkg_name      CONSTANT varchar2(100) := 'OKC_MY_ACTION_ASMLR_PVT';

PROCEDURE acn_assemble(
```

```

/* mandatory parameters do not change */
p_api_version           IN NUMBER,
p_init_msg_list          IN VARCHAR2 DEFAULT OKC_API.G_FALSE,
x_return_status           OUT NOCOPY VARCHAR2,
x_msg_count               OUT NOCOPY NUMBER,
x_msg_data                OUT NOCOPY VARCHAR2,
/* parameters essential to gather attribute values for this action to be added
here */
p_my_param                IN/OUT DATATYPE
. . . ) IS
--
l_api_name                CONSTANT VARCHAR2(30) := 'ACN_ASSEMBLE';
l_api_version              NUMBER := 1.0;
l_init_msg_list             VARCHAR2(1) DEFAULT OKC_API.G_FALSE;
l_return_status varchar2(1) := OKC_API.G_RET_STS_SUCCESS;
--
/*Declare cursor or cursors to gather attribute values using
p_my_param */
CURSOR my_cur IS
SELECT attribute1_value,
       attribute2_value,
       . . . ,
       . . .
FROM  table1,
      table2,
      . . .
WHERE table1.id = p_my_param
AND   . . . ;
my_rec my_cur%ROWTYPE;

/* Declare cursor to get all attributes element_names, format_masks(for
date,number datatypes) for the newly created action*/
/* 'MY_ACTION_CORRELATION' is the correlation of the new action for
which this assembler is being coded */

CURSOR acn_cur IS
SELECT aae.element_name element_name,
       aae.format_mask format_mask
FROM   okc_actions_b acn,
       okc_action_attributes_b aae
WHERE  acn.id = aae.acn_id
AND    acn.correlation = 'MY_ACTION_CORRELATION';
acn_rec acn_cur%ROWTYPE;

--

```

```

l_rec okc_aq_pvt.corrid_rec_typ;
l_tbl okc_aq_pvt.msg_tab_typ;
i      NUMBER := 1;
--
-- BEGIN
-- call start_activity to create savepoint, check comptability
-- and initialize message list
  l_return_status := OKC_API.START_ACTIVITY(l_api_name
                                            ,l_init_msg_list
                                            ,'_PVT'
                                            ,x_return_status
                                            );
-- check if activity started successfully
  IF l_return_status = OKC_API.G_RET_STS_UNEXP_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_UNEXPECTED_ERROR;
  ELSIF l_return_status = OKC_API.G_RET_STS_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_ERROR;
  END IF;

  l_rec.corrid := 'MY_ACTION_CORRELATION';
  l_tbl := okc_aq_pvt.msg_tab_typ();

/*  'ATTRIBUTE1_ELEMENT_NAME' is the 'element name' of the action
attribute1 for the newly defined action */

FOR acn_rec IN acn_cur LOOP
  OPEN my_cur;
  FETCH my_cur INTO my_rec;
  IF acn_rec.element_name = 'ATTRIBUTE1_ELEMENT_NAME' THEN
    l_tbl.extend;
    l_tbl(i).element_name := acn_rec.element_name;
    l_tbl(i).element_value := my_rec.attribute1_value;
  ELSIF acn_rec.element_name = 'ATTRIBUTE2_ELEMENT_NAME' THEN
    l_tbl.extend;
    l_tbl(i).element_name := acn_rec.element_name;
    l_tbl(i).element_value := my_rec.attribute2_value;
  /* If the attribute is of date or number datatype and the datatype is specified
while defining the attribute then do the following */
  ELSIF acn_rec.element_name = 'ATTRIBUTE3_ELEMENT_NAME' THEN
    l_tbl.extend;
    l_tbl(i).element_name := acn_rec.element_name;
    IF acn_rec.format_mask IS NOT NULL THEN
      l_tbl(i).element_value := to_char(my_rec.attribute3_value,
                                         acn_rec.format_mask);
    ELSE

```

```

        l_tbl(i).element_value := my_rec.attribute3_value;
    END IF;

    ELSIF . . .
    . . .
    END IF;
    i := i+1;
    CLOSE my_cur;
END LOOP;

/* This procedure puts the message on the events queue */
OKC_aq_pvt.send_message(p_api_version => l_api_version
                        ,x_msg_count => x_msg_count
                        ,x_msg_data => x_msg_data
                        ,x_return_status => x_return_status
                        ,p_corrid_rec => l_rec
                        ,p_msg_tab => l_tbl
                        ,p_queue_name =>
                        okc_aq_pvt.g_event_queue_name
                        );
-- check if activity started successfully
IF l_return_status = OKC_API.G_RET_STS_UNEXP_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_UNEXPECTED_ERROR;
ELSIF l_return_status = OKC_API.G_RET_STS_ERROR THEN
    RAISE OKC_API.G_EXCEPTION_ERROR;
END IF;

OKC_API.END_ACTIVITY(x_msg_count, x_msg_data);

EXCEPTION
    WHEN OKC_API.G_EXCEPTION_ERROR THEN
        x_return_status := OKC_API.HANDLE_EXCEPTIONS
        (l_api_name,
        G_PKG_NAME,
        'OKC_API.G_RET_STS_ERROR',
        x_msg_count,
        x_msg_data,
        '_PVT');
    WHEN OKC_API.G_EXCEPTION_UNEXPECTED_ERROR THEN
        x_return_status := OKC_API.HANDLE_EXCEPTIONS
        (l_api_name,
        G_PKG_NAME,
        'OKC_API.G_RET_STS_UNEXP_ERROR',
        x_msg_count,
        x_msg_data,
        '_PVT');

```

```
        '_PVT');
WHEN OTHERS THEN
  x_return_status := OKC_API.HANDLE_EXCEPTIONS
  (l_api_name,
   G_PKG_NAME,
   'OTHERS',
   x_msg_count,
   x_msg_data,
   '_PVT');
END acn_assemble;

END OKC_K_MY_ACTION_PVT;
/
commit;
exit;
```

Coding Functions for Function Expressions in Condition Lines

The Functions used in the Function Tab of the Conditions Form should always return 'T' or 'F' as a return value. Following example shows how a function expression can be used to determine if a contract belongs to a particular customer. While entering the parameters for the function expression in conditions form, the user can enter a value instead of selecting an attributes for parameters 'p_role' and 'p_customer_name'. This way he or she can check if the contract belongs to a customer they are interested in. The condition evaluator executes this function with the user entered values at runtime and returns 'T' or 'F', based on which certain outcomes can be executed.

```
FUNCTION customer_exists(p_kid IN NUMBER,
                         p_customer_name IN VARCHAR2,
                         p_role      IN VARCHAR2)
RETURN VARCHAR2 IS

CURSOR cust_cur IS
select 'X'
from  okc_k_party_roles_v role,
      okx_parties_v party
where role.object1_id1 = party.id1
and   role.object1_id2 = party.id2
and   role.jtot_object1_code in ('OKX_PARTY','OKX_OPERUNIT')
and   upper(role.rle_code) = upper(p_role)
and   role.chr_id = p_kid
and   party.name = p_customer_name;
cust_rec  cust_cur%ROWTYPE;
```

```

BEGIN
  OPEN cust_cur;
  FETCH cust_cur INTO cust_rec;
  IF cust_cur%FOUND THEN
    RETURN('T');
  ELSE
    RETURN('F');
  END IF;
  CLOSE cust_cur;
EXCEPTION
  WHEN others THEN
    RETURN('F');
END customer_exists;

```

Coding Outcomes for Conditions

All Outcomes that do not have the standard set of parameters should have a wrapper API which provides these parameters and default values. All outcomes should have p_init_msg_list set to OKC_API.G_TRUE to initialize message stack.

p_api_version	IN NUMBER,
p_init_msg_list	IN VARCHAR2 DEFAULT OKC_API.G_TRUE,
x_return_status	OUT NOCOPY VARCHAR2,
x_msg_count	OUT NOCOPY NUMBER,
x_msg_data	OUT NOCOPY VARCHAR2,

Outcomes should not commit transaction. Especially if called in a workflow or a concurrent process or by any other pl/sql process where other activities are involved. The Outcomes or the wrapper program should always contain standard exception block.

```

EXCEPTION
  WHEN OKC_API.G_EXCEPTION_ERROR THEN
    x_return_status := OKC_API.HANDLE_EXCEPTIONS
    (l_api_name,
     G_PKG_NAME,
     'OKC_API.G_RET_STS_ERROR',
     x_msg_count,
     x_msg_data,
     '_PVT');
  WHEN OKC_API.G_EXCEPTION_UNEXPECTED_ERROR THEN
    x_return_status := OKC_API.HANDLE_EXCEPTIONS
    (l_api_name,

```

```
    G_PKG_NAME,
    'OKC_API.G_RET_STS_UNEXP_ERROR',
    x_msg_count,
    x_msg_data,
    '_PVT');
WHEN OTHERS THEN
    x_return_status := OKC_API.HANDLE_EXCEPTIONS
    (l_api_name,
    G_PKG_NAME,
    'OTHERS',
    x_msg_count,
    x_msg_data,
    '_PVT');
END;
```

Any messages specific to the outcome should be created in the FND_NEW_MESSAGES table. If the Outcome is a Workflow, then it need not have any out parameters. The errors in workflow will be handled by standard workflow functionality.

Rules when creating Actions

1. All Actions defined should have at least one attribute.
2. Counter based Actions should have COUNTER_GROUP_ID and COUNTER_GROUP_LOG_ID as mandatory attributes.

Using Contracts Core

This topic group provides process-oriented, task-based procedures for using the application to perform essential business tasks.

Working with the Launchpad and the Contract Navigator

Use the Launchpad to view contracts in your inbox, recently modified contracts, or bookmarked contracts. See [Understanding Roles and Contact Sources](#) for details about the Contract Launchpad.

Use the Launchpad and the Contract Navigator tabs to access a variety of functions related to a contract. You can access most of these functions from the Contract Navigator by right clicking on a contract or by selecting Tools->Options from the menu bar.

Select one of the functions in this list for a more detailed procedure.

Functions

[Creating a New Contract](#)

[Creating a Contract from a Template](#)

[Opening a Contract](#)

[Creating a New Version](#)

[Viewing the History of a Contract](#)

[Creating a Change Request](#)

[Closing the Change Request for a Contract](#)

[Copying a Contract](#)

[Creating a Subcontract](#)

[Renewing a Contract](#)

[Changing the Status of a Contract](#)

[Extending a Contract](#)

[Extending a Contract Line](#)

[Terminating Contract Lines](#)

[Viewing the Schedule](#)

[Communications](#)

[Authoring Contracts](#)

Creating a New Contract

You can create a new contract manually or from a template. Use this procedure to create a new contract manually. For more information on the Authoring Screen and creating a contract, see [Authoring Contracts](#)

Prerequisites

None

Steps

1. In the Navigator, select Launchpad. The Launchpad screen opens.
2. Select the Contract Navigator Tab.
3. Right-click to display a popup menu, select **New**.
4. Click **Create a new Contract Manually**.
5. Select a category from the list of values.
6. Click **Create**. You now are in the Authoring Screen.

Creating a Contract from a Template

Use this procedure to create a contract from a template. For more information on the Authoring Screen and creating a contract, see [Authoring Contracts](#)

Prerequisites

None

Steps

1. In the Navigator, select **Launchpad**.
2. Select the Contact Navigator tab.
3. Right click and select **New** from the list of options.

4. Click **Create a Contract from a Template**.
5. Select a template from the list of values.
6. Optionally, click the **Find** button to find an existing template.
 - f. In the Search Templates Window you can enter criteria to find templates.
 - g. Select a template and double-click to open it, or click **Open**.
7. Click the **Create** button. You now are in the Authoring Screen.

Opening a Contract

Use the procedure to open a contract. For more information on the Authoring Screen and creating a contract, see [Authoring Contracts](#)

Prerequisites

None

Steps

1. In the Navigator, select **Launchpad**. The Launchpad screen opens.
2. Select the Contract Navigator tab.
3. Select a contract group.
4. Right-click a contract to display a popup menu, select **Open**. The Authoring Screen is displayed.

Creating a New Version

Use this procedure to create a new version of a contract.

Prerequisites

None

Steps

1. In the Navigator, select **Launchpad**. The Launchpad screen opens.
2. Select the Contract Navigator tab.
3. In the left window, select a contract group.

4. Right-click a contract to display a popup menu, select **Create New Version**. This creates a new version of your contract.

Guidelines

When you create a new version, the digit in front of the period (the main version), is increased by one. The minor version is reset to zero.

Viewing the History of a Contract

Use this procedure to view older versions of a contract.

Prerequisites

None

Steps

1. In the Navigator, select Launchpad.
2. In the Contract Navigator, select a contract group.
3. Double click on the contract for which you want to see a previous version. A summary window appears.
4. Select the **History** tab.
5. In the Select list, select the **Contract Versions** to see all versions. All major versions are displayed.
5. Select the version you want to open.
6. Click **Open**.

You now can view the version of the contract.

Creating a Change Request

Change Requests are a formal way to document changes to a contract. A request is approved by the approver defined in the workflow definition of the change request. Once your change request is approved, the administrator entered in the change request receives a notification that he/she can change the contract.

An internal key gives the change request administrator update access to the contract. When the administrator changes, it is locked for other change requests until the change request is closed. See [Closing the Change Request for a contract](#).

Use the procedure to create a change request.

Prerequisites

The contract must be in status active, and no other change requests may be in progress.

Steps

1. In the Contract Navigator, select a contract group.
2. Select the contract you want to change.
3. Right click on the contract. A popup list appears.
4. Select **Create Change Request** from the popup list. The **Create a Change Request** Window opens.
5. Enter the requestor's name and add a description for the approver.
6. Enter the administrator's name. This person will receive an workflow notification when changes are ready to be made.
7. Enter the request date, and optionally, modify the request date and/or effective dates.
8. Record the lines where you are requesting a change and the exact nature of the change.
9. Select the **Version** checkbox if you want to version the changed contract.
10. Record if a signature is required and add any other parties and when they approved the changes.

Closing the Change Request for a Contract

When you receive a notification to apply changes, open the contract and apply the changes as described in the change request notification.

Once you start editing the contract, you automatically lock the contract for other users with an approved change request.

Use this procedure to complete the changes and unlock the contract.

Prerequisites

None.

Steps

1. In the Navigator, select **Launchpad**.
2. Select the Contracts Navigator tab.
3. Select the contract for which you have applied the changes
4. Double-click on the contract to open the Execution window, and from the History tab select the change request.
5. Optionally, right click on the contract and select Change Request from the popup list. This opens the change request menu where you can then query for a change request.
6. Enter the current date in the Apply Date field and save your entry. That closes the change request and releases the lock on the contract.

Copying a Contract

There are several ways to copy a contract, parts of a contract, or multiple contracts.

Prerequisites

None.

Steps

1. There are several ways to navigate to the copy window:
 - In the Contract Navigator, select a contract group, a contract, and select **Copy** from the popup list.
 - Open an existing contract, and in the Authoring window, select **Copy from Contract/Template** from the Action menu.
2. In the Copy Window, enter the new contract name, and optionally, a contract modifier.
3. From the left side, select the contract or part of the contract you want to copy.
4. Click the double right-arrow button to move the existing contract or parts on the left side to the right side. Anything on the right side is considered part of the new contract.
5. Click the double left-arrow button to move a new contract or parts of a contract back to the existing contract area on the left side.
6. Click **Create**. Your new contract with copied contract information is now saved.

7. You can then make changes to the newly created contract in the same way that you would when authoring a contract.

Creating a Subcontract

In order to fulfill obligations of an existing contract, a party may need to establish a new contract with third parties to provide some or all of the items (or materials, or deliverables, etc.) required in the original contract. This is called creating a subcontract. You can copy a contract or elements of a contract into a subcontract. Use this procedure to create a subcontract.

See [Defining Sources](#).

Prerequisites

A finalized, active contract has to be in place.

Steps

1. In the Contract Navigator, select a contract group.
2. Right-click a contract to display a popup menu. The Tools menu also has these options.
3. Select **Subcontract** from the popup menu. The **Subcontract** window opens.
4. On the right side, enter the name and optionally a modifier of the new contract.
5. Select a category for the new subcontract.
6. Optionally correct the intent and enter the perspective of the contract.
7. Click the Keep References Checkbox if you want to reference to the subcontracting contract instead of copying components. You cannot change referenced components in the new subcontracted contract.
8. Highlight the desired components in left navigator, click the '>' button and click on the 'Create' button in the subcontract components selection popup.

You can make changes to the new contract in the Contract Authoring screen.

Renewing a Contract

A renewed contract does not affect the current contract, but rather it creates a new contract by copying the current contract. By choosing a new date range that must start after the expiration date of the current contract, you renew a contract. Use this procedure to renew a contract.

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.
2. Right-click a contract to display a popup menu. The Tools menu also has these options.
3. Select **Renew** from the popup menu to open the **Renew Window**.
4. To keep the new contract distinguishable, the modifier is defaulted to the current date. You can change the modifier to another value.
5. The new start date is defaulted to the date following the end date of the current contract. Either select an end date for the renewed contract or enter values for duration and period. For example, to renew a contract for six months, enter 6 in the duration and select month in the period field.
6. Click **Renew**.

Changing the Status of a Contract

The current status determines the operations allowed against the contract and the status type you can change the contract to. For more information, see [Understanding Status and Operations](#). Use this procedure to change the status or status type of a contract.

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.

2. Right-click a contract to display a popup menu, select **Change Status**. The Change Status Window appears.
3. Select the new status from the list of values.

Extending a Contract

This only changes the end date of a contract. The contract then is scheduled to expire on the new end date. Use this procedure to extend a contract.

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.
2. Right-click a contract to view the popup menu. The Tools menu also has these options
3. Select **Extend** from the popup menu to open the **Extend** window.
4. Either select a new end date for the contract or enter values for duration and period. For example, to extend a contract for further six months, enter 6 in the duration and select month in the period field.
5. Click **Extend**.

Extending a Contract Line

Extending a line changes the end date of a contract. The extended line then is scheduled to expire on the new end date. Use this procedure to extend a line on an active contract.

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.
2. Double click on a contract to see the overview.
3. Select the line you want to extend.

4. Right click to open the popup menu. Select **Extend** to open the **Extend** window.
5. Either select a new end date for the line or enter values for duration and period. For example, to extend a contract line for further six months, enter 6 in the duration and select month in the period field.
6. Click **Extend**.

Guidelines

If you extend a contract line beyond the end date of a contract, the end date of the contract is automatically updated to the end date of the extended line.

Terminating a contract

Use this procedure to terminate a contract that is active or effective and on hold.

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.
2. Right-click a contract and a popup list appears.
3. Select **Terminate** to open the **Terminate** window.
4. Enter a termination date and a termination reason.
5. Click **Terminate**.

Guidelines

If you want to add more reasons for a termination, your system administrator can add more options in the Lookup Codes, see [Defining Termination Reasons](#).

Terminating Contract Lines

Use this procedure to terminate a contract line in an active contract. If you want to add more reasons for a termination, your system administrator can add more options in the Lookup Codes, see [Defining Termination Reasons](#).

Prerequisites

None

Steps

1. In the Contract Navigator, select a contract group.
2. Double-click on a contract to see the overview.
3. Select the line you want to terminate.
4. Right-click to open the popup menu. Select **Terminate** to open the **Terminate** window.
5. Enter a termination date and a termination reason.
6. Click **Terminate**.

Viewing the Schedule

Use the Schedule to get an overview of events scheduled.

Steps:

1. Navigate to **Launchpad**.
2. Select the Contract Navigator Tab.
3. Select the contract for which you want to see scheduled tasks or completed outcomes.
4. Double-click on the contract and then select the Schedule tab.

You can use this form to record contingent events and the completion of tasks.

Communications

The Communications tab allows you to record and view communications between the parties within a contract. Such communications can range from phone calls through e-mails to formal letters sent.

Prerequisites

None.

Steps

1. In the Navigator, select the **Launchpad**.
2. Select the Contract Navigator Tab.
3. Select the contract for which you want to see or enter communications.
4. Double-click on the contract and select the Communications tab.

Authoring Contracts

From the Authoring Contracts screen, you can create or modify a contract.

Important: Before selecting a tab from the contracts screen, you must enter general contract information in the header area:

- Contract number: enter the contract number in this field
- Modifier: Use this option if the contract number you have chosen already exists to distinguish contracts with a contract name otherwise the same.
- Select an intent - buy or sell -.
- Enter an end date
- You must at least enter one Group name in the details sub tab.

You can perform the following functions from the Authoring window:

[Entering Summary Information](#)

[Entering Contract Parties](#)

[Specifying Contract Line Item Information](#)

[Entering Article Information in a Contract](#)

[Managing Rules](#)

[Checking the Contract](#)

[Submit for Approval](#)

Conditions

Templates

At any time you can save an incomplete contract as a template, using Save as Template from the tools menu.

Entering Summary Information

The Summary tab has three subtabs: Text, Details, and Administration.

Using this subtabs you can do the following:

- Text subtab: Enter comments that may contain special or important instructions. You can also enter a description.
- Details subtab: Enter the Quality Assurance procedure and you can add groups.
- Administration subtab: Enter the approval workflow and provide access to other groups or persons

You can also enter condition types and actions, quality check a contract or submit it for approval using the buttons at the bottom of the window.

At any time, you can check if your contract passes quality assurance. The result of the on-line quality assurance check is displayed as soon as it completes. There is no need to run the quality assurance before you can submit the contract for approval, since the quality assurance is automatically run before submitting for approval.

Use this procedure to enter summary information.

Prerequisites

None.

Steps

1. In the Effective region of the Summary tab, enter this information:
 - Intent (buy or sell)
 - Perspective
 - Start and end dates
2. In the Details tab, enter the approval workflow information.

- a. Choose QA Checklist from the list of values. You may select another QA checklist, however, Contracts automatically executes the default checklist in addition to your own checklist.
 - b. If it applies to the contract, select **Auto Renew**. You must also provide the number of days before the contract is automatically renewed by a concurrent program.
 - c. Select a group. You must assign the contract to at least one group.
3. In the Administration you select your approval workflow. You can give access to groups or contract users in the Access region.

Entering Contract Parties

In the Parties tab, you can list all the personnel involved with the creation and management of the contract. The Parties tab also contains all the relevant sales contacts information. The information you enter here depends on the category and intent. Use this procedure to enter contract parties.

Prerequisites

None.

Steps

1. Navigate to the Parties tab.
2. Enter the role information.
3. Enter the name. You may add an alias or awakening-as' identifier.
4. Navigate to the Affirmative Action region.
5. Enter the minority group, if applicable.
6. Enter the small business, if applicable.
7. Enter the woman-owned information if applicable.
8. Navigate to the Party Contacts region to enter a contact, sequence, and a name of the contact. The sequence determines order the contacts should appear.

Specifying Contract Line Item Information

In the Line Items tab, you can detail all the contract line items, which are the contract deliverables and relevant items. You can enter more than one item per

contract line on the line items tab in the subline region. For more information on lines, see Understanding Line Styles. Use this procedure to enter details for a line item.

Prerequisites

None.

Steps

1. In the Level region, provide the information for Line #.
2. Enter the line style and a number for the Display Sequence.
3. Depending on the line style definition, you may enter the description or select a description from the list of values.

Note: If the your line item is priced at the top level, you can enter quantity, unit of measure and price information.

4. Enter the unit of measure.
5. Enter the unit price. The extended price is calculated from the quantity time the unit price. The negotiated price allows you to override the extended price.

Note: The Line Items tab has four subtabs: Sublines, Items, Rules, and Parties. You can use the subtabs to add and exclude data specific to a line.

6. Optionally, depending on the line type you selected, use the Sublines tab to enter more detail in the sublines.
You can actually build an infinite level of sublines. This window allows you to view and edit the hierarchy at any level. Navigate through levels by using the Drill Up and Drill Down buttons.
7. Use the Items tab to store more details about an item. Use the exception flag to document, that an item may be excluded from the contract line.
8. Enter rules on lines the same way rules are entered on the contract.

9. Use the Parties tab to add line specific information that is not valid for the entire contract. Entering a party in a subline is similar to entering a party information for the contract.

References

Entering Rules

Entering Article Information in a Contract

An article is the text that describes and details the terms and conditions that are attached to a contract. You can enter two kinds of articles while authoring a contract: standard and nonstandard articles. See [Understanding Contract Articles](#).

These are types of articles you can enter:

[Entering a Standard Article](#)

[Entering a Non-Standard Article by copying a Standard Article](#)

[Entering a Non-Standard Article](#)

Use these procedures to enter contract articles.

Entering a Standard Article

Use this procedure to enter a standard article.

Prerequisites

None.

Steps

1. In the Article window, leave the standard flag checked.
2. Click on the name field, select the article list from which want to select an article.
3. Double-click on the article set and select an article. If you know your article, you can type in fragments of the article to reduce the list of values matching your search criteria. This second method by passes the navigation through article sets.
4. Reference an article from the article list. You can add variations in the text field when you click **Show Text**.

Entering a Non-Standard Article by Copying a Standard Article

Use this procedure to copy an existing standard article and modify it.

Prerequisites

None.

Steps

1. In the Article window, leave the standard flag checked.
2. Click the name field, and select an article set from the drop-down list.
3. Double-click on the article set and select an article. If you know your article, you can type in fragments of the article to reduce the list of values matching your search criteria. This second method bypasses the navigation through article sets.
4. Copy an article from the article list. You can modify the text when you click **Show Text**. The standard flag is automatically unchecked when you choose to copy an article instead of referencing it.

Entering a Non-Standard Article

Use this procedure to enter a non-standard article.

Prerequisites

A contract must have been created, even if it is only in its initial authoring stages.

Steps

1. Deselect the Standard checkbox.
2. Enter a name to describe your non-standard article.
3. You can enter the article text when you click **Show Text**.

Note: It is important to note that as part of the initial set-up of the **Standard Article** feature, you can set and define rules. So whenever you use a **Standard Article** within a contract in any way, all rules that have been associated with the **Standard Article** are copied into the new contract being authored.

Entering Rules

In the Rules tab you can add rule groups. Edit the rules using the Edit button. You can use the Rules Editor to add, change, or edit rules at any level that the rule is instituted at. See [Understanding Contract Rules](#). Use this procedure to add rule groups and edit rules.

Prerequisites

None

Steps

1. Click **Edit Rules** from either the Rules tab of the contract, the rules tab of the line items or from the articles tab. The Rules Editor appears.
2. Select a rule group from the list of values.
3. A tab with the rule group name is displayed in the rules editor. Fill in the fields. A rule group may have more than one rule. When you close the editor, each rule and its contents are displayed in one line.
4. The optional flag is a display-only field to indicate if the rule is a mandatory or optional entry.
5. Only for billing rules the Party button is relevant. Use this function to mark the billing subject (the billing party) or the object (the billed party).

Checking the Contract

Click the **Check Contract** button to run an on-line quality assurance (QA).

At any time, you can check if your contract passes quality assurance. The result of the on-line quality assurance check is displayed as soon as it completes. There is no need to run the quality assurance before you can submit the contract for approval.

Submitting for Approval

Click the **Submit for Approval** button to submit the contract for approval.

When you submit a contract for approval, quality assurance is automatically executed. The contract is only submitted if it passes quality assurance. The result of the on-line quality assurance check appears as soon as it completes. You then may choose to continue with the submission.

Using Conditions

Click **Conditions** to specify conditions for the contract. For more information on Conditions, see [Understanding Events](#).

You now can either copy condition templates into your contract or you can create a new contract specific condition.

Copying from a Condition Template

Use this procedure to copy a condition template into your contract.

Prerequisites

None.

Steps

1. Select **Copy From Template** from the Actions Menu.
2. Enter the condition template name or Open the list of values to see the Condition Template window. Select your condition template.
3. Modify or amend the condition for the current contract.

Creating a Contract Specific Condition

Use this procedure to create a contract specific condition for your contract.

Prerequisites

None.

Steps

1. Enter a name and a description.
2. Check **Create a Task** if you want to create a task for the CRM Task Manager. Creating a task requires entering a task owner.
3. Check the condition type. For Action you must enter an action, for Date you must enter the number of days, qualify before or after, and you must enter the triggering date.
4. For condition of type action choose if the event is evaluated only once or each time the condition is met. For example: Check the flag if you want to evaluate the condition to true only if the counter exceeds 1,000 for the first time. Leave

the flag unchecked if you want the condition to evaluate to true each time the counter reaches 1,000.

5. Build your condition lines.

Each condition may consist of one or many condition lines, with the sequence determining the order in which the lines are processed. You can enter Expressions, Counter or Function lines.

Expressions:

The left value only allows you to enter attributes defined in the action attributes.

Counter:

Select Product or Service and choose a description of the service or product.

If you select a counter group, then your left value's List of Values lists counter group types. If you choose a counter group master, then your left value's List of Values shows the related counter groups.

Functions:

Select Functions declared in the Process Definitions Window. To enter fix values for parameters, click the **Parameter** button.

6. Display the condition lines and check validity.

Click on **Show Condition** to display all condition lines and check their syntactical validity on-line. The checkbox valid is display-only and gives you a quick overview if the condition has validated successfully.

7. Enter **Outcomes defined in the Process Definitions Window.**

8. Click **Parameter to assign fixed values or action attribute values to the outcome parameters.**

Reviewing Errors from Asynchronous Processes

This window displays all messages of processes that have failed. Usually this window is used by your support representative to resolve issues.