Siebel

Pricing Administration Guide

March 2021



Siebel
Pricing Administration Guide

March 2021

Part Number: F12772-03

Copyright © 2021, Oracle and/or its affiliates.

Authors: Siebel Information Development Team

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

Contents

Preface i

1 What's New in This Release 1

What's New in Siebel Pricing Administration Guide, Siebel CRM 21.3 Update 1
What's New in Siebel Pricing Administration Guide, Siebel CRM 20.1 Update 1
What's New in Siebel Pricing Administration Guide, Siebel CRM 19.1 Update 1

2 Overview of Pricing Administration 3

Overview of Pricing Administration 3
About Pricing Administration 3
Activating Pricing Workflows 4
Enhancements Introduced in Siebel Pricer Version 7.8 4
About the Processing Order of Price Adjustments 6
Clearing Pricing Data from the Cache 6
Which Type of Promotion Pricing to Use 6

3 Creating and Assigning Price Lists 9

Creating and Assigning Price Lists 9
About Price Lists 9
About Sales Products and Service Products 10
Price List Prerequisites 10
Process of Creating a Price List 11
About Setting Up Service Pricing 17
About Effective Dates for Pricing Line Items 20
Giving a Product Multiple Prices with Different Effective Dates 22
Creating a Price List Line Item for a Simple Product Bundle 23
Copying and Modifying a Price List 23
Copying and Transforming a Price List 24
Importing Price Lists 25
Assigning Price Lists to Users 25
Enabling Price List-Specific and Line Item-Specific Pricing Logic 26

4 Creating and Assigning Rate Lists 27
Creating and Assigning Rate Lists 27
About Rate Lists 27
Process of Creating Rate Lists 28
Copying and Modifying a Rate List 30
Copying and Transforming a Rate List 31
Assigning a Rate List to a User 32

5 Creating and Using Cost Lists 33
Creating and Using Cost Lists 33
About Cost Lists 33
Process of Creating Cost Lists 33
Using Asset Mapping to Value Assets 38

6 Creating Volume Discounts 41
Creating Volume Discounts 41
About Volume Discounts 41
Process of Setting Up a Volume Discount 42

7 Creating Aggregate Discounts 47
Creating Aggregate Discounts 47
About Aggregate Discounts 47
Buy and Receive Products for Aggregate Discounts 48
Process of Creating Aggregate Discounts 48
Aggregate Discounts for Buy One, Get One Free Discounts 52
Aggregate Discounts for Products with Components 53

8 Creating Product Promotions 55
Creating Product Promotions 55
About Product Promotions 55
Process of Creating Product Promotions 56
Creating Product Promotions That Upgrade Existing Promotions 71
Viewing a Summary of a Product Promotion 72
Copying a Bundled Promotion or Coupon 72
Managing Versions of Product Promotions 73

Viewing the Schedules of All Product Promotions 73
About Product Promotions for Customizable Products 74

9 Creating Product Offers 79
Creating Product Offers 79
About Product Offers 79
About Versions of Product Offers 80
About Actions for Responses to Product Offers 81
Process of Creating Product Offers 83
Product Offers with Named Parameters 93
Example of Creating Product Offers with Parent Actions 95
Modifying Product Offers 97
Product Offers for Multiple Products 98
Viewing Product Offers for an Account 98
Tasks for Suppressing Product Offer Recommendations 98
Migrating Product Offers Using Workspace Projects 100
Scenarios for Product Offers for Multiple Products Using Web Services 101
Workflow Reference for Product Offers 108

10 Creating Discount Matrices 139
Creating Discount Matrices 139
About Discount Matrices 139
About Discount Matrix Dimensions 140
About Discount Matrix Types 140
Process of Creating a Discount Matrix 144
Discount Matrix Integration 147
About Configuring Discount Matrices 148

11 Creating Attribute Adjustments 153
Creating Attribute Adjustments 153
About Attribute Pricing 153
Attribute Adjustments Design 154
About the Attribute Adjustments View 154
Variable Maps in Attribute Adjustments 156
How Attribute Adjustments Apply Rules 157
Single and Multiple Type Attribute Adjustments 159

Process of Creating an Attribute Adjustment 163
Example of Creating an Attribute Adjustment Based on Product Attributes 169
Example of Creating an Attribute Adjustment with Nonproduct Attributes 174
Guidelines for Creating Attribute Adjustments 182
How the Pricing Procedure Uses Attribute Adjustments 183
Upgrade Considerations for Attribute Adjustments 183

12 Setting Up Pricing for Products with Components 185
Setting Up Pricing for Products with Components 185
About Component-Based Pricing 185
Impact of Product Versions and Price List Line Item Effectivity on Component-Based Pricing 186
Viewing a Product's Components in the Pricing Designer 187
About Setting Up Component-Based Pricing 188
Viewing and Changing the Reference Price 190
Defining a Relationship-Based Price Adjustment 194
Verifying Component Prices Across Time Intervals 196

13 Creating Price Reports 199
Creating Price Reports 199
About Pricing Reports 199
Running Reports 199
About Creating and Modifying Reports 199

14 Pricing in the Run-Time Application 201
Pricing in the Run-Time Application 201
Pricing Waterfalls 201
Contract-Based Pricing 203
Multiple Price Types and Totals 206
About Configuring Pricing Fields in Catalog, eSales, and Configurator 206
Multiple Currencies in a Quote, Order, or Agreement 207
Spread Discount Pricing 207
Totaling and Rollups of Pricing 210

15 Siebel Pricer Technical Reference 211
Siebel Pricer Technical Reference 211
Siebel Pricer Architecture 211

Pricing Procedures and Workflow References 213
Switching from Basic to Dynamic Pricing Procedure 231
Modifying Signals from Basic Pricing Procedure to Dynamic Pricing Procedure 233
Miscellaneous Pricing Workflows 246

16 Siebel Pricer Deployment and Integration 249

Siebel Pricer Deployment and Integration 249
About Pricing Integration 249
Integration Methods for Siebel Pricer 251
Internal Integration of Siebel Pricer 251
External Integration of Siebel Pricer Using Pricer as a Service 267
Siebel Pricing Manager API Reference 276
Workflows for External Pricing API 279
Cache Configuration Settings 281

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *http://docs.oracle.com/*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

1 What's New in This Release

What's New in Siebel Pricing Administration Guide, Siebel CRM 21.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
Constraining the Products Available for a Component of a Product Promotion	Modified topic. Added information on modifying the minimum and maximum cardinality for the Relationship Port.

What’s New in Siebel Pricing Administration Guide, Siebel CRM 20.1 Update

No new features have been added to this guide for this release. This guide has been updated only to correct or remove obsolete product and component terms.

What’s New in Siebel Pricing Administration Guide, Siebel CRM 19.1 Update

No new features have been added to this guide for this release. This guide has been updated only to correct or remove obsolete product and component terms.

2 Overview of Pricing Administration

Overview of Pricing Administration

This chapter gives a general overview of pricing administration for Oracle's Siebel Pricer application, and it describes how you log on as Siebel Administrator. It also describes the processing order of price adjustments. This chapter covers the following topics:

- *About Pricing Administration*
- *Activating Pricing Workflows*
- *Enhancements Introduced in Siebel Pricer Version 7.8*
- *About the Processing Order of Price Adjustments*
- *Clearing Pricing Data from the Cache*
- *Which Type of Promotion Pricing to Use*

About Pricing Administration

Siebel Pricer allows you to work with the following:

- **Price lists, rate lists, and cost lists.** Create price lists, rate lists, and cost lists. For more information, see the following chapters:
  - *Creating and Assigning Price Lists*
  - *Creating and Assigning Rate Lists*
  - *Creating and Using Cost Lists*
- **Volume discounts.** Give customers a discount if they buy in volume. For example, if a customer buys more than one hundred units, then the customer can get a 10 percent discount. For more information, see *Creating Volume Discounts*.
- **Promotion pricing.** There are two types as follows:
  - **Aggregate discounts.** Give customers a discount if they buy a group of products. For more information, see *Creating Aggregate Discounts*.
  - **Product promotions.** Create promotions that can be selected in quotes, orders, and catalogs, can be tracked as assets, can have commitments tied to them, and can have rules and penalties defined for future changes. For more information, see *Creating Product Promotions*.
- **Discount matrices.** Create a matrix of rules to define price adjustments. For more information, see *Creating Discount Matrices*.
- **Attribute adjustments.** Create a matrix of rules to define attribute-based pricing. Attribute adjustment provides flexibility to map rules to attributes or fields in any business component. For more information, see *Creating Attribute Adjustments*.

- **Products with components.** Define pricing for products whose components customers can choose. For example, when a customer buys a computer, the customer can choose to add peripherals such as a CD-ROM drive. For more information, see *Setting Up Pricing for Products with Components*.
- **Pricing reports.** Create a variety of reports to track your pricing policies. For more information, see *Creating Pricing Reports*.
- **Customizing pricing logic.** Pricing logic is based on pricing procedures, so it is fully configurable. For more information, see *Siebel Pricer Technical Reference*.

Activating Pricing Workflows

To make pricing functionality available to users, you must activate the pricing workflows. For a complete list of these workflows, see *Siebel Pricer Technical Reference*. For information about how to activate workflows, see *Siebel Business Process Framework: Workflow Guide* .

Enhancements Introduced in Siebel Pricer Version 7.8

The most important enhancements introduced in Siebel Pricer 7.8 are summarized in this topic.

Pricing Is Based on Pricing Procedures

Pricing logic is defined using pricing procedures. These procedures are based on the Product Selection and Pricing (PSP) engine. The advantage of using the PSP engine is that pricing is fully configurable. For more information about the PSP engine, see *Siebel Order Management Infrastructure Guide* . For information about the procedures used for pricing, see *Siebel Pricer Technical Reference*.

Because you can use pricing procedures to define pricing logic, pricing models and pricing factors are no longer used in Siebel Pricer version 7.8 or later.

Effective Dates for Line Items

You can set the effective dates for the line items of price lists, cost lists, discount matrices, component product price adjustments, or service prices. For more information, see *Giving a Product Multiple Prices with Different Effective Dates*.

Discount Matrices

Matrix-based administration of price adjustments using discount matrices is allowed. For more information, see *Creating Discount Matrices*.

Attribute Adjustments

You use attribute adjustments view to specify pricing for products with attributes. For more information, see *Creating Attribute Adjustments*.

Price Waterfalls

Call center agents and sales representatives can view a price waterfall that shows the calculations that were used to arrive at the net price that is displayed in a quote, order, or agreement. For more information, see *Pricing Waterfalls*.

Enhancements in Pricing for Component Products

The following pricing enhancements for component products are available:

- Product versions allow you to create multiple future release versions for products.
- Line item effective dates in a price list allow you to create multiple line items in a price list for a product, with different effective dates and different prices.
- Every component adjustment can have an effective start and end date.

For more information, see *Setting Up Pricing for Products with Components*.

Volume Discount Enhancements

Volume discounts provide multiple types of price adjustments, such as percentage markup and markup amount. For more information, see *Creating Volume Discounts*.

Enhancements Across the Application

In addition to the features used to set up pricing, there are also many enhancements to pricing behavior in quotes and orders. For example, there is support for multiple price types and multiple currencies. For more information, see *Pricing in the Run-Time Application*.

About the Processing Order of Price Adjustments

The order in which price adjustments are processed can affect the final price. For example, a product costs $100 and you give a $10 promotional discount plus a 10 percent volume discount if the customer buys more than ten. If the customer qualifies for both discounts, then you can calculate the final price in two ways:

- The unit cost would be $81 if the promotional discount is applied first ($100 - $10 = $90, and 10 percent off $90 is $81).
- The unit cost would be $80 if the volume discount is applied first (10 percent off $100 is $90, and $90 - $10 = $80).

The pricing procedure determines the processing order of discounts, but you can customize it. For information about the default processing order, see *Pricing Procedures and Workflow References*.

Clearing Pricing Data from the Cache

After price list data is read from the database, it is cached in memory to improve performance. If a price list is modified, then the cached pricing data may no longer be valid.

When modified or newly developed pricing data is added to the database, the user clicks the Clear Cache button to remove cached data that may be obsolete.

The Clear Cache button is available in the Price Lists view, Volume Discount view, and many other views. In any view, it clears only the selected price list, volume discount, or other pricing data from the cache. Other pricing data remains in the cache to improve performance. For information about setting up caching, see *Siebel Order Management Infrastructure Guide* .

To clear pricing data from the cache

1. Navigate to the Administration - Pricing screen, then Price Lists view, Volume Discount view, or other appropriate view.
2. Select one or more Price List records, Volume Discount records, or other pricing records whose cached data may be obsolete.
3. Click Clear Cache.

Which Type of Promotion Pricing to Use

Siebel Pricer provides two types of promotions:

- **Aggregate discount promotions.** These promotions allow you to give discounts to customers who buy a combination of products, as described in *Creating Aggregate Discounts*. You use this type of promotion to maintain promotions that were defined using bundle pricing factors in earlier versions of Siebel Pricer.
- **Product promotions.** These promotions allow you to create many types of promotions, including bundle promotions, as described in *Creating Product Promotions*. This type of promotion has features such as the

ability to display promotions in product catalogs and quote line items. It is recommended that you use product promotions for most new promotions.

3 Creating and Assigning Price Lists

Creating and Assigning Price Lists

This chapter begins with an overview of price lists, and then describes the most common way of creating a new price list. It also instructs you on assigning price lists to users. This chapter covers the following topics:

- *About Price Lists*
- *About Sales Products and Service Products*
- *Price List Prerequisites*
- *Process of Creating a Price List*
- *About Setting Up Service Pricing*
- *About Effective Dates for Pricing Line Items*
- *Giving a Product Multiple Prices with Different Effective Dates*
- *Creating a Price List Line Item for a Simple Product Bundle*
- *Copying and Modifying a Price List*
- *Copying and Transforming a Price List*
- *Importing Price Lists*
- *Assigning Price Lists to Users*
- *Enabling Price List-Specific and Line Item-Specific Pricing Logic*

About Price Lists

A *price list* is a set of standard prices for products or services.

A product must exist in a price list so that it appears in a catalog. So, the price list is one of the requirements for a product's visibility.

A product price defined in a price list is commonly used as the starting price for other prices generated by Siebel Pricer. Test the price list thoroughly before applying more advanced price adjustments.

This chapter describes the four common ways of creating a price list:

- Create a new price list
- Copy and modify an existing price list
- Copy and transform an existing price list
- Import price lists from other applications using Siebel Enterprise Integration Manager (EIM)

After you create a price list, you must assign it to customers so that it controls the prices available to customers.

A product can appear in many different price lists. For example, you can create:

- One price list with the wholesale prices for all of your products.

- One price list with the retail prices for all of your products.

The line item records of these two price lists will have different prices for the same products. You assign these price lists to the appropriate customers.

About Sales Products and Service Products

Price lists can include prices for two different types of products:

- **Sales products.** This is any product that is sold to a customer.
- **Service products.** This is a specialized type of sales product consisting of a service that covers another sales product, such as a warranty service.

Price List Prerequisites

Before you create any price list, you must create the products to which the prices will apply. You must define products. For more information, see *Siebel Product Administration Guide* .

Optional Prerequisites

The price list includes optional fields that let you associate it with other data in the Siebel application. You must define the other data before you can use these optional fields.

For example, if you want a price list to use volume discounts, then you must define volume discount information before you can associate it with line items in a price list.

You do not have to define this data before creating the price list. You can skip these optional fields when you create the price list, and go back to the price list to fill them in later, after defining the data that is needed.

The types of data that can optionally be associated with price lists include:

- **Pricing procedure.** You can create a custom pricing procedure with pricing logic specific to a price list or to line items. For more information, see *Enabling Price List-Specific and Line Item-Specific Pricing Logic*.
- **Rate lists.** For more information, see *Creating and Assigning Rate Lists*.
- **Cost lists.** For more information, see *Creating and Using Cost Lists*.
- **Volume discounts.** For more information, see *Creating Volume Discounts*.
- **Aggregate discount sequences.** For more information, see *Creating Aggregate Discounts*.
- **Attribute adjustments.** For more information, see *Creating Attribute Adjustments*.
- **Component-based pricing adjustments.** For more information, see *Setting Up Pricing for Products with Components*.

Process of Creating a Price List

A price list consists of one price list header record that is associated with multiple line item records. The header record contains general information about the price list as a whole. The line item records contain prices for specific products.

There are two types of line items:

- **Sales products.** Enter line items for sales products in the Price List Line Items view.
- **Service products.** Enter line items for service products in the Service Pricing view.

A single price list can contain one or both of these types of line items. For more information about sales and service products, see *About Sales Products and Service Products*.

To create a new price list, perform the following tasks:

1. *Defining a Price List Header*.
2. *Adding Price List Line Items for Products*.
3. *Adding Price List Line Items for Service Products*.

> **Note:** Creating a price list line item for a product with components is more complex than creating a line item for a simple product. Consult with a Siebel Administrator before creating price lists for products with components. For more information about products with components, see *Setting Up Pricing for Products with Components*.

Defining a Price List Header

This task is a step in *Process of Creating a Price List*.

Each price list is defined by the data in its header, which includes its name, describes its purpose, and specifies the time period when it will be effective.

The price list header does not hold the actual prices for products. Prices are in the associated price list line items.

To define a price list header

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, add a new record and complete the necessary fields, as described in the following table.

Field	Comments
Name	Enter a meaningful name for this price list. Overwrite any Siebel-Pricer-generated name. If many price lists appear in your Siebel application, then use a consistent naming convention to name all your price lists.
Cost List	Select the cost list you want to associate with this price list.
Currency	Enter the default currency for this price list.

Field	Comments
Shipping Method	Select one default shipping method you want to associate with this price list. This appears as the default Shipping Method when a run-time user creates a quote or order that specifies this price list.
Payment Terms	Select one default set of payment terms to be associated with this price list. This selection appears as the default data for Payment Terms when a run-time user creates a Quote or Order that specifies this price list.
Organization	Select all the organizations that may have prices controlled by this price list. For more information, see *Assigning Price Lists to Users*.
Ship Method	Enter the mode used for shipping such as air, ground, and so on.
Shipping Terms	Select one default shipping charge protocol that you want to associate with this price list. This appears as the default Shipping Terms data when a run-time user creates a quote or order that specifies this price list.
Effective From	Enter the date and time when this price list will become effective. By default, Siebel Pricer assigns the current system date and time when you first create the Price List record.
Effective To	Enter the date and time when this price list will become ineffective. After this time, Siebel applications will not be able to use this price list.
Updated By	By default, Siebel Pricer assigns the user name used to log in to the current session in which this Price List record is created.
Last Updated	By default, Siebel Pricer assigns the current system date and time when you most recently saved this Price List record.
Integration ID	If this Siebel-Pricer-generated ID appears, then you can use it as a unique identifier for this price list to assist with integration tasks. The Integration ID field is used for integration with external applications. This field is populated by Siebel EAI connectors.
Pricing Procedure	Select a custom pricing procedure. The pricing logic in this procedure will be applied to all the line items in the price list. Before this field is available, you must do the setup work described in *Enabling Price List-Specific and Line Item-Specific Pricing Logic*.

Field	Comments
Aggregate Discount Sequence	Select an aggregate discount sequence in order to apply aggregate discounts to the price list. For more information, see *Creating Aggregate Discounts*.

Adding Price List Line Items for Products

This task is a step in *Process of Creating a Price List*.

A price list line item for a product includes the product name, the product price, and related information. Adding a line item is equivalent to adding a product to the price list.

A line item for a product can appear in multiple price lists. For example, your United States price list might have a line item for the product's retail price in the United States, your Canadian price list may have the product's retail price in Canada, and so on.

Note: Creating a price list line item for a product with components is more complex than creating a line item for a simple product, and it requires additional steps that are not included in this procedure. For more information about products with components, see *Setting Up Pricing for Products with Components*.

To add a new price list line item for a product

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price List form, select the price list to which you want to add a line item.
3. Click the Price List Line Items view tab.
4. In the Price List Line Items list, click New.

   The Add Products dialog box appears.
5. Use the Add Products dialog box to add the product to the price list.
6. If the product is a product with components, then see the instructions in *Setting Up Pricing for Products with Components*.
7. Complete the fields in the Price List Item record, and the Price List Line Item Detail form, as described in the following table.

   CAUTION: If you do not specify a price for a product in its price list line item, then that product may appear in a quote or order without a price, effectively making it free to a run-time user who is buying it. If you depend on other pricing mechanisms to supply a missing price, then be sure to test the results in the run-time environment.

Field	Comments
Product	Select a product name using the Add Products dialog box.

Field	Comments
Price Type	This field is read-only and displays the price type for this product. Options are One-Time, Recurring, and Usage. The value is based on the product definition. For more information about multiple price types, see *Multiple Price Types and Totals*.
Customizable	When this check box is selected, it means that the product has components. This value is based on the product definition and is read-only.
List Price	Enter a list price, the standard price used for most transactions and the most commonly used target price for price adjustments. If you enter a promotional price, then it is used instead of this list price. The list price is not checked against the Minimum Price and Maximum Price fields when it is first entered, because these fields are usually entered after the list price. When the minimum and maximum prices are entered, the list price is validated against them. **Note:** If the list price, promotional price, and price override are all omitted, then the product may be offered with no price, the equivalent of a zero price.
Promotional Price	If you enter a promotional price, then it will be used instead of the list price in all functions that do not explicitly identify the list price as the target price.
Volume Discount	Select the volume discount that applies to this line item, if any. More than one volume discount may be available for a given product, and all the volume discounts that can apply to this product are displayed in the drop-down list. For more information, see *Creating Volume Discounts*
Attribute Adjustments	This field specifies the attribute adjustments record that provides attribute-based pricing adjustments for this line item, if any. For more information, see *Creating Attribute Adjustments*.
Start Date and End Date	If this line item has different effective dates from the price list, then enter the start date when the pricing becomes effective and the end date when it is no longer effective. To enter values in these fields, use the method described in *Giving a Product Multiple Prices with Different Effective Dates*.
Product Line	This field displays the product line to which this line item product belongs. This value is taken from the product information.
Part Number (Part #)	This field displays the part number assigned to the line item product. This value is derived from the product information.

Field	Comments
Cost	Enter the cost of the line item product. You can use this as the target price for some price adjustment calculations.
MSRP	Enter the manufacturer's suggested retail price (MSRP) for reference purposes. In some cases, you can use this as the target price for price adjustment calculations.
Purchase Price	Enter the purchase price of the line item product. You can use this as the target price for some price adjustment calculations.
Unit of Measure	This field displays the unit of measure for the product item. This value comes from the product definition.
Minimum Price	If a minimum price value is provided, then Siebel Pricer does not offer a lower price for this line item, unless a run-time user (such as an agent) manually overrides this price. As a default, specify a minimum price of zero to prevent amount discounts from resulting in negative number prices. The minimum price does not apply to components of products with components.
Maximum Price	If a maximum price value is provided, then Siebel Pricer does not offer a higher price for this line item, unless a run-time user (such as an agent) manually overrides this price. If a maximum price is provided for a price list line item product and if that product appears as a component in a product with components, then this maximum price will apply to that component product as well.
% Margin	Read-only. This is a calculated field that calculates the margin based on the following formula: ((List or promotional price) – cost)/List*100
Pricing Procedure	Select a custom pricing procedure. The pricing logic in this procedure will be applied to this line item. Before this field is available, you must do the setup work described in *Enabling Price List-Specific and Line Item-Specific Pricing Logic*.

Adding Price List Line Items for Service Products

This task is a step in *Process of Creating a Price List*.

A price list line item for a specific service product includes the product name, the product price, and related information, such as the time when the service price is effective. Adding a record to the Service Pricing view is equivalent to adding a service product to the price list.

As a shortcut, if you are adding the same service product with different time intervals, then you can click the New Effective Date button to add another record for the same service product, so you have to enter only the new start date, end date, and price. The time interval is the period when the service is delivered.

You can also base service pricing on the covered asset, as described in *About Setting Up Service Pricing*.

To add a new price list line item for a service product

1. Navigate to the Administration - Pricing screen, then Price Lists view.

   The Price Lists view appears, displaying the currently defined price lists.
2. In the Price List form, select the price list to which you want to add a line item.
3. Click the Service Pricing view tab.
4. In the Service Pricing list, click New.

   The Add Products dialog box appears.
5. Use the Add Products dialog box to add the product to the price list.
6. If the product is a product with components, then see the instructions in *Setting Up Pricing for Products with Components*.
7. Complete the fields in the Service Pricing record and the Service Pricing Details form, as described in the following table.

Field	Comments
Product	Select a product name using the Add Products dialog box.
Customizable	When this check box is selected, it means that the product has components. This value is based on the product definition and is read-only in this screen.
Service Product	This check box is selected if this is a service product. This information depends on the product definition.
List Price	Enter a list price, the standard price used for most transactions and the most commonly used target price for price adjustments. **Note:** If the list price and price override are both omitted, then the product may be offered with no price, the equivalent of a zero price.
Start Date and End Date	If this line item has different effective dates from the price list, then enter the start date when the pricing becomes effective, and the end date when it is no longer effective. To enter values in these fields, use the method described in *Giving a Product Multiple Prices with Different Effective Dates*.

Field	Comments
Service Price Method	Specify the basis used for calculating the price markup of the service product from the covered product. The options are the list price and net price of the covered product. The percentage entered in the Service Price % field is applied to the price you specify in this field, and the resulting amount is added to the list price of the service to calculate the price.
Service Price %	Enter a percentage that is used to mark up the price of the service product. The price for this service product is the list price of the service product plus this percentage of the net or list price of the covered product, depending on whether you selected net price or list price in the Service Price Method field.

About Setting Up Service Pricing

You can set up service pricing in two ways:

- Price the service as a percentage of the covered product. For example, a one-year warranty can be priced at 10% of the price of the computer system that it covers. For information about how to do this, see *Setting Up Service Pricing as a Percentage of a Covered Product*.
- Price the service with adjustments based on a primary covered product. The price of the service can be equal to the list price of the service with a price adjustment based on the product that is covered. For information about how to do this, see *Setting Up Service Pricing with Adjustments Based on a Covered Product*.

In either case, the service pricing is triggered when the end user creates a quote or order and associates the service to the covered product.

Note: You must enable the Dynamic Pricing Procedure to use service pricing. For information about enabling the Dynamic Pricing Procedure, see *Switching from Basic to Dynamic Pricing Procedure*.

Setting Up Service Pricing as a Percentage of a Covered Product

The price of the service is equal to the list price of the service plus a percentage of the price of the covered product.

This method of pricing is available without configuration.

To set up service pricing as a percentage of a covered product, perform the following tasks:

- *Pricing a Service as a Percentage of a Covered Product*
- *Associating a Service with a Covered Product*

Pricing a Service as a Percentage of a Covered Product

A service can be priced as a percentage of the covered product. Alternatively, there can be a list price for the service, and a percentage of the covered product is added to this list price.

To set up service pricing as a percentage of the covered product

1. Add the service price line item, as described in *Adding Price List Line Items for Service Products*.
2. Complete the fields in the Service Pricing Details form as described in the following table.

Field	Comments
Service Pricing Method	Select Net Price of List Price to determine whether the service price is a percentage of the net of list price of the covered product.
Service Price %	Enter the percentage of the covered product's price to use as the service price.
List Price	Enter a list price for the service product.

Associating a Service with a Covered Product

When the end user creates a quote or order for a service whose price is based on a product, the end user must associate the service with the product. For more information about how an end user creates a quote or order, see *Siebel Order Management Guide* .

To associate a service with a covered product

1. Create a quote or order as described in *Siebel Order Management Guide* .
2. Add the covered product as a line item.
3. From the Line Items applet menu, choose Service.

   A dialog box appears that includes only service products.
4. Select the appropriate service product from the dialog box, and click OK.

   The service product is added as a line item to the quote or order, and it is associated with the covered product. This association appears in the Covered Product field of the service product line item.

   > **Note:** As an alternative to choosing Service from the menu, you can directly populate the Covered Product field of the service line item to create an association between the service and the covered product.

Setting Up Service Pricing with Adjustments Based on a Covered Product

The price of the service equals the list price of the service plus a price adjustment that depends on the primary product that it covers.

> **Note:** This method of pricing requires configuration of the pricing procedure. You must enable the Dynamic Pricing Procedure to use service pricing. For information about enabling the Dynamic Pricing Procedure, see *Switching from Basic to Dynamic Pricing Procedure*.

To set up service pricing with adjustments based on a product, perform the following tasks:

- *Pricing a Service with Adjustments Based On a Primary Covered Product*
- *Associating the Service Product with Covered Products*

Pricing a Service with Adjustments Based On a Primary Covered Product

Pricing administrators can set up service pricing with an adjustment based on the primary product that is covered. For example, there can be a list price for a one-year service agreement, with a 10 percent discount if the agreement covers computer system A and a 15 percent discount if it covers computer system B.

To price a service with adjustments based on a product

1. Add the service price line item, as described in *Adding Price List Line Items for Service Products*.
2. Add a record to the Service Pricing Details list for each product that the service covers and complete the necessary fields, as described in the following table.

Field	Comments
Product	Select the product to which this discount applies.
Adjustment Type	Select the type of the adjustment that the discount will apply. Options are Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.
Target Price	Select the price to which the discount applies.
Adjustment Value	The value in the Adjustment Type field determines how the adjustment value is applied, as follows: o If the Adjustment Type is Discount Amount, then the amount in the Adjustment Value field is subtracted from the price. o If the Adjustment Type is % Discount, then the amount in the Adjustment Value field is treated as a percentage and is subtracted from the price. For example, if the Adjustment Amount is 10, then 10 percent is subtracted from the price. o If the Adjustment Type is Markup Amount, then the amount in the Adjustment Value field is added to the price. o If the Adjustment Type is % Markup, then the amount in the Adjustment Value field is treated as a percentage and added to the price. o If the Adjustment Type is Price Override, then the amount in the Adjustment Amount field is used instead of the price.

Associating the Service Product with Covered Products

When the end user creates a quote or order for products that are covered by this service, the end user must associate the service with one or more products. The price adjustment for the service depends on the primary product. For more information about how an end user creates a quote or order, see *Siebel Order Management Guide* .

To associate a service with covered products

1. Create a quote or order as described in *Siebel Order Management Guide* .
2. Add the service product as a line item of the quote or order.

3. Click the Assets view tab.
4. In the Assets list, add records and in the Asset # field, select the products covered by this service.
5. In the Asset record of the primary product covered by this service, select the Primary check box.

   The price adjustment for the service depends on the primary product.

About Effective Dates for Pricing Line Items

You can create effective dates for individual line items in a price list, cost list, discount matrix, or component-product price adjustment. The application prevents you from entering overlapping effective dates for the same line item when you enter effective dates using the New Effective Date button, as described in *Giving a Product Multiple Prices with Different Effective Dates*.

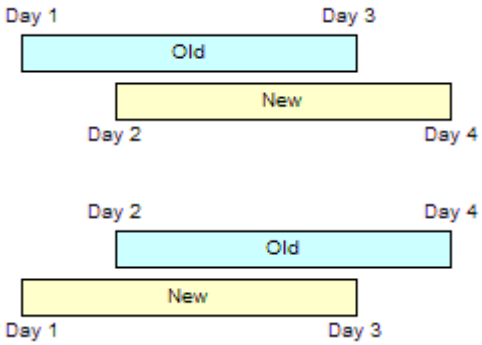
For example, if a price list has a line item for Product X with a price that is effective from January 2005 through June 2005, then do not enter another line item for Product X with a price that is effective from March, 2005 through December 2005, because the product would have two different prices during several months.

This topic describes several scenarios that show how the application resolves overlapping dates.

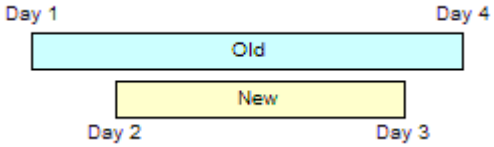
The New Time Interval Overlaps with One Existing Time Interval

This topic describes several possible scenarios where a new time interval overlaps with an existing time interval.

If the effective dates in the new record overlap with the existing record's start date, or with the existing record's end date, as shown in the following figures, then the application displays an error message and does not enter the new effective dates.

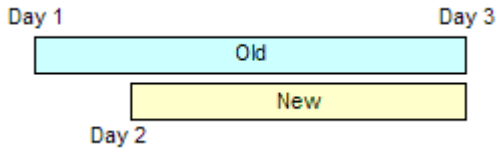


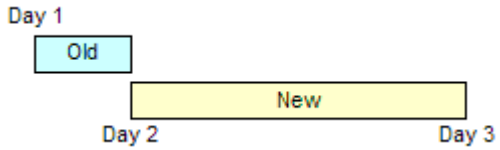
However, if the new time interval is entirely within an existing time interval and does not overlap with the start or end date, as shown in the following figure, then the application creates three time intervals, as shown in the fourth figure in this topic.



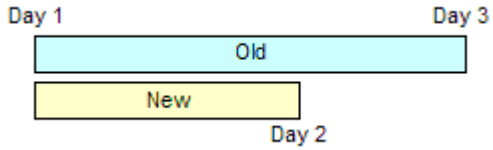


Likewise, if the new time interval is entirely within an existing time interval but its end date is the same as the end date of the existing interval, as shown in the following figure, then the application creates two time intervals, as shown in the sixth figure in this topic.

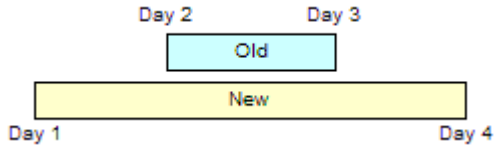




However, if the new time interval is entirely within an existing time interval but its start date is the same as the start date of the existing interval, as shown in the following figure, then the application displays an error message and does not create the new time interval.

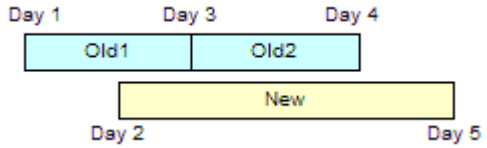


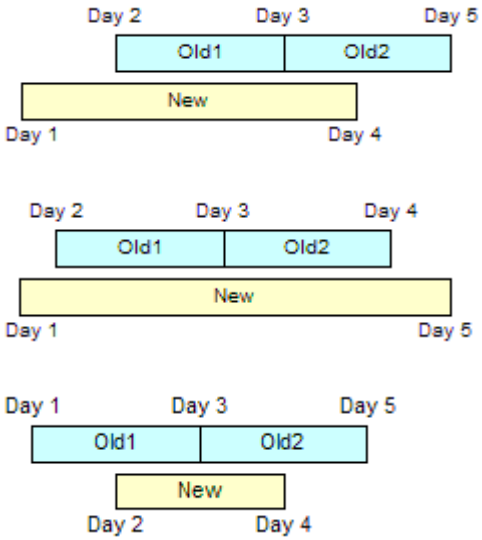
However, if the new time interval extends beyond the existing time interval, as shown in the following figure, then the application displays an error message and does not create the new time interval.



The New Time Interval Overlaps with Two Existing Time Interval

This topic describes the possible scenarios where a new time interval overlaps with two existing time intervals. The effective dates in the new record may overlap with the effective dates in two existing records, as shown in the following figures. In all of these cases, the application displays an error message and does not enter the new effective dates.





Giving a Product Multiple Prices with Different Effective Dates

Price lists, cost lists, discount matrices, and component-product price adjustments have effective dates for the list or group as a whole. You can also enter effective dates for individual line items.

Effective dates for individual line items allow you to change prices rapidly and track changes to a product's prices. For example, if you have different prices for certain products during different seasons, then you can include prices for all the seasons in one price list, so prices change automatically when the season changes.

As a best practice, do not delete line items from a price list or directly modify the value of prices at a line level. In order to make price changes, create multiple line item records with different effective dates, as described in the following procedure.

This topic describes how to create effective dates for line items in price lists. Use the same methods for cost lists, discount matrices, or component-product price adjustments.

When you use the method described here to add multiple line items for the same product with different effective dates, the application does not create line items with overlapping effective dates. For information about how the application resolves this issue, see *About Effective Dates for Pricing Line Items*.

To create multiple prices with different effective dates

1. Create one pricing line item for the product as described in *Adding Price List Line Items for Products*.
2. Select the pricing line item for the product, and then click New Effective Date.

   A new line item is created for the same product.
3. In the new line item, enter the Start Date and End Date for the new time interval.
4. If necessary, repeat Step 2 and Step 3 to continue adding more line items with different effective dates for the product.

Creating a Price List Line Item for a Simple Product Bundle

A product bundle allows you to give customers a discount if they buy a combination of products. For example, you can give customers a price for a dining room table and four chairs that is less than the price they would pay if they bought the table and chairs individually.

You do this by creating a product of type bundle. You assign a price to the bundle product in the same way that you assign a price to an individual product. For information about creating simple bundle products, see *Siebel Product Administration Guide* . Before performing the following procedure, follow the instructions in *Adding Price List Line Items for Products*.

To create a price list line item for a simple product bundle

1. Create a new product line item, as described in *Adding Price List Line Items for Products*.
2. In the Name field, select a product bundle that has been defined in the Product Administration screen.

Copying and Modifying a Price List

After building a price list, you can copy and modify it to create a new price list for different types of customers or markets. For example, you might use a price list named United States Price List as the basis for creating new price lists for different countries, locations, organizations, currencies, accounts, or demographics.

To modify the data in more elaborate ways, you use the Transform functionality, described in *Copying and Transforming a Price List*, instead of modifying the data manually.

Note: Do not edit the Currency field when you copy and modify a price list. To modify this field, use the Transform function, described in *Copying and Transforming a Price List*.

To copy and modify a price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list you want to copy.
3. In the Price Lists list, click the menu button, and then click Copy Record.

   A new Price List record appears with all of the data from the original price list, except for the original price list name.
4. Modify the information in the fields of the price list header and the price list line items.

   Either edit the fields directly or click the menu button, click Change Records, and then use the Change Records dialog box to modify the record.

Copying and Transforming a Price List

The Transform function allows you to transform a copy of an existing price list by making sweeping price changes to product prices. It lets you change every list price without modifying each individual line item separately. Using the Transform options, you can:

- Change all list prices by the percentage you specify. For example, you can increase all list prices to 130 percent of their previous value.
- Convert all list prices to a different currency. You specify the currency and the conversion date, and the prices are converted automatically.
- Apply a pricing procedure to a price list, changing some or all product prices depending on the design of the procedure. To do this, you must create the pricing procedure before you do the transformation. For more information about creating pricing procedures, see *Siebel Order Management Infrastructure Guide* .

The Transform function transforms the list price as well as the promotional price fields.

To transform a price list

1. Select the price list you want to use as a basis for the new price list.
2. From the Price List menu, select Transform.

   The Transform Price List form appears.
3. Enter information in the Transform Price List form, as described in the following table, and then click Transform.

Field	Comments
Source Price List Name	Enter the name of the price list to be transformed.
Destination Price List	Enter the name of the price list resulting from transforming the Source Price List.
Currency Code	Select the currency you want to use for prices in the transformed price list. The default value is the currency of the copied price list. If you are not changing currencies, then keep the default.
Exchange Date	If you are converting currencies, then enter the date for the currency exchange rate. The Siebel application will use the exchange rate on that day to convert currency.
Markup or Discount %	Enter a percentage to be applied to the list price for each product in the price list. For example, to increase all list prices by 30 percent, you would enter 130. To decrease all list prices by 20 percent, you would enter 80. The default value is 100 percent. If you do not want to change prices, then keep this default.
Pricing Procedure	If you are using a pricing procedure to transform the prices, then select the Workflow that calls the pricing procedure.

Field	Comments

Note: After clicking the Transform button to change the data, be sure to validate your results.

Importing Price Lists

It is common to use Siebel Enterprise Integration Manager (EIM) to import price lists into the Siebel application to handle large numbers of price lists that were created in other applications.

Price lists are defined by organization. By default, the column ENTERPRISE_FLG of a manually entered price list is set to N, which enables organizational visibility of price lists and associated price list items.

If you use Siebel EIM to import price lists and price list items, then set ENTERPRISE_FLG to N to retain organizational visibility. If ENTERPRISE_FLG is set to Y, then the price lists and associated price list items are become visible across the enterprise, and are routed to all users regardless of organization. For more information about using Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide* .

Assigning Price Lists to Users

The price list that controls a user's prices depends on both organization and account:

- **Organization.** This option limits visibility. The user cannot see a price list unless it was assigned to the user's organization in the Organization field of the price list record. One price list can be assigned to many organizations, and many price lists can be assigned to a single organization.
- **Account**. This option controls which price list is active for the user. The active price list is the price list associated with the user's account record.

When you associate a price list with an account, that price list is automatically defaulted in the quote or order when the account is entered for the quote or order, and the price list's currency is set as the header-level currency for the quote or order.

Note: If you use Siebel Sales to sell to the public through the Web, then customers may not have accounts. The default price list is for customers without accounts. For information about setting a default price list, see the topic about setup tasks in *Siebel eSales Administration Guide* .

To assign a price list to a user

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list you want to assign to the user.
3. In the price list's Organization field, click the select icon.

   The Organization's dialog box appears.

4. If the user's organization is not already listed in this dialog box, then click New, add it, and then click OK.
5. Navigate to the Accounts screen, then Accounts List view.
6. In the Accounts list, click the name of the user's account.
7. Click the More Info view tab.
8. In the More Info view, go to the Price List field list and select the price list for this account.

Enabling Price List-Specific and Line Item-Specific Pricing Logic

You can associate a custom pricing procedure with either an entire price list or an individual line item in a price list. To associate a custom pricing procedure with an entire price list, select it in the Pricing Procedure field of the Price List record, as described in *Defining a Price List Header*.

To associate a custom pricing procedure with a line item, select it in the Pricing Procedure field of the Line Item record, as described in *Adding Price List Line Items for Products*.

Before you select the custom pricing procedure in these fields, you must create the custom pricing procedure and integrate it with the default pricing procedure. For more information about creating pricing procedures, see *Siebel Order Management Infrastructure Guide* .

To create price list-specific or line item-specific pricing logic, follow this general process:

- Create a custom pricing subprocedure with the pricing logic that you want to add to the price list or line item.
- Add a dynamic subprocedure step to the default pricing procedure.
- Modify the default pricing procedure to retrieve this subprocedure during Price List Lookup.
- Associate the custom subprocedure with the price list or line item as follows:
    - To associate it with the price list, select it in the Pricing Procedure field of the Price List record.
    - To associate it with a line item, select it in the Pricing Procedure field of the Line Item record.

4 Creating and Assigning Rate Lists

Creating and Assigning Rate Lists

This chapter begins with an overview of rate lists, and then describes the most common ways of creating a new rate list. It also instructs you on assigning rate lists to users. This chapter covers the following topics:

- *About Rate Lists*
- *Process of Creating Rate Lists*
- *Copying and Modifying a Rate List*
- *Copying and Transforming a Rate List*
- *Assigning a Rate List to a User*

About Rate Lists

A rate list is a set of standard rates for resources that bill by the hour. For example, if you run a temporary employment agency, then you might charge your customers different standard hourly rates for employees who do word processing, who do data entry, who do secretarial work, and who answer the phone at a call center. These hourly rates for these different types of resources would be included on your rate list.

After you have set up rate lists administratively, they are used by service orders and time sheets to calculate the cost of resources.

If you assign a product price in a rate list, then it can appear in catalogs. Rate lists control the visibility of products, just as price lists do.

This chapter covers the general process of creating rate lists. More detailed processes for using rate lists with specific features of Siebel applications are covered in other books. Common uses of rate lists include:

- **Project Management for Professional Services.** Rate lists let you manage projects that use professional services to determine the billing rates for professional services for that project. For more information, see *Siebel Project and Resource Management Administration Guide* .
- **Field Service.** Rate lists determine the rates that customers pay for the time of field service representatives. For more information, see *Siebel Field Service Guide* .
- **Pricing Screen.** After the pricing administrator sets up rate lists using the Administration - Pricing screen, employees can view rate lists using the Pricing screen.

This chapter looks at three common ways of creating a rate list:

- Create a new rate list
- Copy and modify an existing rate list
- Copy and transform an existing rate list

After you create a rate list, you must assign it to users to make it control the rates they pay. For example, if you run a temporary help agency, then you may charge different rates to customers who need help for short-term projects and

long-term projects, or you may charge different rates to larger and smaller customers. You would create and assign different rate lists to these customers to make their rates visible to them.

Process of Creating Rate Lists

A rate list consists of one rate list record and multiple rate list line item records. The rate list record contains general information about the rate list as a whole. The line item records contain rates for specific resources.

To create a new rate list, you go through the following process:

1. *Creating Resources as Products*
2. *Creating a Rate List Record*
3. *Creating Rate List Line Items*

Creating Resources as Products

This task is a step in *Process of Creating Rate Lists*.

Before you create any rate list, you must create the resources that the rates apply to as products in the Product Administration screen. For example, if you run a temporary help agency, then you might create products named word processing operator, data entry clerk, secretary, and so on.

You must:

- Create the products.
- Associate each product with its product lines.

> **Note:** To use these products as resources in the rate list, you must select the Project Resource check box when you create the product in the Product Administration screen.

For more information about creating products, see *Siebel Product Administration Guide* .

Creating a Rate List Record

This task is a step in *Process of Creating Rate Lists*.

The Rate List record includes general information about the rate list as a whole, such as its name, description, and effective time period.

To create a rate list record

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate Lists list, add a new record and complete the necessary fields, as described in the following table.

Field	Comments
Name	Enter a unique, meaningful name for this rate list.

Field	Comments
Description	Enter a description of the rate list for your own use.
Cost List	Select the cost list you want to associate with this rate list.
Currency	Select the currency for this rate list.
Effective From	Enter the date and time when this rate list will become effective. By default, the application assigns the current system date and time when you first create the Rate List record.
Effective To	Enter the date and time when this rate list will become ineffective. After this time, Siebel applications will not use this rate list. If no value is entered here, then the rate list remains in effect indefinitely.
Updated By	By default, the application assigns the user name used to log in to the current session in which this Rate List record is created.
Last Updated	By default, the application assigns the current system date and time when you most recently saved this Rate List record.
Organization	Select all the organization that may have rates controlled by this rate list. For more information, see *Assigning a Rate List to a User*.

Creating Rate List Line Items

This task is a step in *Process of Creating Rate Lists*.

A rate list line item contains rate data for a specific resource. It includes the resource name and information about the rates charged for that resource.

If you have multiple rate lists, then the same resource can have a line item in each rate list. For example, your United States rate list might have a line item for the rate charged for a resource in the United States. The Canadian rate list may have the rate charged for the resource in Canada, and so on.

Alternatively, you can add a resource to multiple rate lists using the Administration - Product screen, then Rate List view.

> **Note:** Because rate list line items do not have fields for their start and end dates, you cannot have more than one line item for a given resource.

To create a rate list line item

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate List, select the rate list for which you want to add a line item.
3. Click the Rate List Line Items view tab.
4. In the Rate List Line Items list, click New.

   The Add Position Types dialog box appears.
5. Use the Add Position Types dialog box to add the type of resource to the rate list.
6. Complete the fields of the Rate List Item record, as described in the following table.

Field	Comments
Resource	This field displays the type of resource that you selected from the Add Position Types dialog box. A rate list line item uses this product name as its unique ID.
Rate Per Hour	Enter the standard hourly rate you charge for this resource.
Overtime Rate Per Hour	Enter the hourly overtime rate you charge for this resource.
Extended Overtime Rate Per Hour	Enter the hourly extended overtime rate you charge for this employee, if applicable.
Material Mark-Up	Enter the percentage of markup charged for materials used by this resource.
Expense Mark-Up	Enter the percentage mark-up charged for expenses incurred by this resource.
Description	Enter a description of this resource for your own use.

Copying and Modifying a Rate List

After creating a rate list, you can copy and modify it to create a new rate list for different types of customers or markets. For example, you might use a rate list named United States Rate List as the starting basis for creating new rate lists for different countries.

First, copy the rate list. This retains the original rate list and creates a new rate list with the same data. Then, modify the data in the copy as necessary.

To modify the data in more elaborate ways, use the transform feature described in *Copying and Transforming a Rate List*, instead of modifying the data manually. For example, you can use the transform feature to convert all the currencies in a rate list to a different currency.

Note: Do not edit the Currency field when you copy and modify a rate list. To modify this field, use the Transform function described in *Copying and Transforming a Rate List*.

To copy and modify a rate list

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate Lists list, select the rate list you want to copy.
3. In the Rate Lists list, click the menu button, and then click Copy Record.

   A new Rate List record appears, with all of the data from the original rate list, except for the original rate list name.
4. Modify the information in the fields of the rate list record and the rate list line items.

   Either edit the fields directly or click the menu button, and then click Change Records, using the Change Records dialog box to modify the record.

   Note: Do not edit the Currency field directly. To modify this field, use the Transform function, described in the *Copying and Transforming a Rate List*.

Copying and Transforming a Rate List

The Transform function allows you to transform a copy of an existing rate list by making sweeping changes to rates. It lets you change every rate without modifying each individual line item separately. Using the Transform options, you can:

- Change all list prices by the percentage you specify. For example, you can increase all list prices to 130 percent of their previous value.
- Convert all list prices to a different currency. You specify the currency and the conversion date, and the prices are converted automatically.
- Apply a pricing procedure to a rate list, changing some or all rates depending on the design of the procedure. To do this, you must create the pricing procedure and the workflow that calls it before you do the following procedure. For more information, see *Siebel Order Management Infrastructure Guide* .

To transform a rate list

1. Select the rate list you want to transform.
2. From the Rate Lists list menu, select Transform.

   The Transform Rate List form appears.
3. Enter information in the Transform Rate List form, as described in the following table, and then click Transform.

Field	Comments
Transformed Rate List Name	Enter the name of the rate list to be transformed.
Destination Rate List	Enter the name of the rate list resulting from the transformation.

Field	Comments
Currency Code	Select the currency you want to use for prices in the transformed rate list. The default value is the currency of the copied rate list. If you are not changing currencies, then keep the default.
Exchange Date	Enter the date for the currency exchange rate. The Siebel application will use the exchange rate on that day to convert currency. You must enter the date if you are converting currencies.
Markup or Discount %	Enter a percentage to be applied to the list price for each product in the rate list. For example, to increase all list prices by 30 percent, you would enter 130. To decrease all list prices by 20 percent, you would enter 80. The default value is 100 percent. If you do not want to change prices, then keep this default.
Pricing Procedure	If you are using a pricing procedure to transform the prices, then select the workflow that calls the pricing procedure.

Assigning a Rate List to a User

The rate list that an employee sees depends on that employee's organization. The employee cannot see a rate list unless it was assigned to the user's organization in the Organization field of the rate list record.

One rate list can be assigned to many organizations. Each organization can have multiple rate lists.

You can configure your application using Web Tools to make multiple rate lists available to one organization. For more information about Web Tools, see *Using Siebel Tools* .

To assign a rate list to a user

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate Lists list, select the rate list you want to assign to the user.
3. In the rate list's Organization field, click the select icon.

   The Organizations dialog box appears.
4. If the user's organization is not already listed in this dialog box, then click New, add it, and then click OK.

5 Creating and Using Cost Lists

Creating and Using Cost Lists

This chapter describes how to create and use cost lists, and covers the following topics:

- *About Cost Lists*
- *Process of Creating Cost Lists*
- *Using Asset Mapping to Value Assets*

About Cost Lists

A cost list is a set of costs for products or services. You can attach a cost list to either a rate list or a price list to determine the profit margin.

There are four cost methods that you can use to determine the cost of a product or service:

- **Standard Cost.** This is a predetermined operating cost that is compared with the actual cost to measure the performance of a given department or operation.
- **Last Cost.** This is a cost metric used in the LIFO (Last-In, First Out) costing method. LIFO calculates cost by assuming the last goods purchased are the first goods sold, so the ending inventory consists of the earliest goods purchased.
- **Next Cost.** This is a costing method that allows the user to maintain the cost manually.
- **Average Cost.** This is a costing method that calculates product cost as the average (arithmetic mean) of all the purchase costs of an inventoried product.

You calculate these values outside of the Siebel application, and then you can enter all four of these costs for each product or service that is a line item in the cost list. You use the Cost List record to choose which of these cost methods is used for all the line items in the list.

You can attach the same cost list to multiple price lists. For example, you may have two price lists with different prices for the same products, one with the prices for customers and the other with the prices for resellers. Although the products may have the same cost, regardless of the price you charge for them, you can attach the same cost list to both of these price lists.

Process of Creating Cost Lists

To create costs lists, perform the following tasks:

1. *Creating the Price List or Rate List the Cost List Will Reference*
2. *Creating the Cost List Record*
3. *Creating the Cost List Line Item Records*
4. *Entering the Indirect Costs*

5. *Associating the Cost List with a Price List or Rate List*
6. *Updating a Price List's Costs*
7. *Viewing the Product Cost and Margin*

Creating the Price List or Rate List the Cost List Will Reference

This task is as step in *Process of Creating Cost Lists*.

As a prerequisite for creating a cost list, create the price list or the rate list with which the cost list will be associated. The cost list will include the cost of products included in the price list or the hourly cost of the services included in the rate list. For more information about creating price lists, see *Creating and Assigning Price Lists* For more information about creating rate lists, see *Creating and Assigning Rate Lists*

Creating the Cost List Record

This task is as step in *Process of Creating Cost Lists*.

The cost list record includes general information about the cost list as a whole, such as its name, the organizations that can use it, and its cost method.

To create the cost list record

1. Navigate to the Administration - Pricing screen, then Cost Lists view.
2. In the Cost Lists list, add a new record and complete the necessary fields, as described in the following table.

Field	Comments
Cost List	Enter a unique name for this cost list. If you have many cost lists, then it is important to use meaningful names and a consistent naming convention for them all.
Description	Enter a description of this cost list for your own use.
Effective From	Enter the date and time when this cost list will become effective. By default, the application assigns the current system date and time when you first create the Cost List record.
Effective To	Enter the date and time when this cost list will become ineffective. After this time, Siebel applications will not use this cost list.
Cost Method	Select one of the four cost methods: Standard, Average, Next, or Last. These cost methods are described in *About Cost Lists*.
Created By	This field displays the user name of the person creating this record.
Created	This field displays the date and time when the record was created.

Field	Comments
Organization	Select the organizations that will use this cost list.
Updated By	This field displays the user name of the last person who updated this record.
Updated	This field displays the date and time when the record was last updated.

Creating the Cost List Line Item Records

This task is as step in *Process of Creating Cost Lists*.

You must add a line item to the cost list for each product or service in the cost list. The cost list line item has specific information about the cost of that product or service.

You can enter four costs for each product, using the four cost methods described in *About Cost Lists*. The cost method used depends on the cost method you choose in the Cost List record.

This record is used only for the direct cost you pay for the product. Indirect costs associated with the product are entered later.

Note: Alternatively, you can add a product to multiple cost lists using the Administration - Product screen, then Cost List view.

To create a cost list line item

1. Navigate to the Administration - Pricing screen, then Cost Lists view.
2. In the Cost Lists list, select the cost list to which you want to add a line item.
3. Click the Cost List Line Items view tab.
4. In the Cost List Line Items list, click New.

   The Add Internal Products dialog box appears.
5. In the Add Internal Products dialog box, select the product you want to add to the price list, and click Add.
6. Complete the fields in the Cost List Line Item record, as described in the following table.

Field	Comments
Product	The name of the product or service that you selected from the Add Products dialog box is entered here.
Part #	The part number of the product or service that you selected from the Add Products dialog box is entered here. This number uniquely identifies the product or service.

Field	Comments
Standard Cost	Enter the standard cost, as described in *About Cost Lists*.
Last Cost	Enter the last cost, as described in *About Cost Lists*.
Next Cost	Enter the next cost, as described in *About Cost Lists*.
Average Cost	Enter the average cost, as described in *About Cost Lists*.
Start Date and End Date	If the line item has different effective dates from the cost list, then enter the start date when the costing becomes effective and the end date when it is no longer effective. These fields work in the same way as the start and end date fields of price list line items, as described in *Giving a Product Multiple Prices with Different Effective Dates*.

Entering the Indirect Costs

This task is as step in *Process of Creating Cost Lists*.

After entering the direct cost of the product, you can also enter indirect costs associated with the product, such as your cost of ordering products, receiving products, and shipping products to customers. These indirect costs apply to all the products in the cost list. Enter an average cost for a unit for all your products.

To enter indirect costs

1. Navigate to the Administration - Pricing screen, then Cost Lists view.
2. In the Cost Lists list, select the cost list to which you want to add indirect costs.
3. Click the Cost List Indirect Costs view tab.
4. In the Cost List Indirect Costs Items list, click New.
5. Complete the fields in the Cost List Indirect Costs record, as described in the following table, and continue to add these records until you have entered all indirect costs.

Field	Comments
Expense Object	Select the type of indirect cost, such as shipment, receipt, or demonstration.
Cost Per Unit	Enter the cost for each unit of this indirect cost, spreading the total indirect cost across all the products or services in the cost list.
Comments	Enter a comment for your own use.

Associating the Cost List with a Price List or Rate List

This task is as step in *Process of Creating Cost Lists*.

After creating the cost list, you can associate it with either a price list or a rate list. You can associate the same cost list with more than one price list or rate list.

Associating a Cost List with a Price List

Use the following procedure to associate a cost list with a price list.

To associate a cost list with a price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list with which you want to associate the cost list.
3. In the More Info form, go to the Cost List field and select the cost list.

Associating a Cost List with a Rate List

Use the following procedure to associate a cost list with a rate list.

To associate a cost list with a rate list

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate Lists list, select the rate list with which you want to associate the cost list.
3. In the More Info form, go to the Cost List field and select the cost list.

Updating a Price List's Costs

This task is as step in *Process of Creating Cost Lists*.

If the cost list is associated with a price list, then you must update the cost for that price list to make the new cost list take effect.

This is not necessary if the cost list is associated with a rate list.

To update a price list's costs

1. Navigate to Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list whose cost you want to update.
3. In the Price Lists list, from the applet level menu, select Update Cost from the menu.

Viewing the Product Cost and Margin

This task is as step in *Process of Creating Cost Lists*.

Now that cost lists have been set up and associated with a price list or rate list, any user can view the cost and the profit margin of products on that price list. The margin is displayed as a percentage.

Note: Before viewing the cost and margin for a rate list, you must configure the product using Web Tools to expose those fields. For more information about using Web Tools, see *Using Siebel Tools* .

Viewing a Product Cost and Margin in a Price List

Use the following procedure to view a product cost and margin in a price list.

To view a product cost and margin in a price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list for the product whose cost you want to view.
3. Click the Price List Line Items view tab.
4. In the Price List Line Items list, select the product whose cost you want to view.
5. Navigate down to view the Price List Line Item Detail form.
6. If necessary, click the Show More button in the Price List Line Item Detail form.

   The product cost is in the Cost field and the profit margin is in the % Margin field of the Price List Line Item Detail form.

Viewing a Product Cost and Margin in a Rate List

Use the following procedure to view a product cost and margin in a rate list.

To view a product cost and margin in a rate list

1. Navigate to the Administration - Pricing screen, then Rate Lists view.
2. In the Rate Lists list, select the rate list for the product whose cost you want to view.
3. Click the Rate List Line Items view tab.
4. In the Rate List Line Items list, select the product whose cost you want to view.

Using Asset Mapping to Value Assets

When you use asset mapping, an asset is valued as a percentage of its value on the cost list. The percentage can be based on the current condition of the asset. For example, a new asset in excellent condition would be 100% of the cost list value, while a used asset in average condition would be 40% of the cost list value.

Adjustments depend on the Condition, Value Basis, Cost List, and Cost Method fields for each asset. These factors are used to calculate the Asset Values and the Replacement Costs on the Assets screen, Value view.

If there are no asset mapping records and a cost list is specified, then the Assets screen, Value view uses the asset cost from the Cost List Line Items view. For more information about assets, see *Siebel Field Service Guide* .

Note: Asset mapping applies to product lines and to all assets that belong to the same product line and cost list.

To map the cost of assets for specific product lines

1. Navigate to the Administration - Pricing screen, then Cost List screen.
2. In the Cost Lists list, select a cost list.

3. Click the Asset Mapping view tab.
4. In the Asset Mapping form, add a new record and complete the necessary fields, as described in the following table.

Field	Comments
Cost Field	Select the value that is used as the basis of the cost calculation. options are as follows: o **Asset Value.** This calculation is based on the original cost of the product. o **Replacement.** This calculation is based on the cost list value of the product.
Condition	Select the condition of the asset.
Value Basis	Select the value basis of the asset. The value of the asset is adjusted based on your selection. options are as follows: o **Scrap.** This is the value of the asset as scrap. In many cases, this involves depreciating to zero. o **Original.** This is the original basis for the value of the asset. o **Adjusted.** This is an adjustment based on a one-time event. For example, you could use this if you drop the asset and the damage reduces its value. o **New.** This is an adjustment for a new product. For example, a new car drops in value as soon as you drive it off the car lot. o **Refurbished.** This is the value of the asset when it is refurbished. o **Used.** This is the value of the asset when it is sold as used.
Cost Method	This is the same cost method used in the Cost List Line Items view.
Factor	You can leave the Condition and Value Basis fields blank and instead enter a value in the Factor field, which specifies the percentage of its cost at which this asset is valued. If Condition or Value Basis are entered, then they override the Factor field.

6 Creating Volume Discounts

Creating Volume Discounts

This chapter describes the two different types of volume discounts and the process of setting up a volume discount. It also covers special features of volume discounts for products with components. This chapter covers the following topics:

- *About Volume Discounts*
- *Process of Setting Up a Volume Discount*

About Volume Discounts

A volume discount is an adjustment to the price of a product based on the quantity of that product in the quote line or order line item.

When you define a volume discount, you select one of two discount methods:

- **Simple.** This method applies the same discount to every line item. The discount for all items depends on the total quantity of that item purchased.
- **Tiered.** This method applies the discount defined by each Volume Discount Line Item record to the quantity of items defined in that Volume Discount Line Item record.

For example, you create a volume discount with line items that:

- Apply a 10 percent discount when the quantity is five to 10 items
- Apply a 20 percent discount when the quantity is 11 to 20 items
- Apply a 30 percent discount when the quantity is 21 items or more

If this is a simple volume discount, and if the customer buys 23 items, then the customer gets a discount of 30 percent on all 23 items.

If this is a tiered volume discount, and if the customer buys 23 items, then the customer gets no discount on items one to four, a 10 percent discount on items five to 10, a 20 percent discount on items 11 to 20, and a 30 percent discount on items 21 to 23.

Volume discounts apply to the quantity in the quote line or order line, not the total quantity in the entire quote or in an entire order. If a user splits an order for a product into two or more lines on a quote, then the volume discount calculation for that product is not based on the total of the two lines. However, you can configure the application to base volume discounts on multiple line items, as described in *Volume Discounts Across Line Items*.

The volume discount applies to the start price after attribute pricing and customizable product pricing adjustments have been applied.

> **Note:** If you use volume discounts with other types of price adjustments, then the final price depends on the order in which Siebel Pricer applies the pricing adjustments. For more information, see *About the Processing Order of Price Adjustments*.

For more information about volume discounts, see the following information:

- *About Reusing Volume Discounts for Multiple Currencies*
- *Volume Discounts and Products with Components*
- *Volume Discounts Across Line Items*
- *Process of Setting Up a Volume Discount*

About Reusing Volume Discounts for Multiple Currencies

You can use the same volume discount schedule for price lists in different currencies.

You can create the volume discount schedule in one currency. When the salesperson creates a quote, the salesperson uses this volume discount schedule but specifies other currencies for the line items. While calculating the volume discounts, the pricing procedure automatically converts the currency of price adjustments in the volume discount to the currency of the quote or order line item.

Volume Discounts and Products with Components

Products with component are discussed in *Setting Up Pricing for Products with Components*. This topic explains how volume discounts apply to component products.

The pricing procedure completes the pricing for the product with components before applying any volume discounts. For more information, see *About the Processing Order of Price Adjustments*.

Volume discounts can apply to the components of a customizable product. For example, if disk drives are one component of a computer, then you can give volume discounts based on the number of disk drives within a single instance of a disk drive in the customizable product. The volume discount applies to an individual line item within a customizable product, and takes into account the existence of multiple instances of the disk drive within the customizable product or quote. However, you can configure the product so it applies discounts based on the quantity in multiple line items of an order, as described in *Volume Discounts Across Line Items*.

Volume Discounts Across Line Items

You can modify the pricing procedure to create volume discounts that work across line items.

The pricing procedure is a PSP procedure. For more information about PSP procedures and about the Aggregate Transform, see *Siebel Order Management Infrastructure Guide* .

Process of Setting Up a Volume Discount

To set up a volume discount, perform the following tasks:

1. *Creating a Volume Discount Record*
2. *Creating Volume Discount Line Items*
3. *Linking the Volume Discount to a Price List or Discount Matrix*

Creating a Volume Discount Record

This task is a step in *Process of Setting Up a Volume Discount*.

A volume discount record provides the top-level information about a volume discount, including its name, whether it is a simple or tiered volume discount, its start and end dates, and its currency.

CAUTION: If you delete a volume discount record, then all references to this discount become invalid. Any price list line items that used this discount will no longer have a volume discount.

To create a volume discount record

1. Navigate to the Administration - Pricing screen, then Volume Discount view.
2. Add a new record to the Volume Discounts list and complete the necessary fields, as described in the following table.

Field	Comments
Name	Enter a name for this volume discount.
Discount Method	Select a method from the picklist. The options are Simple and Tiered, as described in *About Volume Discounts*.
Start Date	Enter the date when this volume discount becomes effective.
End Date	Enter the date when this volume discount becomes ineffective. If the volume discount will not expire, then leave this field blank.
Currency	Enter the currency in which the volume discount is applied.
Exchange Date	If you are converting currencies, then enter the date for the currency exchange rate. The Siebel application will use the exchange rate on that day to convert currency.

Creating Volume Discount Line Items

This task is a step in *Process of Setting Up a Volume Discount*.

Volume discount line items include information about quantity range and the corresponding price adjustments.

Add a volume discount line item for each quantity range in the volume discount. For example, if you want to apply a 10 percent discount when the quantity is between five and ten, and a 20 percent discount when the quantity is 11 or more, then you must add two volume discount line items.

When you add volume discount line items, be careful not to create gaps or overlapping quantities, which result in the following errors:

- If you leave a gap in a simple discount, then quantities that fall within the range of the gap receive no discount at all.
- If you leave a gap in a tiered discount, then quantities that fall within the range of the gap default to the lower tier.
- If you create overlapping discount items and quantity ranges, then the discounting behavior is not predictable.

How the discounts in the line items are applied depends on whether this is a simple or tiered volume discount, as described in *About Volume Discounts*.

To create a volume discount line item

1. Navigate to the Administration - Pricing screen, then Volume Discount view.
2. Select the volume discount for which you want to create line items.
3. Click the Volume Discounts Line Items view tab.
4. In the Volume Discounts Line Items list, add a new record and complete the necessary fields, as described in the following table.

Field	Comments
Name	Enter a name for this volume discount line item. This name appears in the quotes that your sales representatives create for customers, so use a name that describes the discount, such as “10 percent Discount for Buying 5 to 10." The value in this field is also used to create an upsell message about the additional quantity to purchase in order to get a bigger discount.
Min Qty	Enter the minimum quantity that must be purchased to get this discount rate.
Max Qty	Enter the maximum quantity that gets this discount rate. If this field is blank, then this discount rate applies to all quantities greater than the minimum quantity.
Adjustment Type	Select the type of the adjustment that the discount will apply. Options are Discount Amount, % Discount, Markup Amount, % Markup, Price Override.
Adjustment Amount	The value in the Adjustment Type field determines how the adjustment amount is applied, as follows: ○ If the Adjustment Type is Discount Amount, then the value in the Adjustment Amount field is subtracted from the price. ○ If the Adjustment Type is % Discount, then the value in the Adjustment Amount field is treated as a percentage and is subtracted from the price. For example, if the Adjustment Amount is 10, then 10 percent is subtracted from the price. ○ If the Adjustment Type is Markup Amount, then the value in the Adjustment Amount field is added to the price. ○ If the Adjustment Type is % Markup, then the value in the Adjustment Amount field is treated as a percentage, and is added to the price. ○ If the Adjustment Type is Price Override, then the value in the Adjustment Amount field is used instead of the price.

Linking the Volume Discount to a Price List or Discount Matrix

To apply the volume discount, you have to link it to a price list line item or a discount matrix line item. This task is a step in *Process of Setting Up a Volume Discount*.

Linking a Volume Discount to a Product in a Price List

Use the following procedure to link a volume discount to a product in a price list.

To link a volume discount to a product in a price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list to which you want the volume discount to apply.
3. Click the Price List Line Items view tab.
4. Select the line item you want.
5. In the Price List Line Items Detail form, in the Volume Discount field, select the volume discount you want to apply to that line item.

Linking a Volume Discount to a Line Item in a Matrix

Use the following procedure to link a volume discount to a line item in a discount matrix.

To link an volume discount to a line item in a discount matrix

1. Navigate to the Administration - Pricing screen, Discount Matrices, and then Discount Details view.
2. In the Discount Details form, in the Volume Discount field, select the volume discount.

7 Creating Aggregate Discounts

Creating Aggregate Discounts

This chapter covers aggregate discounts and includes the following topics:

- *About Aggregate Discounts*
- *Buy and Receive Products for Aggregate Discounts*
- *Process of Creating Aggregate Discounts*
- *Aggregate Discounts for Buy One, Get One Free Discounts*
- *Aggregate Discounts for Products with Components*

Note: As an alternative to the methods described in this chapter, you can create aggregate discounts using a discount matrix, as described in *Creating Discount Matrices*.

About Aggregate Discounts

When you define aggregate discounts, you specify a bundle of items and the discount that applies to specific items if the user buys the entire bundle.

For example, you might specify that customers who buy a dining room table and four chairs would get 10 percent off on the price of the chairs.

In addition to the method described in this chapter, there are two other ways to create aggregate discounts:

- **Simple Product Bundles.** In simple cases, you can create a simple product bundle with all the products in the promotion and create a special price for the bundle. For example, create a product bundle with a dining room table and four chairs, and set a price for this bundle that is less than the total of the individual prices of these items. For more information about simple bundle pricing, see *Creating a Price List Line Item for a Simple Product Bundle*.
- **Product Promotions.** Product promotions allow you to create bundled promotions, to show these promotions in quotes, orders, and catalogs, and to recommend these promotions to customers who buy products in the bundle. This is usually the preferred method of creating bundle discounts. For more information, see *Creating Product Promotions*.

  Note: If you use an aggregate discount with other types of price adjustments, then the final price depends on the order in which Siebel Pricer applies the pricing adjustments. For more information, see *About the Processing Order of Price Adjustments*.

Buy and Receive Products for Aggregate Discounts

When you add the components to an aggregate discount, you specify whether items are *buy* or *receive* products:

- **Buy products.** To qualify for the aggregate discount, the customer must buy at least the specified quantity of all the Buy products in the list. The Buy products themselves may be discounted if the customer qualifies for the aggregate discount.
- **Receive products.** The Receive products are discounted if the customer qualifies for an aggregate discount. There is no requirement to buy a specific number of Receive products.

Some examples of Buy and Receive products in aggregate discounts are as follows:

- If you buy a certain model of desk, then you get 50 percent off on a certain model of ergonomic chair. (The desk is a Buy product with a quantity of 1 and no price adjustment. The chair is a Receive product with a quantity of 1 and a price adjustment of 50 percent discount.)
- If you buy 10 desks, then you get 10 percent off on the desks and 50 percent off on the ergonomic chairs for them. (The desk is a Buy product with a quantity of 10 and a price adjustment of 10 percent discount. The chair is a Receive product with a quantity of 10 and a price adjustment of 50 percent discount.)
- If you buy 10 desks, then you get 10 percent off on the desks and 50 percent off on any model of desk chair for them. (The desk is a Buy product with a quantity of 10 and a price adjustment of 10 percent discount. Each model of desk chair is a Receive product with a quantity of 10 and a price adjustment of 50 percent discount.)
- If you buy 10 desks, then you get 10 percent off on the desks. (The desk is a Buy product with a quantity of 10 and a price adjustment of 10 percent discount. There is no Receive product.)

  Note: The Buy product quantity is a minimum and the Receive product quantity is a maximum. For example, the bundling discount says that if you buy 10 desks, then you get 50 percent off on 10 ergonomic chairs for them. You must then buy a minimum of 10 desks to get the discount, and you can get a discount on a maximum of 10 chairs. This prevents a customer from buying 10 desks in order to get a discount on thousands of chairs.

Process of Creating Aggregate Discounts

By default, aggregate discount functionality is not designed to be used with promotions that include customizable products. If you want to use aggregate discount functionality with promotions that include customizable products, then it is recommended that you redesign the workflow with regard to the specific business scenario. For more information about the aggregate discount workflow, see *Pricing Procedure - Aggregate Discounts Workflow*.

To create aggregate discounts, perform the following tasks:

1. *Adding an Aggregate Discount Record*
2. *Defining Aggregate Discount Sequence*
3. *Attaching an Aggregate Discount Sequence to a Price List*

Adding an Aggregate Discount Record

This task is a step in *Process of Creating Aggregate Discounts*.

To define an aggregate discount, you enter general information about the group of products in the Aggregate Discount record and add all the products in the group to the Discount Details list.

To define an aggregate discount

1. Navigate to the Administration - Pricing screen, then Aggregate Discounts view.
2. Add a record to the Aggregate Discounts list, and complete the necessary fields in the record and the More Info form, as described in the following table.

Field	Comments
Name	Enter a name for the aggregate discount.
Currency	Select the currency for the aggregate discount.
Exchange Date	If multiple currencies are used for the products in the bundle, then enter the date of the exchange rate that will be used for currency conversion.
Effective From	Enter the start date when this discount becomes active.
Effective To	Enter the end date when this discount is no longer active.
Active	Select this check box to activate the aggregate discount. The discount is active only if the date of an order is between the Effective From and Effective To dates, and you have selected this check box so the aggregate discount is active.

3. Click the Discount Details view tab.
4. For each product in the bundle, add a record to the Discount Details list and complete the necessary fields, as described in the following table.

Field	Comments
Product Name	Select the product.
Buy	If it is a Buy product, then select this check box. For more information, see *Buy and Receive Products for Aggregate Discounts*.
Receive	If it is a Receive product, then select this check box.

Field	Comments
	Note: You cannot select both the Buy and Receive check box in a single record. If you want to make a product both a Buy and a Receive product, then add two records for that product, and select Buy in one and Receive in the other.
Qty	Enter the quantity of the product. For Buy products, this is the minimum quantity that you must buy to receive the aggregate discount. For Receive products, this is the maximum quantity that can receive a discount. For more information, see *Buy and Receive Products for Aggregate Discounts*.
Adjustment Type	Select the type of price adjustment applied to this product. Options are Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.
Adjustment Amount	Enter the amount of the Adjustment. How this value is used depends on the selection in the Adjustment Type field: o **Discount Amount.** The adjustment amount is subtracted from the product price. o **% Discount.** The adjustment amount is treated as a percentage of the product price, which is subtracted from the product price. o **Markup Amount.** The adjustment amount is added to the product price. o **% Markup.** The adjustment amount is treated as a percentage of the product price, which is added to the product price. o **Price Override.** The adjustment amount is used as the product price, overriding the original price.

Defining Aggregate Discount Sequence

Aggregate discount sequences are used to define the stacking logic for aggregate discounts. This logic can define:

- The order in which aggregate discounts are applied.
- Which aggregate discounts are mutually exclusive.
- Which aggregate discounts can be “stacked," or applied in conjunction with one another.

You can use aggregate discounts within a quote or an order only if they are part of an aggregate discount sequence. Therefore, you must define an aggregate discount sequence even if you have only one aggregate discount. An aggregate discount can be a part of many aggregate discount sequences.

Aggregate discount sequences are linked with price list and discount matrix line items (like volume discounts and attribute adjustments). At run time, the pricing procedure picks the aggregate discount sequence to be used and evaluates the aggregate discounts to be applied.

The order in which aggregate discounts are applied within an aggregate discount sequence can affect the resulting price. For example, if aggregate discount (a) gives a discount of $100, and aggregate discount (b) gives you 10% off the price of the product, the resulting price would be lower if aggregate discount (b) is applied before aggregate discount (a).

To apply all the aggregate discounts in a sequence, you attach the sequence to either a price list or a product with component line items. A sequence includes all the aggregate discounts that apply to one price list or to one product with components in a price list.

Use the following procedure to define the aggregate discount sequence. This task is a step in *Process of Creating Aggregate Discounts*.

To define the aggregate discount sequence

1. Navigate to the Administration - Pricing screen, then Aggregate Discount Sequences view.
2. Add a record to the Aggregate Discount Sequence list and complete the necessary fields as described in the following table.

Field	Comments
Name	Enter a name for the aggregate discount sequence.
Effective From	Enter the date when the aggregate discount sequence becomes effective.
Effective To	Enter the date when the aggregate discount sequence is no longer effective.
Active	Select this check box to activate the aggregate discount sequence.

3. Click the Aggregate Discounts view tab.
4. Add records to the Aggregate Discounts list for every aggregate discount that you want in this sequence and complete the necessary fields, as described in the following table.

Field	Comments
Sequence	Enter the sequence in which this aggregate discount is applied.
Name	Select the aggregate discount.
Next Discount if Used	Select the next aggregate discount in the sequence to be used if this discount is applied.
Next Discount if not Used	Select the next aggregate discount in the sequence to be used if this discount is not applied.

Attaching an Aggregate Discount Sequence to a Price List

This task is a step in *Process of Creating Aggregate Discounts*.

You can attach the aggregate discount sequence to a price list and, with some restrictions, also to price list line items. In the case of price list line items, you can attach an aggregate discount sequence to a product with components.

For example, with a computer system, you might want to attach the aggregate discount sequence to a product with components to give discounts to customers who buy a specific CD-ROM drive, monitor, and keyboard.

Attaching an Aggregate Discount Sequence to a Price List

Use the following procedure to attach an aggregate discount sequence to a price list.

To attach an aggregate discount sequence to a price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list to which you will attach the sequence.
3. In the More Info form, in the Aggregate Discount Sequence field, select the sequence to attach to this price list.

   The aggregate discounts in this sequence apply to all prices in this price list.

Attaching an Aggregate Discount Sequence to a Price List Line Item

Use the following procedure to attach an aggregate discount sequence to a price list line item.

To attach an aggregate discount sequence to a price list line item

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. Click the Price List Line Item view tab.
3. In the Price List Item Detail form, in the Aggregate Discount Sequence field, select the aggregate discount sequence.

   You can modify this field only if the product is a product with components.

Aggregate Discounts for Buy One, Get One Free Discounts

Aggregate discounts are generally used to reduce the price of one product when it is sold with one or more different products. For example, a customer could buy a dining room table at full price and receive a discount on dining room chairs.

Aggregate discounts can also be used where you buy a product and receive a discount on the same product. This is commonly called a *Buy One, Get One Free* discount.

To set up this sort of discount, combine them in one aggregate discount with a Buy quantity equal to the total number of products a customer must buy, and the discount set to give the appropriate result. For example, if customers can buy one dining room chair and get one chair for free:

- Set the Buy quantity to two.
- Set the price Adjustment Type to % Discount and the Adjustment Amount to 50 to create a Receive rule that gives 50 percent off the List price for each product.

Similarly, if customers can buy three dining room chairs and get one chair for free:

- Set the Buy Quantity to four.

- Set the price Adjustment Type to % Discount and Adjustment Amount to 25 to create a receive rule that gives 25 percent off the List price for each product.

Aggregate Discounts for Products with Components

Aggregate discounts can be associated with component-based products. For example, for customers buying personal computers, you might create an aggregate discount so that if they buy both a CD-ROM drive and a floppy disk drive, then they get a 10 percent discount on both drives.

Aggregate discounts for product components work in the same general way as other aggregate discounts, but there are several additional constraints:

- Aggregate discounts for product components must include only components of the product. Do not create an aggregate discount that mixes components of one component-based product with components of other component-based products.
- The aggregate discount is applied once the product with component has the required quantity of the Buy and the Receive products.
- If you are defining the required quantities for a component-based product component in a bundling discount, then be aware of the constraints defined for the product, especially the Max Cardinality limit. Do not define aggregate discounts that require the user to buy more products than the cardinality limits. If the pricing rule exceeds cardinality, then it will not be applied.

> **Note:** If the same product appears multiple times as a component, then the different instances of the product may have different prices after all adjustments. In this case, the aggregate discount uses as its target price the instance of the product with the highest net price. For example, if a specific memory chip appears multiple places within a customizable computer system product, then the highest calculated price for that memory chip will be used as the target price for all bundling discounts on that chip.

8 Creating Product Promotions

Creating Product Promotions

This chapter covers product promotions and includes the following topics:

- *About Product Promotions*
- *Process of Creating Product Promotions*
- *Creating Product Promotions That Upgrade Existing Promotions*
- *Viewing a Summary of a Product Promotion*
- *Managing Versions of Product Promotions*
- *Viewing the Schedules of All Product Promotions*
- *About Product Promotions for Customizable Products*

About Product Promotions

Product promotions are created in the Administration - Product screen. They represent the marketing definition of a product. Product promotions are time-sensitive, and they state discounted prices and contractual terms.

Product promotions may be broadly classified to belong to either of these two types:

- **Coupons.** This promotion includes a single product or service. This sort of promotion is used primarily for retail sales and is also called a retail promotion.
- **Bundled promotions.** This promotion includes a group of products or services.

End users can select product promotions in two ways:

- **Top-down selection.** If the customer receives an offer or sees an advertisement for the promotion, then the customer can order the promotion. The customer or sales representative adds the promotion to the order, and all of the products in the promotion are added as line items to the quote or order. This selection method usually applies to bundled promotions.
- **Bottom-up selection.** If the customer adds products to a quote or order that have product promotions available, then the sales representative or the application can recommend the promotion. When a customer is purchasing some products in a bundle promotion, the application can automatically recommend the bundle promotion. There can be an auto-recommendation if the customer has these products in a quote or order as new items, and if they are not already covered by another promotion.

> **Note:** A product promotion is a type of product with which you can associate class or product line. Although a class can be associated, product promotions cannot have child components (structure) or configurator constraints. Class and product line association is useful for defining aggregate relationships for upgrade paths.

For more information about how end users work with product promotions, see *Siebel Order Management Guide* .

Process of Creating Product Promotions

When you create a new Product Promotion record, a version named Work Space is automatically created and the Locked check box is automatically selected in the Product Promotions screen, Versions view. This check box must be selected to change the Effective Dates of the promotion. You do not have to lock the version to change any other information about a promotion, but to see the changes at run time, the user must select Refresh Cache from an applet-level menu item of the promotion header form. For more information, see *Managing Versions of Product Promotions*.

To create a product promotion, perform the following tasks:

1. *Setting Up Image Files for Product Promotions*
2. *Creating the Product Promotion Record*
3. *Specifying Pricing for Product Promotions*
4. *Defining Eligibility Rules for Product Promotions*
5. *Defining Compatibility Rules for Product Promotions*
6. *Customizing Recommendation Messages for Product Promotions*
7. *Creating Translations of Recommendation Messages for Product Promotions*
8. *Specifying Terms and Conditions of Product Promotions*
9. *Releasing the Product Promotion*
10. *Activating Workflows for Product Promotion*

Setting Up Image Files for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

When you create a product promotion, you can associate an image file with it. This image will be displayed next to the promotion in catalogs. If you want to show the image in catalogs, then you must create the image and import it into the Siebel application in the Administration - Document screen. This makes it available to associate with the product promotion record.

To set up image files for product promotions

1. Create the images in either JPG or GIF format.
2. Navigate to the Administration - Document screen, then Literature view.
3. In the Literature list, click New File.
4. In the Choose File dialog box, select the image file.

Creating the Product Promotion Record

This task is a step in *Process of Creating Product Promotions*.

Create the Product Promotion record, which contains general information about the entire product promotion.

> **Note:** When you create a Product Promotion record, you specify the effective dates for the promotion. Before you do this, it can be useful to view a schedule of all your product promotions. For more information, see *Viewing the Schedules of All Product Promotions*.

To create a product promotion record

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. Add a new record to the promotions list and complete the fields as described in the following table.

Field	Comments
Name	Enter a name for the promotion.
Promotion ID	This field displays a unique, Siebel-Pricer-generated ID for the promotion.
Type	Select the type of the promotion. The options are coupon and bundled promotion. For more information about these types, see *About Product Promotions*.
Image	Select an image file that will display if this promotion is shown in a catalog. Before you can select this, you must set up the image, as described in *Setting Up Image Files for Product Promotions*.
Period	Enter the number of periods to specify the duration of your promotion. For example, if the selected unit of measurement in the Period UOM field is years, then you enter a value to correspond to the number of years the promotion is available.
Period UOM	The unit of measure for the promotional time period, for example, years, months, days.
Description	Enter a description for the promotion.
Organization	Select one or more organizations to give them visibility to this promotion. Users who are not in these organizations cannot see this promotion in the Siebel application. For more information about organizations, see *Siebel Security Guide* .
Campaign	If the promotion is part of a marketing campaign, then select the campaign. For more information about campaigns, see *Siebel Marketing Installation and Administration Guide* .
Score	Enter the ranking that determines where on the list of product promotion recommendations this promotion will appear when the end user displays recommendations. Promotions with higher scores appear higher on the list. For more information, see the topic about recommendations in *Siebel Order Management Guide* .
Effective Dates	This field displays the start date and end date of the period when this promotion is effective. Users can edit these dates in the Versions view.

Field	Comments
Message	Enter a message for the promotion.
Grace Period	Enter a value for the period of time before a penalty is imposed on a customer. For example, if the selected unit of measurement in the Grace Period UOM field is days, then you enter a value to correspond to the number of days before the customer is subject to a penalty.
Grace UOM	The unit of measure for the grace time period, for example, years, months, days.
Instances	Select a value that specifies how many times a customer is allowed to use this promotion. The options are One Per Customer, One Per Order, and No Limit.
Show in Catalog	Select this check box to display the promotion in the product catalogs, so the end user can choose it in quotes and orders the same way the end user chooses a product. If you select the check box for bundle promotions, then end users can select the bundle promotion in a catalog to add all the products in the promotion to a quote or order. You generally do not select this check box for coupon promotions. For coupon promotions, end users generally add the product, and then select a coupon promotion that is available for that product.
Track as Asset	Select this check box to track the promotion as an asset. This is necessary if you provide ongoing service to the buyer of the promotion, or have some other reason to keep a record of which buyers have purchased the promotion. For more information, see the topic about asset-based ordering in *Siebel Order Management Guide* . **Note:** This check box applies only to the promotion, not to the products in it. To track products within the promotion as assets, you must select the Track as Asset check box for these products in the Administration - Products screen, Products view.
Track as Agreement	Select this check box to track the promotion as an agreement. This is necessary if the agreement involves some ongoing commitment to the customer that you must refer to in the future. For more information about agreements, see *Siebel Project and Resource Management Administration Guide* . **Note:** If you want to create a penalty record, then you must select the Track as Agreement flag on the promotion definition record.

Note: These fields are also available for relationships of a customizable product which is component of the child promotion. However you must define these fields at relationship level.

Specifying Products for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

Every promotion applies to specific products. The promotion may apply to one product, such as a particular model of table and chair, or it may apply to many products, such as all of your dining room tables.

If you are creating a bundle product promotion, then list all the products in the bundle in the Products list. The customer must buy all these products to get a discount.

If you are creating a coupon product promotion, then you do not have to list the product here. Add the product and the price discount for it in the Product Promotion screen, Pricing view.

You can add the products covered by a promotion using the Product Promotions, Products, and then Components view. For example, if you give customers a discount if they buy a specific model of table and four chairs, then add that table to the Products view with a minimum quantity of 1, and add the chair with a minimum quantity of 4.

You can also add product lines or product classes to a promotion using the Product Promotions, Products, and then Aggregate view. For example, if you have a product line named Dining Room Tables, then you can add it to the promotion so that the promotion applies to all dining room tables. You specify the default dining room table that is included in the promotion, but the customer can select any product in the Dining Room Table product line to get the bundle discount.

Specifying Individual Products for a Product Promotion

Use the following procedure to specify individual products for a product promotion.

To specify individual products for a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are specifying products.
3. Click the Products view tab.

   If necessary, in the link bar of the Products view, click Components.
4. For each product that is covered by the promotion, add a new record to the Product Components list and complete the necessary fields as described in the following table.

Field	Comments
Product Name	Select a product that is covered by the promotion. Products are available in the dialog box if they are released and have a currently active version. If this is a product with components, then you can click the product name to display the explorer view used to define products with components.
Description	Enter a description of the product.

Field	Comments
Min	Enter the minimum number of this product that the customer must buy to qualify for this promotion, if there is a minimum requirement.
Max	Enter the maximum number of this product that the customer may buy to get the discount for this promotion, if there is a maximum limit.
Default	Enter the default quantity of this product that is included in the promotion, if you want to display a default quantity. If the customer selects this promotion in a quote or order, then the default quantity of all products in the promotion will be added as line items to the quote or order. This field is optional. If you do not enter a value here, then the application uses the value entered in the Min field as the default value.
Customizable	This check box is selected if the product selected in the Product Name field is customizable. The promotion can also be used to constrain the selections for the customizable product. For more information, see *About Product Promotions for Customizable Products*.
Recommendable	Select this check box to allow the Siebel application to recommend this product promotion to end users when the product exists in a quote or order and is not associated with any other product promotion.
Commitment	Select this check box if the product involves a commitment. For more information, see *Specifying Terms and Conditions of Product Promotions*. **Note:** Only bundled promotions have commitments. Coupon promotions do not have commitments.
Commitment Period	Enter the number of periods to specify the duration of the customer's commitment. For example, if the selected unit of measurement in the Commitment UOM field is years, then you enter a value to correspond to the number of years the commitment is valid.
Commitment Period UOM	The unit of measure for the commitment time period, for example, years, months, days.
Apply Component Charge	Select this check box to apply an early termination penalty charge based on when the component is terminated and on the proration plan. If this check box is not selected, then the component penalty charges are not added.

Field	Comments
Grace Period	Enter a value for the period of time before a penalty is imposed on a customer. For example, if the selected unit of measurement in the Grace Period UOM field is days, then you enter a value to correspond to the number of days before the customer is subject to a penalty.
Grace UOM	The unit of measure for the grace time period, for example, years, months, days.
Termination Charge	The amount that is charged if the component is terminated after end of grace period, but is terminated before the end of the commitment period.
Prorate Plan	This is calculated in the conditional charge plan. For more information, see *Siebel Field Service Guide* .

5. To enter translations of descriptions of the product to be shown in catalogs that are not in the default language, add a new record to the Translations list for each language you support and complete the necessary fields, as described in the following table.

Field	Comments
Language Name	This field displays the name of the language after you select the language code.
Language Code	Select the language code. For example, select ENU for American English.
Description	Enter the description of the promotion to be shown in catalogs and promotion recommendations using this language.

Note: The catalog in the default language shows the description that you entered in the Description field of the Product Component record, as described in Step 4.

Specifying Product Lines or Product Classes for a Product Promotion

Use the following procedure to specify product lines or product classes for a product promotion.

Note: Classes for promotions must not have attributes, because they are not supported in all scenarios.

To specify product lines or product classes for a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are specifying products.
3. Click the Products view tab, and in the link bar, click Aggregate.

4. For each product that is covered by the promotion, add a new record to the Product Aggregate list and complete the necessary fields as described in the following table.

Field	Comments
Product Line	Select a product line that is covered by the promotion. Select either a product line or a product class, but not both.
Product Class	Select a product class that is covered by the promotion.

   Complete the other fields in the same way as in the Components view.
5. In the Default Product view, click the Default check box for each product that is a default product for the promotion and enter the default quantity for each.
6. To enter descriptions of the promotion to be displayed in Edit Promotion view, add a new record to the Translations list for each language you support and complete the necessary fields in the same way as in the Components view.

Specifying Pricing for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

All promotions involve giving customers price adjustments on some or all products. For example, a promotion may offer a discount on a table if the customer buys a set that includes the table and four chairs.

Use the Pricing view to define the price adjustments for this promotion. You can define two types of price adjustments:

- **Component.** Define price adjustments for individual products.
- **Aggregate.** Define price adjustments for multiple products. There are two types of aggregate rules:
  - **Aggregate Conditional Rules.** Define rules for all the products in a product class or product line. For example, you could give a 10 percent discount on all digital cameras and a 15 percent discount on all film cameras, if these are two product lines with products that have been included in the promotion.
  - **Aggregate Default Rules.** Define rules that are applied by default if the line item product does not match any individual product pricing rule or any conditional aggregate pricing rule.

Specifying Price Adjustments for Individual Products in a Product Promotion

Use the following procedure to specify price adjustments for individual products in a product promotion.

To specify price adjustments for individual products in a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are specifying pricing.
3. Click the Pricing view tab.
4. Click Components on the link bar in Pricing view.
5. In the Pricing Components list, add a record for each product whose price will be changed by the promotion and complete the necessary fields as described in the following table.

Field	Comments
Product Name	Select a product.
Adjustment Type	Select the type of price adjustment applied to this product. The options are Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.
Adjustment Value	Enter the value of the Adjustment. How this value is used depends on the selection in the Adjustment Type field: ○ **Discount Amount.** The adjustment value is subtracted from the product price. ○ **% Discount.** The adjustment value is a percentage of the product price, which is then subtracted from the product price. ○ **Markup Amount.** The adjustment value is added to the product price. ○ **% Markup.** The adjustment value is a percentage of the product price, which is then added to the product price. ○ **Price Override.** The adjustment value is used as the product price, overriding the original price.

Specifying Price Adjustments Applied to a Product Class or a Product Line in a Product Promotion

Use the following procedure to specify price adjustments applied to a product class or product line in a product promotion.

To specify price adjustments applied to a product class or product line in a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are specifying the pricing.
3. Click the Pricing view tab.
4. Click Aggregate on the link bar in the Pricing view.
5. In the Pricing Aggregate list, add a new record and complete the necessary fields as described in the following table.

Field	Comments
Type	Select the type of aggregate adjustment. The options are as follows: ○ **Aggregate – Default.** This is a default pricing rule. If the line item product does not match any individual product pricing rule or any conditional aggregate pricing rule, then the line item price is adjusted using this default rule. ○ **Aggregate – Conditional.** The adjustment is applied to product classes or product lines. When this value is selected a user must define the Group Field and Group Value fields.
Product Line	If you selected Aggregate - Conditional as the type, then you can select a product line to which the adjustment applies.

Field	Comments
Class	If you selected Aggregate - Conditional as the type, then you can select a class to which the adjustment applies.
Adjustment Type and Adjustment Value	Enter the price adjustment. These fields are used in the same way as in the Pricing Components view, which was described in the previous procedure.

6. If you selected Aggregate - Conditional as the type, then you can add more records to define adjustments for additional product lines and classes.

Setting Up Promotion Price Adjustments for Components

Administrators can set up adjustments to prices for components that are part of a customizable product in a promotion. When a user creates an order or quote for that customizable product in the promotion, the user can view the adjusted prices for the components. You can add promotion components to price lists in the same way that you add other components to price lists. The prices for components in a promotion supersede any other prices for those components.

For example, if a promotion includes a bundle that allows for free voice mail for wire line service or a 50% discount for mobile service, then the administrator can set up the voice mail component for the wire line service with no price and the voice mail component for the mobile service with a 50% discount. When a user creates an order that includes this bundle, the user can view the $0 price for voice mail if the customer wants the wire line service and the 50% discount price for voice mail if the customer wants the mobile service. If voice mail is priced at $10 for each month when it is not part of a promotion, then the user does not see this $10 monthly price.

Use the following procedure to set up promotion price adjustments for components.

Note: Setting up promotion price adjustments for components applies to Industry Applications only.

To set up promotion price adjustments for components

1. Navigate to the Administration - Product screen, then the Product Promotions view.
2. In the Promotions list, select the promotion for which you want to set up adjustments to component prices.
3. Click the Pricing tab.
4. In the Pricing Components list, select a customizable product for which you want to set up adjustments to component prices.

   Note: A check mark displays in the Customizable field for the products that you can customize.

5. Click the Product Name field to display the components for the customizable product.
6. For each component for which you want to set up a price adjustment, select the component, and add a price adjustment record to the Adjustments list, as described in the following table.

Field	Comments
Adjustment Type	Select the type of price adjustment applicable to the component. Options include: Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.

Field	Comments
Adjustment Value	Enter the value of the adjustment. How this value is used depends on the value in the Adjustment Type field: ○ **Discount Amount.** The adjustment value is subtracted from the component price. ○ **% Discount.** The adjustment value is a percentage of the component price and is subtracted from the component price. ○ **Markup Amount.** The adjustment value is added to the component price. ○ **% Markup.** The adjustment value is a percentage of the component price and is added to the component price. ○ **Price Override.** The adjustment value is used as the component price and overrides the original component price.
Maximum Price	Enter the price that the adjusted price cannot exceed.

7. (Optional) Click Verify Adjustments to view the component prices in time intervals after adjustments.

Defining Eligibility Rules for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

Promotions may be restricted to only some of your customers. For example, a promotion may be offered only to customers in certain states. You can define rules that specify which customers are eligible for the promotion. For more information, see the topic about eligibility rules in *Siebel Order Management Guide* .

Note: Effective in the 7.8.2 version, the Check Eligibility flag and Inclusive Eligibility flag have been moved to the promotion header from the Versions view.

To define eligibility rules for a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are defining the eligibility rules.
3. Click the More Info view tab, select the Check Eligibility and Compatibility check box.
4. Click the Eligibility and Compatibility view tab and in the Rules link bar, click Eligibility.
5. Add new records to the Rules list and complete the necessary fields, as described in *Siebel Order Management Guide* .

Defining Compatibility Rules for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

You might have policies that keep customers from using more than one product promotion at the same time. For example, you might say that a coupon for a product may not be used with other coupons for the same product, but that the coupon may be used with bundle promotions for the same product.

You can define compatibility rules for a product promotion to specify the other product promotions with which it may and may not be used. For more information, see the topic about compatibility rules in *Siebel Order Management Guide* .

To define compatibility rules for a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion for which you are defining compatibility rules.
3. Click the Eligibility and Compatibility Rules view tab and then in the link bar, click Compatibility.
4. Add new records to the Rules list and complete the necessary fields, as described in the topic about compatibility in *Siebel Order Management Guide* .

Customizing Recommendation Messages for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

By default, if the application recommends a promotion, then it displays a message in the following form:

```
Recommend promotion '[Prod Prom Name]': [Promotion Description]
```

You can customize the form of this message.

To customize the recommendation message for product promotions

1. Navigate to the Administration - Application screen and in the link bar, click Message Types.
2. In the All Message Types list, query to find the record with the name Promotion - Bottomup Recommendation.
3. In the Message Type view, enter the desired message in the Full Text field.

Creating Translations of Recommendation Messages for Product Promotions

This task is a step in *Process of Creating Product Promotions*.

You can translate the recommendation message in multiple languages. The language that is displayed depends on the language of the user's application.

To create translations of recommendation messages for a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion.
3. Click the Messaging view tab.
4. For each language in which the message will be displayed, add a record to the Messaging list and complete the necessary fields as described in the following table.

Field	Comments
Language Name	This field displays the name of the language after you select the language code.

Field	Comments
Language Code	Select the language code. For example, select ENU for American English or ENG for British English.
Description	Enter the text of the recommendation message in the appropriate language.

Specifying Terms and Conditions of Product Promotions

This task is a step in *Process of Creating Product Promotions*.

Some promotions require customers to accept certain terms and conditions. In some industries, these are called commitments. If your promotion includes terms and conditions, then you must specify them.

For example, if a customer accepts a promotion offering a free cell phone to sign up for a certain mobile plan, then the customer may have to keep that mobile service for at least a year. If the customer cancels the service before the year has ended, then there may be a penalty.

When a customer disconnects an asset, the Siebel application checks to see whether the customer is breaking the terms and conditions of a promotion. If so, it displays a warning message telling the customer how much the penalty is. You can set up the product recommendation engine to recommend changes that will avoid the penalty, as described in the topic about recommendations in *Siebel Order Management Guide* . If there is a penalty, then the application calculates the amount. The Siebel application is integrated with the billing application, so the penalty is automatically applied.

You may want to prorate the penalty. For example, if customers get a free satellite dish when they sign up for twelve months of service, then they may have to pay the full penalty if they cancel before one month has elapsed, but pay only one-half of the penalty if they cancel after six months. For more information on proration, and defining conditional charge plans for agreement line items, see *Siebel Field Service Guide* .

A commitment may include recurring credit and cancellation charges. The cancellation charge and credits may need to be prorated for partial periods.

The penalty may be a nonrecurring charge paid once, or it may be a recurring charge paid periodically during a specified time period.

You can use the Commitments view to give customers ongoing benefits that they receive with a promotion, in addition to using it for penalties. For example, if the promotion gives the customer a credit every month for a year, which reduces the customer's monthly bill, then the Commitments view can record this credit.

Entering the Customer's Commitment

In the Charge Plan view of the Commitment applet, enter the charges for customers' ongoing commitment. Enter negative amounts if it is a credit. To do this, you must first create schedules and plans for any recurring, nonrecurring, and usage charges that the customer must pay. For more information, see the topic about contracts and agreements in *Siebel Field Service Guide* .

After you have created the necessary recurring and nonrecurring charge plans, you can enter the amounts that customers are committed to pay.

To enter the customer's commitment

1. In the Product Promotions, Commitment view, click Charge Plan in the link bar.
2. In the Charge Plan form, complete the necessary fields as described in the following table.

Field	Comments
NRC Schedule	Select a schedule for the nonrecurring charge. For example, a schedule may specify that the customer pays the recurring charge by the fifteenth of each month.
Nonrecurring	Enter the total amount of the nonrecurring charge.
NRC Plan	Select a nonrecurring charge plan. For example, the plan may allow the customer to pay off the nonrecurring charge in three monthly payments.
NRC Quantity	If the user does not select a value in the NRC Plan field, then the user can enter a specific number of payments. The charge calculation engine then divides the charge into equal payments of the quantity specified. If the user does select a value in the NRC Plan field, then this field displays the number of payments in the plan and is read-only.
RC Schedule	Select a schedule for the recurring charge. For example, a schedule may specify that the customer pays the recurring charge by the fifteenth of each month.
Recurring	Enter the amount of the recurring charges.
RC Adjust	Specify an adjustment for a recurring charge if the usage period is less than the entire billing period. For example, if the customer starts cable TV service on June 16, then the customer is charged for only 15 days in the month of June. In this case, enter the charge for the day in the RC Adjust field, and select Per Day in the RC Adjust U/M field. The application calculates the adjustment for the 15 days in June.
RC Adjust U/M	Enter the time period to which the RC adjustment applies. The options are Per Day and Per Week.
Adjust Charge	Enter a special adjustment to the amount the customer is charged. For example, if a customer calls to complain that cell phone service has not been working well for the last few days, then the company can agree to give the customer a one-time adjustment. The company enters the adjustment in this field.
Adjust Reason	Enter the reason for the special adjustment.

Field	Comments
Commit	Select this check box when you are done making changes to the record.
Charge Basis	Specify when the charge for a period is generated. The options are as follows: o **Schedule.** Charges are generated after the usage period. For example, June charges are generated on July 1. o **Advance.** Charges are generated in advance of the usage period. For example, June charges are generated on June 1.
Use Plan	Select the usage payment plan for this promotion. For example, there may be different usage payment plans for regular, gold, and platinum customers.
Usage Schedule	Select a schedule for the usage charge. For example, a schedule may specify that the customer pays the usage charge by the fifteenth of each month.

Entering Penalties for Canceling the Commitment

In the Charges and Credits view of the Commitment applet, enter the penalties that the customer must pay for canceling this commitment. You can also use this view to enter credits that the customer receives.

Before you enter penalties, you may have to create a plan used to prorate the penalty charge, if the penalty depends on the amount of time that the customer used the service before canceling. Create the prorate plan as a nonrecurring charge plan, described in the topic about contracts and agreements in *Siebel Field Service Guide* .

To enter the penalty that customers pay for canceling the commitment

1. In the Product Promotions, Commitment view, click Charges and Credits in the link bar.
2. In the Charges and Credits list, add a record for each penalty that the customer must pay, and complete the necessary fields as described in the following table.

Field	Comments
Type	Select the penalty.
Amount	Enter the amount of the penalty. If you enter a negative amount, then the customer will receive a credit instead of paying a penalty.
Plan	Select the nonrecurring charge plan used to prorate the penalty.

Entering Terms Displayed to the Customer

In the Terms view of the Commitment applet, enter the list of terms that will be displayed and included in the contract that the customer accepts.

The user can select a term template instead of entering terms manually, if you first define term templates using the Administration - Contract screen, Term Templates view. For more information about defining term templates, see *Siebel Life Sciences Guide* .

To enter terms displayed to the customer

1. In the Product Promotion, Commitment view, click Terms in the link bar.
2. For each term that will be displayed to the customer, do the following:
   - To select a term template, click Pick from Template.
   - To enter terms manually, add a record to the Terms list and complete the necessary fields as described in the following table.

Field	Comments
Number	Enter a number to control what order this text will be displayed in within the list of terms.
Type	Select the Type of the Term. The options are Standard, Non-Standard, and Special. When you design screens and reports, you may decide to display only terms of certain types.
Section	Enter the text that will be displayed to the customer.

Releasing the Product Promotion

This task is a step in *Process of Creating Product Promotions*.

When you created the new Product Promotion record, a version of it named Work Space was automatically created and the Locked check box was automatically selected in Product Promotions, Versions view.

When you have finished the entire process of defining the product promotion, you must release this version and refresh the cache to make the new version available to users.

Promotion versions work differently from product versions. Promotion versions only impact the Start Date, End Date, and Active fields of the promotion. Every other aspect of product promotions is uniform across all active versions of the promotion. The user must refresh the cache after making any changes to the promotion definition.

To release the product promotion

1. In the Product Promotions, Versions view, in the Work Space record, complete the necessary fields, as described in the following table.

Field	Comments
Required Start Date	Enter the date when the promotion becomes available.
Active	Select this check box to make the promotion available.

2. Click Release New Version.
3. From the Versions form menu, select Refresh Cache.

Creating Product Promotions That Upgrade Existing Promotions

You can design promotions that offer an upgrade to customers who have existing commitments because they have accepted earlier promotions.

For example, if customers have already signed up for a promotion to subscribe to a mobile telephone plan and have commitments to pay $29.95 a month for the next year, then you can create a promotion that offers them a mobile plan with no roaming charges for $39.95 a month. This involves breaking their existing commitment, but they do not have to pay the usual cancellation charge.

You can also offer the new promotion to customers who have accepted any one of many earlier promotions.

To create a promotion that is an upgrade to earlier promotions, define the new promotion as usual, and also use the Upgrade view to enter information about the existing promotions that customers can break without penalty.

To create a product promotion that upgrades existing promotions

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion.
3. Click the Upgrade view tab.
4. Add a record for each earlier promotion that can be broken without penalty and complete the necessary fields, as described in the following table.

Field	Comments
Original Promotion	Select the earlier promotion that customers can break without penalty.
Commitment Start	Specify when the commitment starts for the new promotion. The options are as follows: • **Original Start.** The commitment starts at the same time the commitment for the earlier promotion started. • **Now.** The commitment starts when the customer accepts the new promotion. • **Original End.** The commitment starts when the commitment for the earlier promotion ends.

Field	Comments
Duration	Specify the duration of the new promotion. The options are as follows: • **Original Duration.** The duration of the new promotion is the same as the duration of the original promotion. • **New Duration.** The duration of the new promotion is different from the duration of the original promotion. If you select this, then you must enter the new duration in the Commitments view.
Penalty	Enter the penalty for upgrading from the old to the new promotion, if any. **Note:** The penalty record will only be created if you set the Track as Agreement flag on the promotion definition record. For more information about agreements, see *Siebel Project and Resource Management Administration Guide*.
Prorate Plan	If there is a penalty for upgrading from the old to the new promotion, then select the prorate plan used to prorate this penalty.

Viewing a Summary of a Product Promotion

At any time, you can view a summary of the most important features of a product promotion in a single screen.

This is a read-only view. You use it to look at a summary of the product promotion while you are creating it, and also to provide summary information about the promotion to sales representatives.

To view a summary of a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the promotion.
3. Click the Summary view tab.

Copying a Bundled Promotion or Coupon

At any time, you can copy an existing bundle promotion or coupon, including child entities such as relationships, components and attributes. The only child entities that are not copied are Compatibility Rules.

To copy a product promotion or coupon

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the required bundled promotion or coupon.
3. In the Product Promotions list, click the menu button, and then click Copy Record.

   A duplicate bundled promotion or coupon record appears.
4. Enter a new name for the duplicate record.

Managing Versions of Product Promotions

You manage versions of product promotions using the Product Promotions, Versions view.

Managing versions of product promotions is not the same as managing versions of customizable products and product classes.

Versions are mainly used to schedule product promotions with different effective dates.

> **Note:** Apart from the effective date fields, changes to the header of the product promotion, the More Info view, and the other views are not versioned. These changes take effect when the user selects Refresh Cache from the menu of the Promotion Header form, or when the user logs out and logs in again.

Creating Product Promotion Versions with Different Effective Dates

If you are running a product promotion now and you want to run the same promotion a second time later in the year, then you can do this by creating promotion versions with different effective dates.

For example, if you are running a promotion during the month of April, 2005, and you want to run the same promotion again during the month of June, 2005, then complete the steps in the following procedure.

To create product promotion versions with different effective dates

1. Create the promotion with April 1, 2005 in the Required Start Date field and the Active check box selected. Release this version of the promotion.
2. Create a new version of the promotion with May 1, 2005 in the Required Start Date field and the Active check box deselected. Release this version of the promotion. Because Active is deselected, this version will not be visible to customers.
3. Create a new version of the promotion with June 1, 2005 in the Required Start Date field, and the Active check box selected to make the promotion visible to customers beginning June 1.
4. Create a new version of the promotion with July 1, 2005 in the Required Start Date field and the Active check box deselected, so the promotion is not visible to customers beginning July 1.

Viewing the Schedules of All Product Promotions

When you are planning new promotions, you can view a schedule of all existing product promotions to help you decide on the best effective dates for the new promotion. The Product Promotions Schedule view shows you a schedule of the effective dates for all product promotions.

To view the schedules of all product promotions

1. Navigate to the Administration - Product screen, then Product Promotions Schedule view.
2. Click 1, 7, or 30 to view the schedules for a day, week, or month.
3. From the Date drop-down, select the start date of the schedules you want to view, and click Go.

Red bars are displayed to show the dates that the promotions are active.

About Product Promotions for Customizable Products

If the end user selects a promotion with a special configuration, and then displays Siebel Configurator to configure the product, then the rules defined for a promotion are enforced within the Configurator session.

For example, a specific model of computer may have hundreds of possible configurations, but a promotional offer may allow only a few of these configurations. The promotion may require you to get a specific processor and either a 20 GB or 30 GB hard drive, even though many other possible hard drives are allowed by the configuration rules.

In this example, the user must display Configurator to choose the amount of RAM, the computer monitor, and other features besides the processor and hard drive. Configurator allows you to choose any one of the features that the configuration rules allow. However, Configurator lets you choose only the processor required by the promotion, and it lets you choose between only the 20 GB and 30 GB hard drive that the promotion allows, even though other processors and other hard drives may be allowed by the general configuration rules.

When you create a promotion for customizable products, you can define additional constraints beyond the constraints defined in Configurator. You can create additional constraints in the following ways:

- *Changing the Cardinality of Components of Product Promotions*
- *Constraining the Products Available for a Component of a Product Promotion*
- *Constraining the Attributes Available for a Product in a Product Promotion*
- *Defining Commitment for Customizable Product Relationships in Bundle Promotions*

Changing the Cardinality of Components of Product Promotions

You can change the cardinality of a component of a customizable product used in product promotions. For example, the product promotion can allow the user to select a smaller number of one component than the configurator constraints allow.

To change the cardinality of a component of a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the desired promotion.
3. Click the Products view tab.
4. In the Products link bar, click Components.
5. In the Components list, in Min or Max field for the component, enter the new cardinality constraints.

   The new Min and Max values have to fall within the cardinality constraints defined in the Customizable product definition, which are displayed in the Lower Limit and Upper Limit fields.

Constraining the Products Available for a Component of a Product Promotion

By default, a component of a customizable product in a product promotion has the products available that are defined in Configurator. However, you can modify the products that are available for the component so that the promotion offers only a subset of the products defined in Configurator.

For example, as defined in Configurator, a computer may be available with five different monitors. For the promotion, you may want to offer only two of those monitors.

To constrain the products available as a component of a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the desired promotion.
3. Click the Products view tab.
4. In the Products list, click the Product Name field of the customizable product.
   The structure of that product is displayed.
5. In the Relationships list, select the Component you want to change.
6. In the Relationship Domain list, click Modify.
   This deletes all the products in the Relationship Domain list.

   > **Note:** You must now manually add all the products that you want to make available for the component in the promotion, as described in the following step.

7. In the Relationship Domain list, click Add Item, and in the Domain Product field, select a product.
8. Repeat Step 7 until you have added all the products that are available for that component in the promotion.
9. In the Relationship Port field, modify the minimum and maximum cardinality as required.
   When you set Min Cardinality to 0, this means that the product is optional. If you set this value to 1, this means that the product is required. If the Min Cardinality setting is greater than 1, then this is the number of required products. If you set the Max Cardinality to 0, then the product cannot be added. If you don't want any product to be selected, you must update the Max Cardinality setting accordingly.

   > **Note:** Cardinality on the relationship is not enforced unless you modify the Min or Max Cardinality in the Relationships list.

Constraining the Attributes Available for a Product in a Product Promotion

If a product promotion includes a product with attributes, then you can constrain the domain of values allowed for an attribute. For example, if a shirt comes in ten different colors, then the promotion can be available for all the colors except red and blue.

You can create an exclude rule and list the attributes you do not want to make available, or you can create an include rule and list the attributes you do want to make available.

To constrain the attribute domain for a product in a product promotion

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the desired promotion.
3. Click the Products view tab.
4. In the Products list, click the Product Name field of the customizable product.

   The structure of that product is displayed.
5. Click the Attributes view tab.
6. In the Attributes list, go to the Condition field for the attribute whose domain you want to constrain, and select Exclude to create an exclude rule or select Include to create an include rule.
7. In the Values list:
   a. To create an exclude rule, add records for all the values of the attribute that you do not want to make available within the promotion.
   b. To create an include, add records for all the values of the attribute that you do want to make available within the promotion.

Defining Commitment for Customizable Product Relationships in Bundle Promotions

Use the following procedure to define commitment for customizable product relationships when using bundle promotions.

To define the commitment for customizable production relationships in bundle promotions

1. Navigate to the Administration - Product screen, then Product Promotions view.
2. In the Product Promotions list, select the desired promotion.
3. Click the Products view tab.
4. In the Products list, click the Product Name field of the customizable product.

   The structure of that product is displayed.
5. In the Relationships list, select the required relationship.
6. In the Commitments view, the fields as described in the following table.

Field	Comments
Commitment Period	Enter the number of periods to specify the duration of the customer's commitment. For example, if the selected unit of measurement in the Commitment UOM field is years, then you enter a value to correspond to the number of years the commitment is valid.
Commitment Period UOM	The unit of measure for the commitment time period, for example, years, months, days.
Grace Period	Enter a value for the period of time before a penalty is imposed on a customer. For example, if the selected unit of measurement in the Grace Period UOM field is days,

Field	Comments
	then you enter a value to correspond to the number of days before the customer is subject to a penalty.
Grace UOM	The unit of measure for the grace time period, for example, years, months, days.
Termination Charge	The amount that is charged if the component is terminated after end of grace period, but is terminated before the end of the commitment period.
Prorate Plan	This is calculated in the conditional charge plan. For more information, see *Siebel Field Service Guide* .
Apply Component Charge	Select this check box to apply an early termination penalty charge based on when the component is terminated and on the proration plan. If this check box is not selected, then the component penalty charges are not added.

Activating Workflows for Product Promotion

This task is a step in *Process of Creating Product Promotions*.

Product promotions are based on Siebel Workflows. You must activate these workflows before using the product. For information about activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

The following is a list of the workflows you need to activate:

- ISS Promotion Agreement Covered Assets Subprocess
- ISS Promotion Agreement Management Subprocess
- ISS Promotion Commitment Compliance Check Subprocess
- ISS Promotion Create Agreement Details
- ISS Promotion Disconnect Integration Subprocess
- ISS Promotion Disconnect Process
- ISS Promotion Disconnect Process - for Verify
- ISS Promotion Edit Integration Subprocess
- ISS Promotion Recommendation Subprocess
- ISS Promotion Upgrade Process
- ISS Promotion Verify Subprocess

9 Creating Product Offers

Creating Product Offers

This chapter covers product offers. It includes the following topics:

- *About Product Offers*
- *About Versions of Product Offers*
- *About Actions for Responses to Product Offers*
- *Process of Creating Product Offers*
- *Product Offers with Named Parameters*
- *Example of Creating Product Offers with Parent Actions*
- *Modifying Product Offers*
- *Product Offers for Multiple Products*
- *Viewing Product Offers for an Account*
- *Tasks for Suppressing Product Offer Recommendations*
- *Migrating Product Offers Using Workspace Projects*
- *Scenarios for Product Offers for Multiple Products Using Web Services*
- *Workflow Reference for Product Offers*

About Product Offers

A *product offer* is an offer to sell a product that allows a customer to respond with a single click. For example, you can add product offers to a Web site and display them in an area with the label Hot Offers. One hot offer might say: Do you want to upgrade to our extra value plan: 1,000 long distance minutes for $33.85 a month? This hot offer might have three buttons for the customer to click: Yes, No, and More Information. It is displayed only to customers who already have the standard plan for long-distance minutes. If the customer clicks Yes, then the order is placed automatically. If the customer clicks More Information, then an activity is created and assigned to a customer service representative, who contacts the customer.

To support one-click ordering, product offers have the following features:

- They enable you to define the possible user responses for a given offer.
- They enable you to specify ordering tasks that the Siebel application performs, such as:
  - Adding or deleting asset components, for example, changing a rate plan
  - Updating asset attributes, for example, changing bandwidth
  - Upgrading a promotion, for example, upgrading from the promotion named double play to the promotion named triple play
  - Editing or applying a promotion, for example, changing handset

    - Updating line item fields, for example, changing the value of a manual discount

- They enable you to specify preordering tasks and postordering tasks that the Siebel application performs, such as address validation and order validation.

Product Offers on Low-Interactivity Channels

Product offers are particularly useful for channels that are unassisted or have limited user interactivity, such as short messaging service (SMS), Interactive Voice Response (IVR), and the hot offers bin on a Web site. When the user accepts the order, some exceptions might occur. You must configure exception handling, which is typically channel-specific:

- **Web.** For this channel, display the exception on the Web site.
- **SMS.** For this channel, create an activity and assign it to a customer service representative.
- **IVR.** For this channel, route the exception to a customer service representative.

Because product offers are also available in the Siebel application, they provide customers with a consistent experience on all sales channels. For example, a customer might view the offer on the Web and phone the call center. Then the customer service representative at the call center can view the same product offer in the Siebel application and accept the offer on behalf of the customer. For more information on how customer service representatives can access product offers, see *Viewing Product Offers for an Account*.

Product Offers and Promotions

Both product offers and Siebel promotions give customers the opportunity to buy products or bundles of products at a promotional price. The main differences are:

- Siebel promotions enable customers to buy a single product or a group of products and services at a promotional price. Each product or service within a promotion can have its own configurable options, contractual commitments, and early termination fees.
- Siebel product offers enable customers to buy products or services (including Siebel promotions) through nonassisted channels with limited, user interactivity. Product offers enable one-click ordering. Permitted user responses are clearly defined, as are the actions that the Siebel application performs for each user response. Customers are typically prequalified for product offers.

Upgrading a promotion often requires user interactivity to resolve product selections. Users typically must select the products or services that they want to upgrade or disconnect. Product offers do not support this level of user interactivity, but they can be used to give customers a quick start for creating orders when they are upgrading promotions.

About Versions of Product Offers

Product offer versions work differently from product versions. Product offer versions affect only the Start Date, End Date, and Active fields of the product. Every other aspect of a product offer is uniform for all active versions of the product offer. The user must refresh the cache after making any changes to the product offer definition, using the Clear Cache or Refresh Cache menu option.

These menu options are available for all the views that show versioned objects in the Administration - Product Offer screen: Versions view, Response-Actions view, Recommendations view, Compatibility view, Eligibility view, and Product Line Compatibility view.

If you choose the Clear Cache option, then the application refreshes the cache for all the child objects of the product offer, such as recommendations, compatibility rules, eligibility rules, and so on. The application calls the RefreshBC method of the Row Set Transformation Toolkit.

About Actions for Responses to Product Offers

When you define user responses to product offers, you can also define actions for each response.

The actions create sales orders or quotes, which can be used to:

- Create a new asset.
- Modify an existing asset by doing one of the following:
  - Add, replace, or delete products and services for an asset
  - Update asset line item fields
  - Upgrade products and services for an asset
  - Edit or apply a promotion to products and services in the asset

You can also perform other actions by invoking a Siebel workflow process.

> **Note:** Actions that invoke workflows can be used for preorder tasks and postorder tasks, but they cannot be interspersed among actions that create orders. If an action invokes a workflow process, then create the workflow before you define responses. For more information about creating workflows, see *Siebel Business Process Framework: Workflow Guide* .

You generally create actions that map to common order processes such as:

- **Modify order.** Creates actions modifying an order, adding or deleting products, and providing manual discounts
- **Apply promotion.** Creates actions that include bundled products with promotional pricing

For example, you have a product offer for customers who have pay-as-you-go text messaging that says: Get unlimited text messaging for $10 per month. For the response Accept, you create the following actions:

- To specify the product, create actions that modify the asset Mobile Service so that it has the component product Unlimited Text Messaging.
- To specify the pricing, create an action that updates the Discount Amount field of the component product Unlimited Text Messaging, so it has the value of 10.

This is one example of how product and pricing can be specified using the actions of product offers. Different product offers are specified in different ways.

The Product Administrator has the flexibility to set up offers that validate the customer's assets prior to applying the actions:

- Actions to add a product that the customer already owns can be skipped.

- Actions to remove a product that the customer does not own can be skipped.

The same product can be added and removed within a single offer. For example, an offer named Replace SIM could create an order that removes the customer's old SIM card and adds a new SIM card.

Some responses do not require actions. For example, if the user rejects the offer, then an action is not required usually. In that case, you do not have to define the actions for the Reject response.

You specify an action by selecting in both the Action Type field and the Action Code field. The selection in the Action Type field limits the available selections in the Action Code field. Depending on your selections in these two fields, you might have to specify a value in another field. For example:

- If you select Modify as the Action Type and Update as the Action Code, then you must also make entries in the Field Name and Field Value fields. The action updates that field with that field value.
- If you select Run Process as the Action and Workflow as the Action Code, then you must also select a workflow in the Service Name field. The action runs that workflow.

The following table shows the actions that can be performed by product offers, with the Action Type and Action Code of each. This table includes actions that you specify in the Action record, with no user input. You can also use named parameters to base some of these values on user input, rather than specifying them explicitly in the Action record. For more information, see *Product Offers with Named Parameters*.

Action Type	Action Code	Description
New	Add	Generates an order or quote to create a new asset. Specify the product by selecting it in the Base Product field.
	Update	Modify a field or attribute value of a new asset. To do this requires two actions: • An action with the action type New and the action code Add to add the new asset • An action with the action type New and the action code Update to modify the new asset
Modify	Add	Modifies an existing asset by adding a product or service. Specify the product by selecting it in the Base Product field, and specify the product or service you are adding to it by selecting it in the Component Product field.
	Update	Modifies an existing asset by updating a product or service. Specify the product by selecting it in the Base Product field. Specify the product or service you are updating by selecting it in the Component Product field. Specify how the product or service is being updated by entering values in the Field Name and Field Value fields, or in the Attribute Name and Attribute Value fields
	Delete	Modifies an existing asset by deleting a product or service. Specify the product by selecting it in the Base Product field. Specify the product or service you are deleting by selecting it in the Component Product field.
	Replace	Modifies an existing asset by replacing a product or service that is an existing component with a new component under the same relationship.

Action Type	Action Code	Description
		Specify the product by selecting it in the Base Product field. Specify the product or service with which you are replacing an existing component of the asset by selecting it in the Component Product field. The replace action deletes the item that was previously selected and adds the item that was specified in the actions. If there is no existing component, then it simply adds the new item. **Note:** The Replace action applies to relationships with a maximum cardinality of 1. If more than one item is selected in the asset, then the replace action randomly deletes one of the selected items.
Upgrade	Upgrade	Upgrades products and services for an existing asset. Specify the Upgrade From product in the Base Product field, and specify the Upgrade To product in the Upgrade Product field.
Run Process	Workflow	Runs a workflow. Select the workflow in the Workflow Name field.

Process of Creating Product Offers

In addition to the tasks for defining product offers in this process, you must create the Web site and possibly other channels that present the offer to customers, and you must integrate them with the Siebel application.

To define a product offer, perform the following tasks:

- *Creating Product Offers*
- *Creating Responses for Product Offers*
- *Creating Actions for Responses to Product Offers*
- *Creating Eligibility and Compatibility Rules for Product Offers*
- *Creating Product Offer Recommendations*
- *Creating Translations for Product Offer Messages*
- *Releasing Product Offers*

Creating Product Offers

The first step in defining a product offer is to create a Product Offer record, which includes general information about the product offer. The view tabs under the Product Offer record hold the detailed definition of the product offer.

This task is a step in *Process of Creating Product Offers*.

To create a product offer

1. Navigate to the Administration - Product screen, then Product Offers view.

2. Add a new record to the Product Offers list, and complete the necessary fields in the new record and in the More Info form, as described in the following table.

Field	Comments
Name	Enter a name for the product offer.
Product Offer ID	This field displays a unique ID for the product offer. Siebel CRM generates the ID.
Organization	Select one or more organizations to enable users to see this product offer. Users who are not in these organizations cannot see this product offer in the Siebel application.
Description	Enter a description of the product offer.
Effective Dates	This field displays the start date and end date of the period when this product offer is effective. Users can enter or edit these dates in the Versions view.
Message Type	The default value for the Unified Message Framework (UMF) message type is Product Offer Recommendation. This message type is part of the seed data and contains the responses Accept and Reject. If you want to create new responses, such as Enroll and Decline, then you must create a new message type, such as Loyalty Recommendation, using the Unified Message Framework, and associate it with the Product Offer Message Group. For more information about the Unified Messaging Framework, see *Siebel Order Management Infrastructure Guide* .
Message	Enter a message for the product offer.
Thumbnail	Select the small thumbnail version of the image associated with the product offer.
Image	Select a larger version of the image associated with the product offer.
Check Eligibility	Select this check box to make the Siebel application check customers' eligibility to get this product offer. If you select this check box, then you can define rules in the Eligibility and Compatibility Rules view for the product offer.
Inclusive Eligibility	Select this check box to specify inclusive eligibility. If this check box is not selected, then the Siebel application uses exclusive eligibility.

Field	Comments
	With exclusive eligibility, customers are eligible to buy the product if they are not excluded by the eligibility or compatibility rule. With inclusive eligibility, customers are eligible to buy the product only if they are explicitly included by the eligibility rule.

3. From the More Info menu, select Clear Cache.

   For more information about refreshing the cache, see *About Versions of Product Offers*.

Creating Responses for Product Offers

Because product offers accept a one-click response, you must specify the different responses offered to the user. Examples of responses are:

- For the offer, Get 500 points when you enroll in online billing, the responses might be **Enroll** and **Decline**.
- For the offer, Upgrade your handset to Daytech 500, the responses might be **Accept**, **Decline** and **Learn More**.

If you keep the default value (Product Offer Recommendation) in the Message Type field of the Product Offer record, then the default responses, Accept and Reject, are displayed in the Responses list. You do not have to change them.

This task is a step in *Process of Creating Product Offers*.

To create responses to product offers

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Responses view tab.
4. Add a new record to the Responses list for each response that will be displayed to the user, and complete the necessary fields as described in the following table.

Field	Comments
Sequence	Enter the order in which the responses are displayed to users.
Response	Select the response. The options can be the following: o The default values: Accept and Reject. o The messages that you created in the Unified Messaging Framework and associated with the Message Type of this product offer in *Creating Product Offers*.
Comments	Enter comments for your own use.
Inactive	Select this check box to make this response inactive, so it is not displayed as a possible response for the product offer. To use this field, you must select Columns Displayed from the menu.

Creating Actions for Responses to Product Offers

For each response that you created, you can define one or more actions that the Siebel application performs if the user selects that response. For more information about actions, see *About Actions for Responses to Product Offers*.

In most cases, you define the action using only fields in the Action record, but there are two cases where you can define more complex actions:

- **Workflow parameters.** If the action runs a workflow, then you can also specify parameters that are passed to the workflow. For more information, see *Creating Parameters for Actions That Invoke Workflows*.
- **Named parameters.** Rather than specifying certain values in the Action record, you can use named parameters to enable the user to select from a range of values. For more information, see *Product Offers with Named Parameters*.

This task is a step in *Process of Creating Product Offers*.

To create actions for a response to a product offer

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Responses view tab.
4. Select the response that you want.
5. Add one or more new records to the Responses - Actions list, and complete the necessary fields as described in the following table.

Field	Comments
Sequence	Displays the order in which the actions are executed. The sequence number is generated automatically, and you cannot change it. **Note:** The sequence number is important when there is more than one action for the same product. For example, because a product must exist before you can change its attributes, the sequence number for an action that adds a product must be lower than the sequence number for an action that updates an attribute for that product.
Action Name	When you step off the action record, the Action Name field is automatically populated with a unique number. You can change it to give it a meaningful name that helps you to identify the action.
Action Type	Select the type of action. The options are as follows: o **New.** Select this value to create a new sales order (if the customer is residential) or quote (if the customer is not residential). o **Modify.** Select this value to modify an existing asset by adding, updating or deleting products and services. o **Upgrade.** Select this value to upgrade products and services for an asset. Also, select this value to upgrade an existing asset.

Field	Comments
	○ **RunProcess.** Select this value to run preorder or postorder tasks by invoking a workflow.
Action Code	Select an action code to specify the action. The options are limited depending on your selection in the Action field. The options include: ○ **Add.** Select this code to add new products and services for an asset. ○ **Update.** Select this code to modify products and services for an asset. ○ **Replace.** Select this code to delete the currently selected item and add the new product for a specific product path. **Note:** The Replace action code applies to product paths that have a maximum cardinality of 1. You cannot have more than one product selected in the product path. ○ **Delete.** Select this code to delete obsolete products and services for an asset. ○ **Upgrade.** Select this code to upgrade products and services for an asset. ○ **Workflow.** Select this code to invoke a workflow.
Base Product	Select a base product.
Component Product	You can enter a component in the Component Product field if you select an action code of Add, Replace, Update, or Delete and if the base product has components. You cannot enter a component in this field when the action code is Upgrade or Workflow.
User Input	Select this check box if the action requires user input. This field is used only with named parameters. For more information, see *Product Offers with Named Parameters*.
Workflow Name	If you selected workflow as the action code, then use this field to select the workflow process. If you invoke a workflow, then you can also pass parameters to this workflow if necessary. For more information, see *Creating Parameters for Actions That Invoke Workflows*.
Field Name and Field Value	If the action code is Update, then you can select a field name and field value to update the field with the specified value. If you select a component product, then only the attributes and fields applicable to that component product are available for selection. If you do not select a component product, then only the fields applicable to the base product are available for selection.
Attribute Name and Attribute Value	If the action code is Update, then you can select an attribute name and attribute value to update the attribute with the specified value. If you select a component product, then only the attributes and fields applicable to that component product are available for selection. If you do not select a component

Field	Comments
	product, then only the attributes and fields applicable to the base product are available for selection.
Upgrade Product	If the action code is Upgrade, then select the product that you want to upgrade.
Named Parameter	If you are using a named parameter, then select one of the following options to specify what type of object the named parameter applies to: o Base Product o Component Product o Upgrade Product o Attribute Value o Field Value Do not enter a value in the equivalent fields of the Action record if you are specifying those values using named parameters. For example: o Do not enter a value in the Base Product field when you select Base Product in the Named Parameter field. o Do not enter a value in the Attribute Value field when you select Attribute Value in the Named Parameter field. You can either enter these values directly in the fields of the Action record, or you can specify these values using named parameters. For more information about how to specify values using named parameters, see *Product Offers with Named Parameters*.
Parent Action	Enter a parent action when the user input for this action is dependent on a user selection for an earlier action. This field can be used with named parameters. For more information, see *Product Offers with Named Parameters*. This field can also be used for other purposes. For more information, see *Example of Creating Product Offers with Parent Actions*.
Allow Multiple Instance	This check box specifies the number of instances of a product within a quote or order. By default, the check box is selected and it allows multiple line items with the same product. To allow only one instance of a product in a quote or order, deselect the check box. For example, if you were creating an offer that adds the International Calling Package to the customer's existing home phone service, you would deselect the Allow Multiple Instance check box, because a customer would never need two or more International Calling Packages.

Creating Parameters for Actions That Invoke Workflows

If you defined an action for the product offer that invokes a workflow, then you can define parameters that are passed to the workflow.

For example, you have a number of product offers that require customers to have a minimum, prepaid balance, with different balances required for the different product offers. You create a workflow that checks an external application to determine the customer's prepaid balance and that generates an order only if the customer has more than the required balance. Different product offers pass different parameters to the workflow, depending on the minimum balance that the offer requires. When you create a new product offer that requires a minimum balance of $50, you call this workflow, and enter 50 as the workflow parameter.

For this product offer, you would enter the name of this workflow that checks minimum balance in the Workflow field, and you would enter 50 in the Value field in the following procedure. For a different product offer that requires a minimum balance of $200, you would enter the name of this workflow in the Workflow field, and you would enter 200 in the Value field.

If you want to provide inputs from the current context, such as an order ID generated by earlier actions or custom fields on the account header, then you must extend the Default Product Offer Recommendation variable maps unless you have configured custom variable maps

To specify parameters for actions that invoke workflows

1. Create the action, as described earlier in this topic, with the values shown in the following table.

Field	Value
Action	RunProcess
Action Code	Workflow
Workflow	The name of the workflow that is invoked

2. Click the Workflow Parameter view tab.
3. Add one or more new records to the Workflow Parameter list, and complete the necessary fields as described in the following table.

Field	Comments
Name	Enter a workflow parameter name.
Value	Enter the workflow parameter value.

Creating Eligibility and Compatibility Rules for Product Offers

You can define eligibility rules and compatibility rules for product offers. For more information, see the topics about eligibility rules and compatibility rules in *Siebel Product Administration Guide* .

An *eligibility rule* is a rule that restricts product offers to only some of your customers. For example, a product offer might be offered only to customers in certain states.

A *compatibility rule* is a rule that specifies other products or services with which product offers can and cannot be used. For example, the product offer, Free Unlimited Text Messages, requires customers to have the plan, Pay-As-You-Go Text Messages.

This task is a step in *Process of Creating Product Offers*.

To create eligibility and compatibility rules for product offers

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. In the More Info view tab, make sure that the Check Eligibility check box is selected and that the Inclusive Eligibility check box is selected if you want to create inclusive rules or that it is deselected if you want to create exclusive rules.
4. Click the Eligibility and Compatibility view tab.
5. Display the appropriate view as follows:
    - If you are checking eligibility, then click the Eligibility link.
    - If you are checking compatibility, then click the Compatibility link.
    - If you are checking compatibility based on product line, then click the Product Line Compatibility link.
6. Define the eligibility and compatibility rule in the same way that you define these rules for products.

    For more information, see *Siebel Product Administration Guide* .

Creating Product Offer Recommendations

You can define upsell and cross-sell recommendations for product offers. You can recommend the product offer to customers who have a specific installed asset or to customers who add a specific item to a quote or order.

You can define translations of the recommendations so they are displayed in the customer service representative's language. For information about how the customer service representative can view the recommendations, see *Viewing Product Offers for an Account*.

For example, you are creating a product offer for unlimited text messaging for $10 for a month. You want to present this product offer as an upsell recommendation to mobile service customers who already have the pay-as-you-go text message plan. To accomplish this, you do the following:

- You create a recommendation when you define this product offer, and you select Mobile Service as the subject product. This product offer is the default value of the object product.
- You set up a compatibility rule that states that this product offer requires customers to have the pay-as-you-go text message plan.

When customer service representatives view the installed assets of mobile services customers, they can select Recommendations from the menu to display this product offer. The compatibility rule ensures that the product offer is displayed only for customers who have the pay-as-you-go text message plan. For more information about recommendations, see the chapter about product recommendations in *Siebel Order Management Guide* .

This task is a step in *Process of Creating Product Offers*.

To create product offer recommendations

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Recommendations view tab.
4. Add a new record to the Recommendations list, and complete the necessary fields as described in the following table.

Field	Comments
Subject Product	Select the product that the customer must own or be interested in buying to get this recommendation.
Object Product	This field displays the product offer, which is the upsell or cross-sell product recommended to the customer.
Reason	Enter the message to be displayed in the Messages list. The message is used by customer service representatives to tell customers the reason that they should purchase the upsell or cross-sell product.
Effective From and Effective To	Enter the start date and end date of the period when this recommendation is active.

5. To specify translations for the text message that you entered in the Reason field, do the following:
   a. Select the record that you want to translate in the Recommendations list.
   b. In the Translations list that is after the Product Recommendations list, add a new record for each translation, and complete the necessary fields as described in the following table.

Field	Comments
Language	Enter the language code for the translation. For example, if this record contains the French translation of the message text, then enter the code FRA.
Reason	Enter the translation of the message text. This translation is displayed in implementations in the language that you specified. For example, if the company has a Siebel implementation or a Web site in French, then this French translation is displayed in that implementation.

Creating Translations for Product Offer Messages

You can use the Messaging view to enter translations for the product offer description and message that you entered in the product offer header.

> **Note:** In addition to translating the product offer description and message, you can also translate the names of the products using the Administration Products screen, Translations view. For more information, see *Siebel Product Administration Guide* .

The translations are displayed on both Web sites and in Siebel applications, depending on the language of each channel.

This task is a step in *Process of Creating Product Offers*.

To create translations for product offer messages

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Messaging view tab.
4. For each language in which the description is displayed, add a new record to the Messages list, and complete the necessary fields as described in the following table.

Field	Comments
Language Name	After you select the language code, this field displays the name of the language.
Language Code	Select the language code. For example, select ENU for U.S. English or ENG for British English
Description	Enter the translation of the Description field.
Message	Enter the translation of the Message field.

Releasing Product Offers

When you created the new Product Offer record, a version of it named Work Space was automatically created, and the Locked check box was automatically selected in Product Offer, Versions view. When you have finished the entire process of defining the product offer, you must release this version and refresh the cache to make the new version available to users. For more information about refreshing the cache, see *About Versions of Product Offers*.

This task is a step in *Process of Creating Product Offers*.

To release product offers

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Versions view tab.
4. If you have not cleared the cache since modifying information in this view, from the Versions menu, select Refresh Cache.

   For more information about refreshing the cache, see *About Versions of Product Offers*.
5. Click Release New Version.
6. Click the More Info view tab.

Product Offers with Named Parameters

Named parameters enable you to create a generic product offer to serve as a template for a number of similar offers. The named parameter is a variable that can take a number of different values, based on the user's input during run time. After accepting the product offer, the user selects an option from a list of possible values.

A product offer can include only one action that takes user input in the Siebel application. Although product offers are aimed at unassisted channels, named parameters provide some user interactivity: they enable a two-click order instead of a one-click order. Only one named parameter is allowed for a product offer to limit user interactivity. After the user accepts the product offer, the user is allowed to select from only one more prompt.

This limited user interactivity is allowed to make it easier to set up product offers. The administrator can create one product offer that uses a named parameter to let the user choose from 20 options, instead of having to create 20 different product offers that are almost identical.

The following topics cover product offers with named parameters:

- *Scenario for Setting Up Product Offers with Named Parameters*
- *Creating Product Offers with Named Parameters*

Scenario for Setting Up Product Offers with Named Parameters

This scenario shows one way that you can set up product offers using named parameters. You might use named parameters differently, depending on your business model.

A telecommunications company wants to set up a product offer that allows customers to get a 20% discount on all of its Short Messaging Service (SMS) options, 50 messages, 200 messages, 500 messages, or unlimited messages. Rather than setting up a separate product offer for each option, the company sets up one product offer with named parameters, which allows the customer to choose an option. The product offer says: Get 20% off SMS. If the customer clicks Accept, then the product offer displays a list that gives the customer the choice of selecting 50, 200, 500, or unlimited messages.

If you do not have named parameters, then you have to create four different offers, all with the same set of actions:

- Action Type is set to Modify.
- Action Code is set to Add.
- Base Product is set to Mobile Service.
- Component Product is set to 50 SMS, 200 SMS, 500 SMS, or Unlimited SMS.

Using named parameters, you create one product offer, with the following actions:

- Action Type is set to Modify.
- Action Code is set to Add.
- Base Product is set to Mobile Service.
- Component Product. (Leave this field blank.)
- Named Parameter is set to Component Product.

In this case, all of the options for SMS are set up as possible component products of the base product that is used to provide customers with their telecommunications package.

If the customer accepts the product offer, then the Siebel application displays a list SMS options for the user to choose from. You define the domain of the named parameter to specify which items are in this list. For more information on named parameters, see *Creating Product Offers with Named Parameters*

Creating Product Offers with Named Parameters

To create product offers with named parameters, create the product offers in the usual way, described in *Process of Creating Product Offers*, and enter the values described in the following procedure.

The following procedure shows you how to define a named parameter where the Siebel application gets the user input. You select the User Input check box, and you use the named parameter to present the user with a list of choices. If you are using some other application as the user interface that receives input, then you do not have to use these fields.

To create product offers with named parameters

1. When you create the action, as described in *Creating Actions for Responses to Product Offers*, enter the values described in the following table in the Action record.

Field	Comments
User Input	Select this check box to specify that the value of the named parameter is determined by the user's input.
Named Parameter	Select an option to specify which value will be determined using a named parameter. The options are as follows: ○ **Base Product.** The named parameter sets the value of the base product in the product offer. ○ **Component Product.** If you select Component Product in the Named Parameter field, then you must also enter a value in the Base Product field to specify which base product has a component product set by the named parameter. ○ **Upgrade Product.** The named parameter sets the value of the upgrade product. ○ **Attribute Value.** If you select Attribute Value in the Named Parameter field, then you must also enter a value in the Attribute Name field to specify which attribute has its value set by this named parameter. ○ **Field Value.** If you select Field Value in the Named Parameter field, then you must also enter a value in the Field Name field to specify which field has its value set by this named parameter.

Field	Comments
Parent Action	Enter a parent action if the input for this action depends on a user selection for an earlier action. For example, a product offer allows the customer to get a 20% discount on any accessory for a particular brand of smartphone. You must create two actions: o The first action adds the accessory selected by the user to the order. The named parameter is the base product. The user is prompted to select any product in the product line that includes all accessories for this smartphone. o The second action applies a 20% discount to the product that the user selected in the first action. You must select the first action as the parent action for the second action, so the Siebel application can determine that it applies the discount to the product that was selected in the first action.

2. Click the Named Parameters view tab.

   The domain of the named parameter is the list of values that the user can choose from. You can specify a component domain, which allows you to define the products in the domain by explicitly selecting all of the individual products, attribute values, or field values that are available.
3. Add a record to the Component Domain list for each allowed value in the domain, and complete the necessary fields as described in the following table.

Field	Comments
Sequence	Enter the sequence in which this value is listed when the allowed values are displayed to the user.
Base Product/ Component Product/ Upgrade Product/ Attribute Value/ Field Value.	Enter an allowed value for this parameter. The name of this field is dynamic and depends on what you select in the Named Parameter field of the Action record. For example, if you select Base Product in the Named Parameter field of the Action record, then this field displays the field name Base Product, indicating that you enter one allowed value for the base product in this field.

When the user accepts the product offer, the Siebel application displays a list of all the products in the domain that you defined, so that the user can choose one product.

Example of Creating Product Offers with Parent Actions

This topic gives one example of using parent actions in product offers. You might use this feature differently, depending on your business model.

The Parent Action field is in the Action record. For more information, see *Creating Actions for Responses to Product Offers*.

You use the Parent Action field when you create products offers with named parameters. For more information, see *Creating Product Offers with Named Parameters*.

You can also use the Parent Action field for other purposes. This example describes one possible use of parent actions for other purposes.

You have a product offer that gives 50% discount on an accessory, a product named Bluetooth Headset. The first action adds this product, and the second action gives a 50% discount to the product added in the first action. The second action has the first action as its parent action, because it uses a value from the first action.

To create this product offer with a parent action, create the product offers in the usual way, described in *Process of Creating Product Offers*, and when you add the Action records, enter the values described in the following procedure.

To create actions for a response to a product offer

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Responses view tab.
4. Select the response that you want.
5. To add the headset to the order, add a record to the Responses - Actions list, and complete the necessary fields with the values shown in the following table.

Field	Value
Sequence	1
Action Name	Add Headset
Action Type	New
Action Code	Add
Base Product	Bluetooth Headset

6. To give a 50% discount for the product that was added in the first action, add a second record to the Responses - Actions list, and complete the necessary fields with the values shown in the following table.

Field	Value
Sequence	2
Action Name	Give 50% Discount
Action Type	New
Action Code	Update

Field	Value
Base Product	no value
Field Name	Discount Percent
Field Value	50
Parent Action	Add Headset

Modifying Product Offers

Because product offers are versioned, you modify them by modifying the Workspace version in the Versions view. Whenever you release a product offer, the product offer that you release is given a version number, and a new version named Work Space is automatically created.

Versions of product offers are different from versions of products and promotions because only the Start Date, End Date, and Active fields of the product offer can be different in different versions. Apart from these three fields, every other field of a product offer is uniform for all versions of the product offer. You must refresh the cache after making any changes to the product offer definition, so these fields remain uniform. For more information, see *About Versions of Product Offers*.

Note: If you change the data in any other field except the Start Date, End Date, and Active Field, then the change populates all of the older versions of the product offer as well as the new version.

To modify product offers

1. Navigate to the Administration - Product screen, then Product Offers view.
2. Select the Product Offer that you want.
3. Click the Versions view tab.
4. Make sure that the Work Space version is selected.
5. Select the Locked Flag check box.
6. Modify the Work Space version of the product offer by clicking the other view tabs and modifying the information.

   For more information about the information in each view, see *Process of Creating Product Offers*.
7. When it is time for customer service representatives and customers to view the modified product offer, click Release New Version.
8. From the Versions menu, select Clear Cache.

Product Offers for Multiple Products

Product offers allow you to create a single offer for a single product. You can also use product offers to create quotes or orders for multiple products by:

- Creating a product offer for each of the products, as described earlier in this chapter.
- Chaining together these product offers using the Apply Multiple Product Offers web service. For background information about how to use this web service, see *Scenarios for Product Offers for Multiple Products Using Web Services*. For technical information about this web services, see *Siebel CRM Web Services Reference* .

Typically, you use the Apply Multiple Product Offers web service to create orders for residential customers and to create quotes for business customers.

Viewing Product Offers for an Account

Customer service representatives can view all product offers for an account's installed assets.

This procedure displays all upsell or cross-sell recommendations for the account's installed assets that are either promotions or product offers.

Note: Product offers are not displayed in the Catalog or Product Pick applet of the Siebel application.

To view product offers for an account

1. Navigate to the Accounts screen, then the Accounts List view.
2. In the accounts list, drill down on the name of the Account that you want.

   The Account Summary view appears, with the Installed Assets list and the Billing Items list on either side.
3. In the menu for the Installed Assets list, select Recommendations.

   A dialog box appears containing a list of all recommendations for this account.
4. In the dialog box, select recommendations in the list, and click the buttons to accept or reject the product offer.

Tasks for Suppressing Product Offer Recommendations

The following topics describe how to suppress product offer recommendations:

- *Suppressing Product Offer Recommendations that Have Been Accepted or Rejected*. You must suppress product offer recommendations so that product offers are not displayed after customers have already accepted or rejected them.
- *Suppressing Component Product Recommendations*. You can suppress product offer recommendations for component products to improve performance.

Suppressing Product Offer Recommendations that Have Been Accepted or Rejected

You suppress recommendations so that the same recommendation is not shown to the same customer or for the same asset after it has already been accepted or rejected.

For example, a customer has two mobile service subscriptions, one for himself and one for his wife. There is a product offer giving a 20% discount on any SMS to mobile service subscribers. This customer is presented with two offers, one for each of the mobile subscriptions. The customer is not interested in the SMS option for his wife, so he rejects the offer. The next time the customer calls, the offer for his wife is not displayed because it has already been rejected, but the offer for him is displayed. The customer accepts the offer for himself, and this offer will not be shown again because it has already been accepted.

This procedure assumes you are using the default Product Offer Recommendation message type and default responses. If you have created other message types or responses, you can turn suppression on and off in the same way.

To suppress recommendations that have been accepted or rejected

1. Navigate to the Administration - Order Management screen, then the Message Types view.
2. In the Message Types list, query for Product Offer Recommendation.
3. Click the Responses view tab.
4. For the Accept and Reject responses, select the Suppress Repetition check box to turn suppression on, or deselect this check box to turn suppression off.

Suppressing Component Product Recommendations

Product offer recommendation can be set at either the component product or root product level. By default, the product displays all recommendations for the assets of a given account. To improve performance, you might want to show recommendations for only the root products, not component products.

To suppress component product recommendations, you set a search specification for the variable map that controls the recommendation, which is Default Product Offer Recommendation Variable Map - Row Set, unless you have configured product offers by creating custom variable maps.

To configure product offers to suppress recommendations

1. Navigate to the Administration - Order Management screen, then the Variable Maps view.
2. In the Variable Maps list, query for the variable map you want.

   If you are using the default variable maps shipped with the product, query for Default Product Offer Recommendation Variable Map - Row Set
3. Drill down on the Work Space version.
4. Lock the Work Space version.
5. Click the Modes view tab.
6. In the row for the mode Any, in the SearchSpec field, to filter out the recommendations that you want to suppress, enter the search specification

```
[Asset Id]=[Root Asset Id] .
```

7. Release the Work Space version, and refresh the cache.

Migrating Product Offers Using Workspace Projects

You can migrate product offers from one Siebel CRM environment to another by using workspace projects.

You can export just the product offer header or the full structure of the product offer, including all its child objects. If you export the full structure, then all the objects that the product offer refers to must exist in the target environment, including products, promotions, and eligibility and compatibility matrixes. For example, if you migrate a product offer that includes the product named Mobile Service and the component product named SMS Unlimited, then the product offer import fails if the products Mobile Service and SMS do not exist in the target environment.

To migrate product offers using workspace projects

1. Export the product offer from the source environment:
    a. Navigate to the Administration - Product screen, then the Workspace Projects view.
    b. In the Workspace Projects list, add a new record and complete the necessary fields as described in the following table.

Field	Comments
Name	Enter a name for the workspace project.
Effective Start Date	Enter the date that the application uses to determine what versions must be exported for dependent items. When you import a workspace, this field is also used to set the start date of the versions released from the workspace project.

    c. Click the Contents view tab.
    d. In the Contents list, click Add to add a new record.
    e. In the Select Versioned Object dialog box, select the product offer that you want.
    f. Continue to add new records and to select new product offers until you have selected all the product offers you want to migrate.
    g. From the Workspace Project menu, select Export Contents.
    h. In the Export Contents dialog box, choose one of the export options and click OK. Options are as follows:
        - **Object(s) only export.** Exports the product offer header information only.
        - **Full structure export.** Exports the product offer header and all of its child objects, including Responses, Actions, Recommendations and Messages. Full structure does not include product or promotion definitions.
    i. Use the dialog box to save the contents in an XML file in a location that you can access from the target environment.
2. Import the product offer to the target environment:

a. Navigate to the Administration - Product screen, then the Workspace Projects view.
b. In the Workspace Projects list, add a new record and complete the necessary fields as described in the following table.

Field	Comments
Name	Enter a name for the workspace project.
Effective Start Date	Enter the date that is used to set the Start Date of the versions released from the workspace project.

c. From the Workspace Project menu, select Import Contents.
d. In the dialog box, select the XML file with the content you want, and click on OK

   The product offers from the XML file appear in the Contents view, and are all locked.
e. Click Release All.

Scenarios for Product Offers for Multiple Products Using Web Services

This topic describes how a telecommunications company might set up product offers for multiple products. You might use product offers differently, depending on your business model.

Before you use any of the scenarios in this topic, the telecommunication company must have an eCommerce web site that can present the products and product offers that are defined in Siebel CRM application. Here is a simplified description of how residential customers could use this Web site to place an order:

- First, the user launches the telecommunication company's web site and clicks on the link to shop for residential products and services.
- The user enters the address where the service would be installed.
- The eCommerce Web site displays a list of services available in this location. In this example, the user selects the triple play bundle, which includes telephone, internet and television. Each of these options is defined as a product offer.
- The eCommerce Web site displays a list of add-ons based on the user's selection. For example, because the selected package includes television and home phone service, the user can select add-ons including Basic Movie Package, Sports package, and International calling package. Each of these specific add-on options is defined as a product offer, and users select the combination they want.
- After finishing choosing specific options, the user clicks the Review Order button. At this point, the eCommerce application invokes the Apply Multiple Offers to Create Order web service to generate the complete order detail along with the final price. The order is generated in memory and is not saved to the Siebel database.
- If user is satisfied with this order, the user clicks the Place Order button. At this point, the eCommerce application invokes the Order web service to save the order to the Siebel application.

This topic includes the following scenarios for using web services related to product offers for multiple products:

- *Scenario for a New Customer to Create a Quick Quote*
- *Scenario for a New Customer to Add Items to an Existing Quote*
- *Scenario for Upsell Offers*
- *Scenario for Changing Service Plan*
- *Scenario for Build Your Own Bundle*

Scenario for a New Customer to Create a Quick Quote

In this scenario, a new customer gets a quick quote for a service bundle.

1. A new customer browses the eCommerce Web site of a telecommunications provider. The customer enters his home zip code and views a list of available services. The customer decides to get a quick quote for a new promotion offered by the provider, called the Medium Triple Play Bundle, which includes Internet +Home Phone + Basic TV Bundle for $120 /month.
2. The eCommerce web application calls a Siebel web service to return a list of eligible product offers for the customer. The customer selects the following offers:
    - Two add-ons for TV:
        - Basic Movie package
        - Spanish Combo channel
    - One add on for Internet:
        - Online Security
3. To generate a quote, the eCommerce web application calls the Siebel web service ApplyMultipleProductOffers.

    The web service requires the list of product offer ids along with the sequence in which they are selected. The web service has the option to save the quote in Siebel CRM or generate the quote in memory. Because this is an in-memory Quick Quote, you specify that the quote is not saved by populating the SaveFlag with N in the input message.

    To view an example of this web service, navigate to the documentation library for Siebel applications on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html), and click the Related Files link next to the link for Siebel CRM Web Services Reference to access a zip file. Extract the files in the zip file, and see the following files to view the sample code:
    - ...\Product Offer-Apply Multiple Product Offers\New Acquisition - InMemory - Request Message.txt
    - ...\Product Offer-Apply Multiple Product Offers\New Acquisition - InMemory - Response Message.txt
4. The web service output message is a document generated in memory with the structure shown in the following table.

Sequence	Product	Promotion	Action Code
1.0	Medium Triple Play Bundle		Add
2.0	Internet Service	Medium Triple Play Bundle	Add

Sequence	Product	Promotion	Action Code
2.1	Fibe 100GB		Add
2.2	Online Security		Add
3.0	Voice Service	Medium Triple Play Bundle	Add
3.1	Flat National		Add
4.0	TV Service	Medium Triple Play Bundle	Add
4.1	Essential Plan		Add
4.2	Basic Movie Package		Add
4.3	Spanish Combo Package		Add
5.0	Internet Router	Medium Triple Play Bundle	Add

Note: The newly generated order applies the default promotion constraints. Eligibility & Compatibility rules and Product Configurator constraints are enforced using the Configurator BatchValidate API. Pricing rules are enforced by invoking the Siebel Pricing workflows.

Scenario for a New Customer to Add Items to an Existing Quote

In this scenario, the customer reviews the quote described in *Scenario for a New Customer to Create a Quick Quote* and selects the premium set-top box instead of the default option for basic set-top box.

1. To generate a quote, the eCommerce web application calls the Siebel web service ApplyMultipleProductOffers. Because we are updating an existing quote, you specify the quote structure in the input message. The web service applies the offers to the existing quote instead of generating a new quote.

   To view an example of this web service, navigate to the documentation library for Siebel applications on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html), and click the Related Files link next to the link for Siebel CRM Web Services Reference to access a zip file. Extract the files in the zip file. See the following files to view the sample code:

   - ...\Product Offer-Apply Multiple Product Offers\New Acquisition - Add Equipment - Request Message.txt
   - ...\Product Offer-Apply Multiple Product Offers\New Acquisition - Add Equipment - Response Message.txt

2. The web service output message is a document generated in memory with the structure shown in the following table.

Sequence	Product	Promotion	Action Code
1.0	Medium Triple Play Bundle		Add
2.0	Internet Service	Medium Triple Play Bundle	Add
2.1	Fibe 100GB		Add
2.2	Online Security		Add
3.0	Voice Service	Medium Triple Play Bundle	Add
3.1	Flat National		Add
4.0	TV Service	Medium Triple Play Bundle	Add
4.1	Essential Plan		Add
4.2	Basic Movie Package		Add
4.3	Spanish Combo Package		Add
5.0	Internet Router	Medium Triple Play Bundle	Add
6.0	Premium Set-Top box	Medium Triple Play Bundle	Add

Note: A line item has been added to the order for the Premium Set-Top Box.

3. The customer can complete customizing the order by selecting more services, but in this case, the customer just clicks the Place Order button.
4. The eCommerce web application calls the Siebel ApplyMultipleProductOffertoCreateOrder web service with SaveFlag set to Y.

Scenario for Upsell Offers

An existing customer has TV service with the Basic Movie package. This customer logs in to the telecommunication provider's e-Commerce site to pay his bill.

1. The telecommunication provider's e-Commerce Web site displays upsell offers that can be added to the customer's existing TV service, and the customer selects the following options:
    - Premium Movie Package
    - On-Demand TV Service
2. To generate a quote, the eCommerce web application calls the Siebel web service ApplyMultipleProductOffers. Because we are modifying the customer's existing service, you specify the asset-id in the input message.

    To view an example of this web service, navigate to the documentation library for Siebel applications on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html), and click the Related Files link next to the link for Siebel CRM Web Services Reference to access a zip file. Extract the files in the zip file. See the following files to view the sample code:

    - ...\Product Offer-Apply Multiple Product Offers\Existing Customer - Upsell Add-ons - Request Message.txt
    - ...\Product Offer-Apply Multiple Product Offers\ Existing Customer - Upsell Add-ons - Response Message.txt
3. The web service output message is a document generated in memory with the structure shown in the following table.

Sequence	Product	Promotion	Action Code
1.0	TV Service	Medium Triple Play Bundle	-
1.1	Essential Plan		-
1.2	Spanish Combo Package		-
1.3	Basic Movie Package		Delete
1.4	Premium Movie Package		Add
1.5	On-Demand TV		Add

Note: Basic Movie Package is replaced with Premium Movie Package

Alternative Use of the Web Service in This Scenario

Alternatively, the ApplyMultipleProductOffers Web service supports the document id in the input message. The document id is either an order or quote id that exists in the Siebel database. For example:

1. The first web service call adds Premium Movie Package and saves the order to the Siebel database.
2. The second web service call updates the order in Siebel with On-Demand TV service.

Scenario for Changing Service Plan

An existing customer has the Medium Double-Play Bundle, which includes Internet and Voice services. She logs in to the telecommunication provider's e-Commerce site to add TV service.

1. The customer places an order for the Medium Triple-Play Bundle, which includes Internet, Voice, and TV services.
2. To generate a quote, the eCommerce web application calls the Siebel web service ApplyMultipleProductOffers. Because we are changing from a Double Play bundle to a Triple Play bundle, you specify the asset-id of the Double Plan bundle in the input message. Because this is an in-memory quote, you specify that the order is not saved by populating the SaveFlag with N in the input message.

   To view an example of this web service, navigate to the documentation library for Siebel applications on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html), and click the Related Files link next to the link for Siebel CRM Web Services Reference to access a zip file. Extract the files in the zip file. See the following files to view the sample code:
   - ...\Product Offer-Apply Multiple Product Offers\ Change Plan - InMemory Upgrade - Request Message.txt
   - ...\Product Offer-Apply Multiple Product Offers\ Change Plan - InMemory Upgrade- Response Message.txt
3. The web service output message is a document generated in memory with the structure shown in the following table.

Sequence	Product	Promotion	Action Code
1.0	Medium Double Play Bundle		Delete
2.0	Medium Triple Play Bundle		Add
3.0	Internet Service	Medium Triple Play Bundle	Update
4.0	Digital Voice Service	Medium Triple Play Bundle	Update
5.0	Internet Router	Medium Triple Play Bundle	Update
6.0	TV Service	Medium Triple Play Bundle	Add

Sequence	Product	Promotion	Action Code
7.0	Premium Set-top box	Medium Triple Play Bundle	Add

Note: Existing Internet and Voice Services are transferred to the new Triple Play Bundle.

Scenario for Build Your Own Bundle

A new customer browses the e-Commerce site of a telecommunications provider and finds that the bundles of services offered on the site do not meet her requirements. She decides to create her own bundle of services by selecting individual options.

1. The customer selects Build Your Own Bundle and chooses the following options:
    - Internet - 500 mbps (highest speed)
    - VoIP - Per minute call
    - Basic Home Security package.
2. To generate a quote, the eCommerce web application calls the Siebel web service ApplyMultipleProductOffers. Because this is an in-memory Quick Quote, you specify that the quote is not saved by populating the SaveFlag with N in the input message.

    To view an example of this web service, navigate to the documentation library for Siebel applications on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html), and click the Related Files link next to the link for Siebel CRM Web Services Reference to access a zip file. Extract the files in the zip file. See the following files to view the sample code:
    - ...\Product Offer-Apply Multiple Product Offers\ BuildYourOwnBundle - Request Message.txt
    - ...\Product Offer-Apply Multiple Product Offers\ BuildYourOwnBundle - Response Message.txt
3. The web service output message is a document generated in memory with the structure shown in the following table.

Sequence	Product	Promotion	Action Code
1.0	Build Your Own Bundle		Add
2.0	Internet Service	Build Your Own Bundle	Add
2.1	Fibe 500GB		Add
3.0	Voice Service	Build Your Own Bundle	Add
3.1	Per Minute Plan		Add

Sequence	Product	Promotion	Action Code
4.0	Home Security	Build Your Own Bundle	Add
4.1	Basic Package		Add

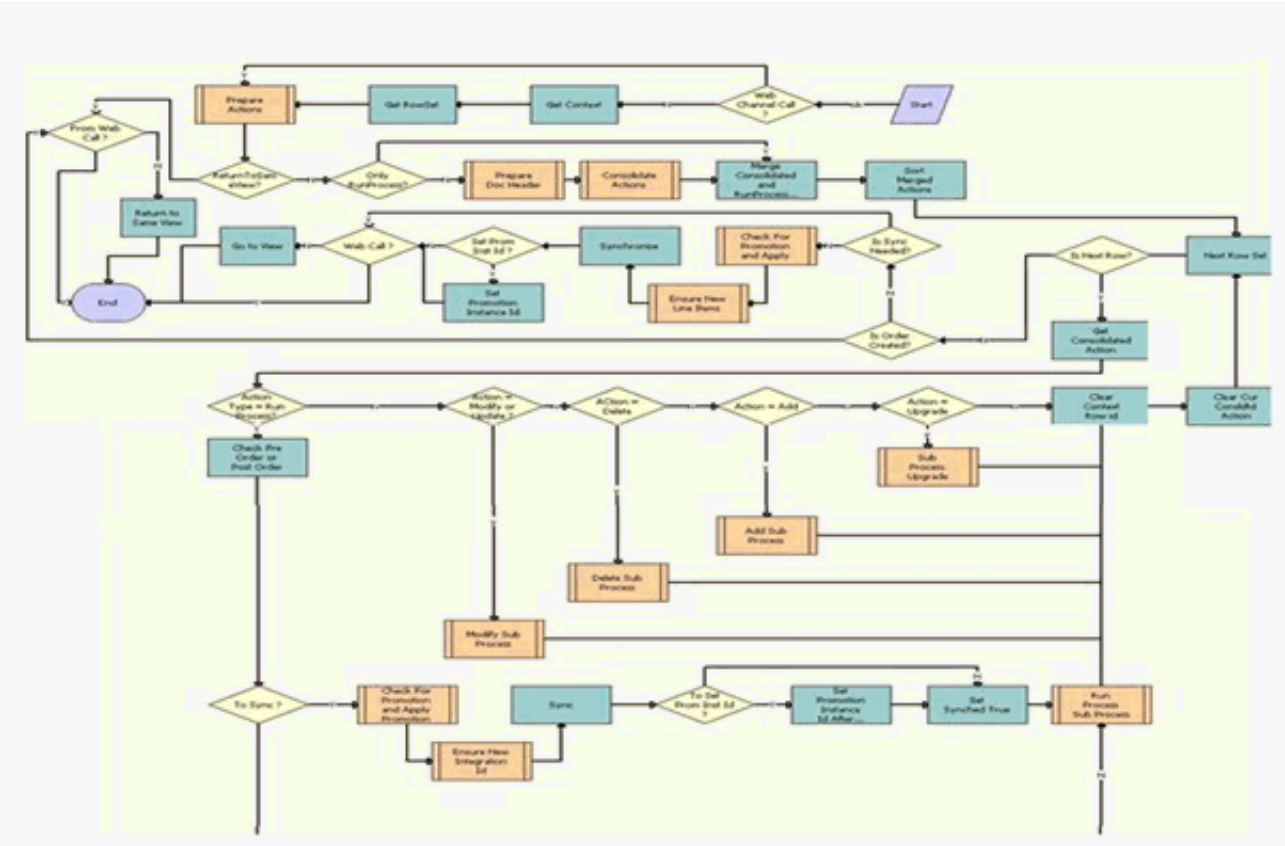
Workflow Reference for Product Offers

Product offers use the following workflows:

- *RAF Create Document Workflow*
- *RAF Update Recommendation Workflow*
- *RAF Update Recommendation SubProcess Workflow*
- *RAF Submit Order Process Workflow*
- *RAF Create Activity Workflow*
- *RAF Product Offer Recommendation For Asset Workflow*
- *RAF Recommendation Driver Workflow*
- *RAF Prepare Header Sub Process Workflow*
- *RAF Prepare Actions Sub Process Workflow*
- *RAF Check Prepaid Balance Workflow*
- *RAF Get Active Document Workflow*
- *RAF Consolidate Actions Sub Process Workflow*
- *RAF Check For Promotion Sub Process Workflow*
- *RAF Ensure New Line Item Workflow*
- *RAF RunProcess Driver Workflow*
- *RAF Get Named Parameter Workflow*
- *RAF Get Named Parameter For Web Call Workflow*
- *RAF Apply Offer To Create Order Web Workflow*
- *RAF Apply Offer To Create Quote Web Service Workflow*
- *Product Offer GetProductOffer Workflow*
- *Product Offer GetProductOfferStructure Workflow*
- *Product Offer GetRecommendedProductOffers Workflow*
- *RAF Apply Multiple Offer To Create Order Web Service Workflow*
- *RAF Apply Multiple Offer To Create Quote Web Service Workflow*

RAF Create Document Workflow

RAF Create Document Workflow, shown in the following figure, is a wrapper workflow to carry out all actions in the product offer. It is invoked through the signal RAF Create Offer Document when the user accepts or rejects a product offer in the recommendation window. It is also invoked called from the Web service flows for processing product offers. This workflow accepts an order or quote structure as input. If the input argument ProcessInMemory=N, it synchronizes the quote or order generated to the database; if ProcessInMemory=Y, it returns the quote or order as an output argument.



Workflow Description

This workflow performs the following actions:

- **Web Channel Call?** This decision step routes the flow to the Prepare Actions step if the process property Web Channel Call is Y or to the Get Context step if it is not Y. If the workflow is invoked through the Web channel, the context and rowset information are prepopulated, and the flow can go directly to the Prepare Actions step because it is not necessary to get the context or the rowset.
- **Get Context.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Context to get the account header information.
- **Get Rowset.** This step calls the method GetRowSetData of Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Row Set to get all asset information for the account.

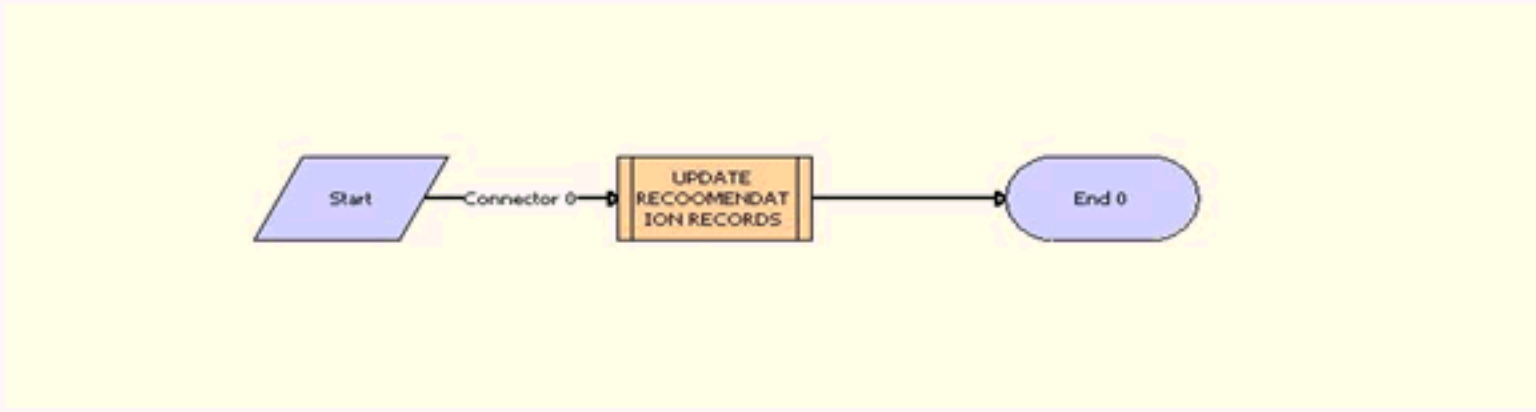
- **Prepare Actions.** This step calls the subprocess RAF Prepare Actions Sub Process which queries for all the actions defined for the particular product offer and sorts them into different property sets depending on Action type. Outputs of this step are property sets such as Modify Action Set, New Action Set, RunProcess Action Set and so on.
- **Return to same view.** This decision step determines if the user should be navigated to a different view or remain in the same view. If the product offer has no actions specified, then the Return to same view flag is set to Y. If this Flag is Y the flow goes to business service step Return to Same View. If not, the flow goes to the step Only RunProcess?
- **Only Run Process?** This decision step routes flow to Merge Consolidated and RunProcess Actions if all the actions defined for Product Offer are of Type Run Process. If not, it routes the flow to Prepare Doc Header.
- **Prepare Doc Header.** This step calls the subprocess RAF Prepare Header Sub Process, which creates an Order or Quote header depending on account type and active doc Id.
- **Consolidate Actions.** This step calls the subprocess RAF Consolidate Actions Sub Process, which consolidates all actions based on Action type (other than Run Process) and groups them under one Base product.
- **Merge Consolidated and RunProcess Actions.** This step merges the consolidated property set from Consolidate Actions and Run Process property set.
- **Sort Merged Actions.** This business service step calls the method SortRowSet of business service RAF Product Offers Service which sorts the merged action property set according to sequence.
- **Next Row Set.** This step picks the next rowset from context property set.
- **Is Next Row?** This decision step routes the flow to the step Get Consolidated Action if Cur Row Id is not Null. Otherwise, it routes the flow to the step Is Order Created.
- **Get Consolidated Action.** This step gets the current consolidated actions and, depending on account type, builds an order or quote context rowset.
- **Action Type = Run Process?** This decision step routes the flow to Check Pre Order or Post Order if the Action Set Type is Run Process. Otherwise, it routes the flow to Action = Modify or Update?
- **Action = Modify or Update?** This decision step routes the flow to Modify Sub Process if the Action Code is - or Modified. Otherwise, it routes the flow to Action = Delete?
- **Action = Delete.** This decision step routes the flow to Delete Sub Process if the Action Code is Deleted. Otherwise, it routes the flow to Action = Add.
- **Action = Add.** This decision step routes the flow to Add Sub Process if the Action Code is New. Otherwise, it routes the flow to Action = Upgrade.
- **Action = Upgrade.** This decision step routes the flow to Sub Process Upgrade if the Action Code is Upgrade. Otherwise, it routes the flow to Clear Context Row id.
- **Check Pre Order or Post Order.** This step checks whether Run Process is the first action to be processed and whether or not an order has been created.
- **To Sync?** This decision step routes the flow to Check For Promotion and Apply Promotion if the response is not reject and the order has not been synchronized already. Otherwise, it routes the flow to Run Process Sub Process.
- **Check For Promotion and Apply Promotion.** This step calls the subprocess RAF Check For Promotion Sub Process which checks whether the Promotion Name is correct and does Apply, Disconnect, Modify or Upgrade promotion.
- **Ensure New Integration Id.** This step calls the subprocess RAF Ensure New Line Item, which ensures that all line items have unique integration ids.
- **Sync.** This step calls the business service method Synchronize of the ABO Bulk Request Processing Service, which synchronizes order and quote line items in the database.

- **To Set Prom Inst Id?** This decision step routes the flow to Set Promotion Instance Id After Sync if promotion has been added, deleted, modified or disconnected. Otherwise, it routes the flow to Set Synched True.
- **Set Promotion Instance Id After Sync.** This step stamps the promotion name and promotion instance Id on line items modified because of the promotion.
- **Set Synched True.** This step sets the Synched flag to Y.
- **Run Process Sub Process.** This step calls the subprocess RAF RunProcess Driver Workflow, which fetches workflow parameters and executes the specified workflow.
- **Modify Sub Process.** This step calls the subprocess ABO Bulk Request - Modify Sub-Process Workflow to carry out the actions in the product offer of the action type Modify. If a quote or order structure is part of the input, it passes the structure to the subprocess for consideration when the offer is being processed; if the Is RAF flag is set to Y (identifying this as a call from a product offer, from either the UI or a Web service, rather than a call from a bulk request) the quote or order structure is taken into consideration during order generation. Instance information (quote or order structure) is only passed when this is called from a product flow, which is when RAF is set to TRUE, and not when it is passed from a bulk request flow.
- **Delete Sub Process.** This step calls the subprocess ABO Bulk Request - Delete Sub-Process Workflow to carry out actions in the product offer of the action code Delete. Instance information is not passed, as it is in bulk request flows. The Is RAF flag is passed to handle exceptions.
- **Add Sub Process.** This step calls the subprocess ABO Bulk Request - Add Sub-Process Workflow to carry out actions in the product offer of the action type New. If a quote or order structure is part of the input, it passes the structure to the subprocess for consideration when the offer is being processed; if the Is RAF flag is set to TRUE, the quote or order structure is taken into consideration during order generation. Instance information (quote or order structure) is only passed when this is called from a product flow, which is when RAF is set to TRUE, and not when it is passed from a bulk request flow.
- **Sub Process Upgrade.** This step calls the subprocess ABO Bulk Request - Upgrade Sub-Process Workflow to carry out actions in the product offer of the action type Upgrade. If a quote or order structure is part of the input, it passes the structure to the subprocess for consideration when the offer is being processed; if the Is RAF flag is set to TRUE, the quote or order structure is taken into consideration during order generation.
- **Clear Context Row id.** This step clears the details of already processed actions from the context header.
- **Clear Cur Consldtd Action.** This step clears the details of already processed actions from the context rowset.
- **Is Order Created?** This decision step routes the flow to Is Sync Needed? if an order is created. Otherwise, it routes the flow to Return to same view.
- **Is Sync Needed?** This decision step routes the flow to Web Call? if the synched flag is Y. Otherwise, it routes the flow to Check For Promotion and Apply.
- **Check For Promotion and Apply.** This step calls the subprocess RAF Check For Promotion Sub Process, which checks whether the promotion name is correct and does Apply, Disconnect, Modify or Upgrade promotion. If the workflow is called from an offer flow, the argument Instance contains an order or quote header to be used by the subprocess.
- **Ensure New Line Items.** This step calls the subprocess RAF Ensure New Line Item, which ensures that all Line items have unique integration ids.
- **Synchronize.** This step calls the business service method Synchronize of ABO Bulk Request Processing Service. Depending on the value of the input argument ProcessInMemory, this business service either synchronizes quote or order line items to the database or returns the quote or order without synchronizing to the database for in-memory processing of a product offer.
- **Set Prom Inst Id?** This step checks whether the line item is added, deleted, modified because of the promotion. If so, it routes the flow to Set Promotion Instance Id.
- **Set Promotion Instance Id.** This step stamps the promotion name and promotion instance Id on line items modified because of the promotion.

- **Web Call?** This decision step routes the flow to End if it is a Web service call. Otherwise, it routes the flow to Go to View.
- **Go to View.** This step navigates the user to Order Entry - Line Items View (Sales) or Order Entry - Network Line Items Detail View (Sales) or Quote Item Detail View or Quote Network Detail View, depending on the account type and the field Compound Product Number.
- **Return to Same View.** This step navigates the user back to SIS OM Customer Account Portal view.

RAF Update Recommendation Workflow

RAF Update Recommendation Workflow, shown in the following figure, is invoked when the user updates the Message Type field of the product offer header. It updates the Message Type field in the Recommendation business component to the same value.



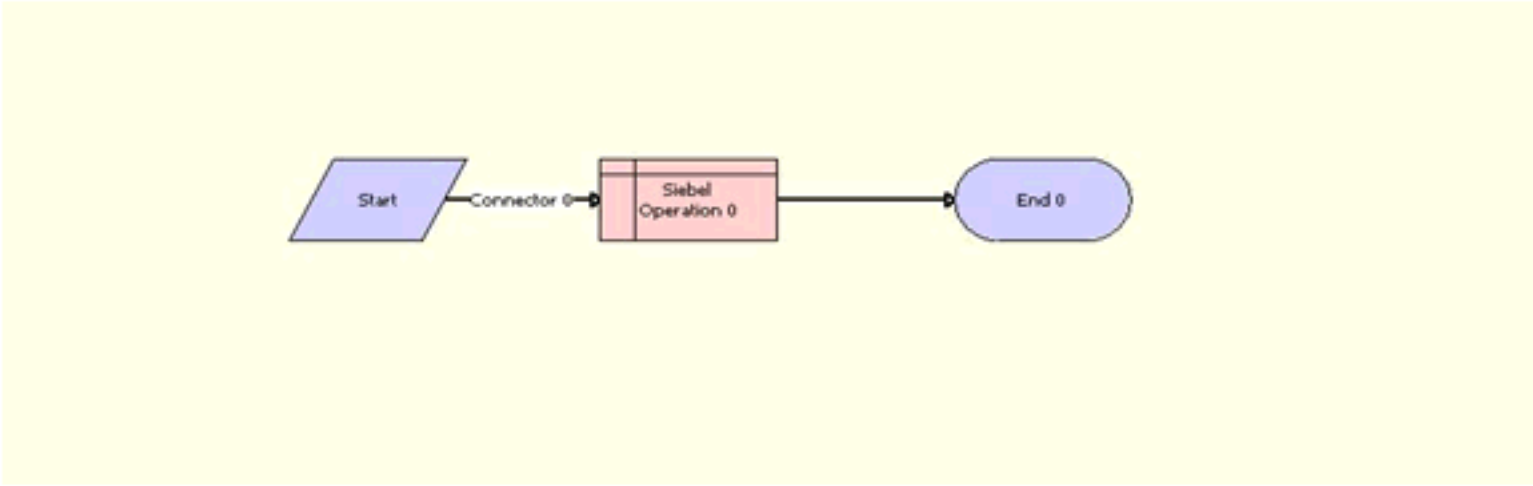
Workflow Description

This workflow performs the following action:

- **UPDATE RECOOMENDATION RECORDS.** This step calls the subprocess RAF Update Recommendation SubProcess Workflow for all the recommendation records in the Product Offer.

RAF Update Recommendation SubProcess Workflow

RAF Update Recommendation SubProcess Workflow, shown in the following figure, is called as a subprocess by the parent workflow RAF Update Recommendation Workflow. It updates the Message Type field of all records in Recommendation business component to match that of Product Header Message Type.



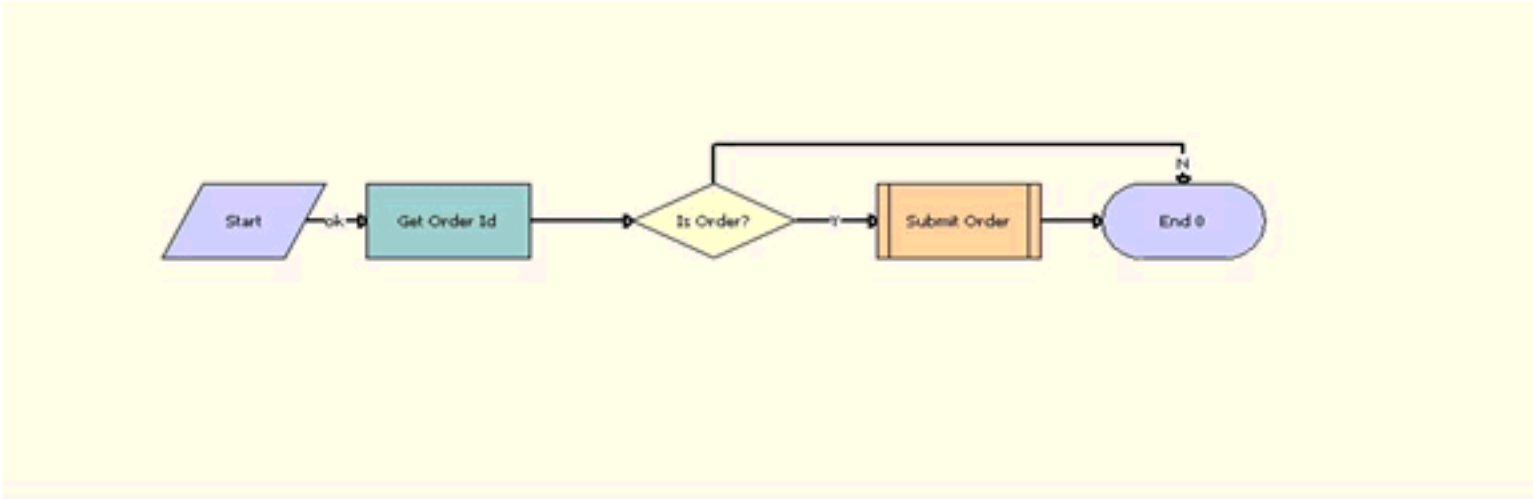
Workflow Description

This workflow performs the following action:

- **Siebel Operation:0**. This step performs an update operation on Recommendation records, updating the UMS Type Id and UMS Type fields.

RAF Submit Order Process Workflow

RAF Submit Order Process Workflow, shown in the following figure, a wrapper workflow called when the action type is Run Process and the workflow name is SIS OM Submit Order Process.



Workflow Description

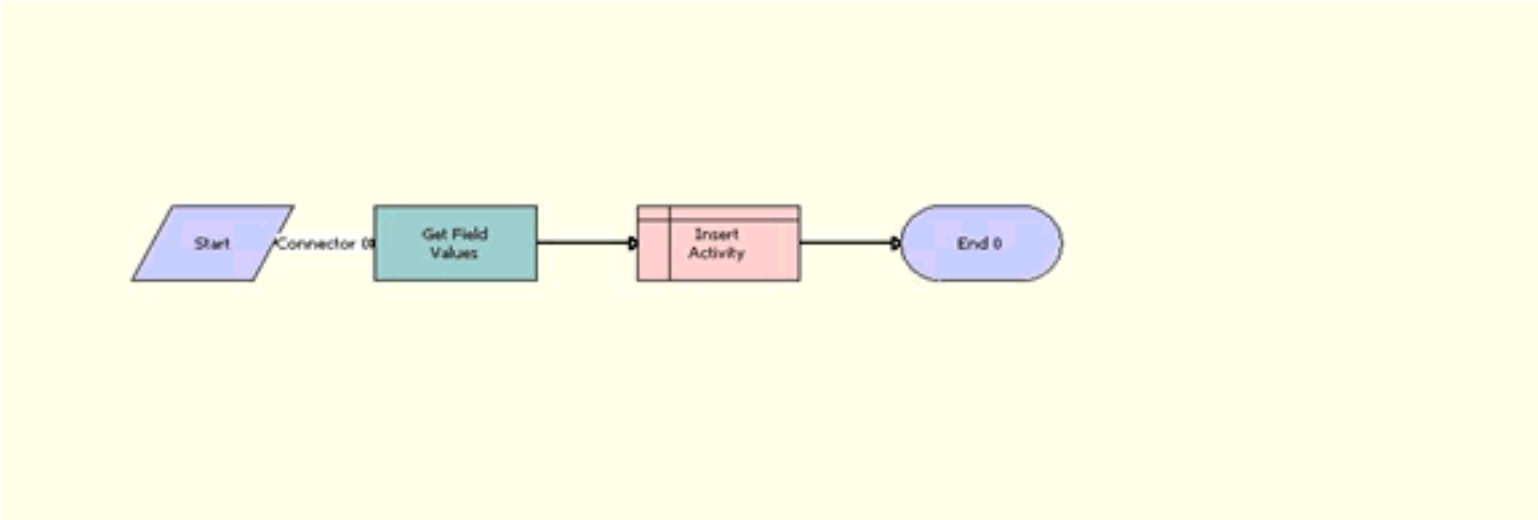
This workflow performs the following actions:

- **Get Order Id.** This step calls the method GetFieldvalue of the business service CMU External Integration Service to get the active order Id for which submit order needs to be called.

- **Is Order.** This decision step routes flow to Submit Order if the Account Type is Residential. Otherwise, it routes the flow to End.
- **Submit Order.** This step calls the workflow SIS OM Submit Order Process, passing the order Id.

RAF Create Activity Workflow

RAF Create Activity Workflow, shown in the following figure, creates a record in activity applet which has details of the rejected product offer. It is called if the action is the type Run Process, and the workflow name is RAF Create Activity.



Workflow Description

This workflow performs the following actions:

- **Get Field Values.** This step calls the method GetFieldValue of the business service CMU External Integration Service to get values for the fields Activity Description, Activity Type and Account Id from workflow parameters defined in Product Offer.
- **Insert Activity.** This step inserts a record in the Action business component with the details from previous step.

RAF Product Offer Recommendation For Asset Workflow

RAF Product Offer Recommendation For Asset Workflow, shown in the following figure, is called when the user clicks the Recommendation menu item for assets in SIS OM Customer Account Portal view.



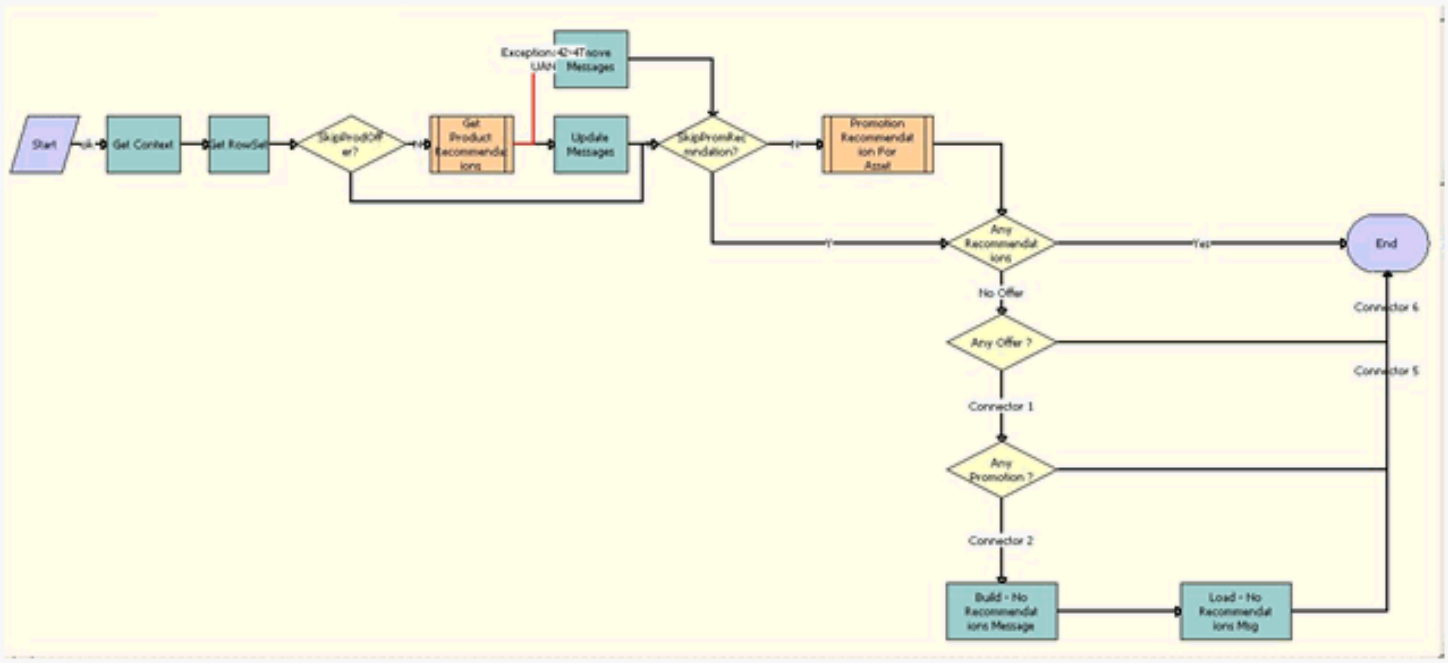
Workflow Description

This workflow performs the following action:

- **Get Recommended Product Offers For Assets.** This step calls the method RAFRaiseSignal of the business service RAF Product Offers Service, which raises the signal Product Recommendation Signal, which in turn calls RAF Recommendation Driver Workflow or Product Recommendation Driver Workflow, depending on instance type.

RAF Recommendation Driver Workflow

RAF Recommendation Driver Workflow, shown in the following is called by the Product Recommendation signal when the instance type is AssetLineItem.



Workflow Description

This workflow performs the following actions:

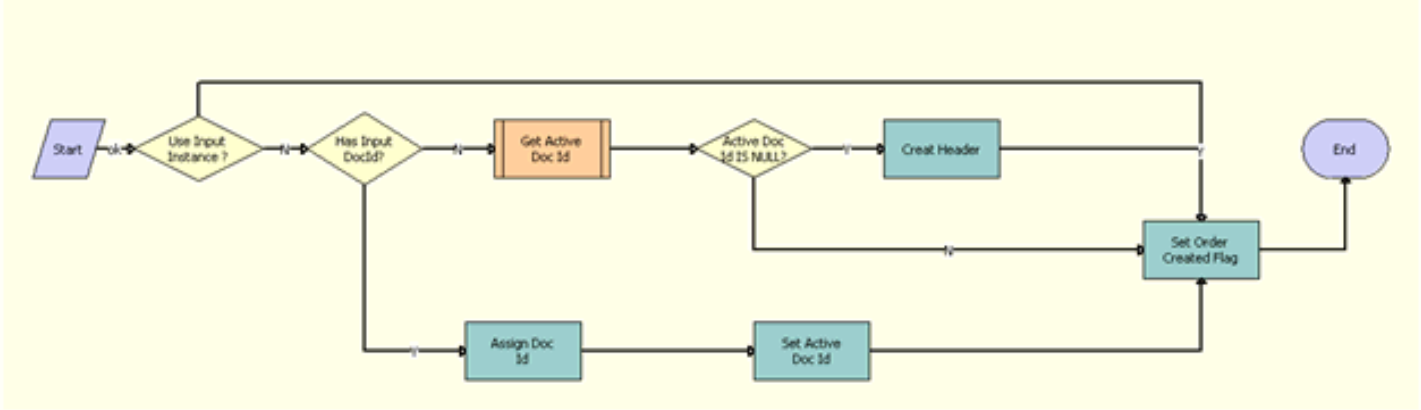
- **Get Context.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Context to get the account header information.
- **Get Rowset.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Row Set to get all asset information for the account.
- **SkipProdOffer?** This decision step routes flow to SkipPromRecmndation? if the SkipProdOffer flag is Y. Otherwise, it routes the flow to Get Product Recommendations.
- **Get Product Recommendations.** This step calls the subprocess Product Recommendation Get Recommended Products, which returns all product offers recommendations for all the assets in the account, and which also performs eligibility and compatibility checks.
- **Remove Messages.** If there is an exception, this step calls the method DeleteMessages of the business service Unified Messaging Service and sets the source to Product Recommendation.
- **Update Messages.** This step calls the method UpdateMessages of the business service Unified Messaging Service and sets the source to Product Recommendation.
- **SkipPromRecmndation?** This decision step routes flow to Any Recommendations if SkipPromRecmndation flag is Y. Otherwise, it routes the flow to Promotion Recommendation For Asset.
- **Promotion Recommendation For Asset.** This step calls the subprocess ISS Promotion Recommendation For Assets to get all promotion recommendations.
- **Any Recommendations.** This decision step routes flow to Any Offer? if there are no recommendations. Otherwise, it routes the flow to End.
- **Any Offer?** This decision step routes flow to Any Promotion? if there are no product Offer recommendations. Otherwise, it routes the flow to End.
- **Any Promotion?** This decision step routes flow to Build - No Recommendations Message if there are no promotion recommendations. Otherwise, it routes the flow to End.
- **Build - No Recommendations Message.** This step calls the method Echo of the business service Workflow Utilities to get the display message if there are no promotion and product offer recommendations.
- **Load - No Recommendations Msg.** This step calls the method UpdateMessages of the business service Unified Messaging Service and sets the source to Zero Recommendation.

RAF Prepare Header Sub Process Workflow

RAF Prepare Header Sub Process workflow, shown in the following figure, checks whether the input specifies a quote or order that must be used.

If the input specifies a quote or order to be used, it does not create a new quote or order. Instead, it checks whether the input includes a DocId. If so, it sets the value of the DocId as the value of the active quote or active order Id.

If the input does not specify a quote or order to be used or if the input does not include a DocId, then there is no active document, and the workflow calls RAF Get Active Document Workflow to create a new quote or order. If there is no active document it generates the order header for further processing without synchronizing it to the database, so there is not an empty order header created if there is a failure during processing of offer actions.



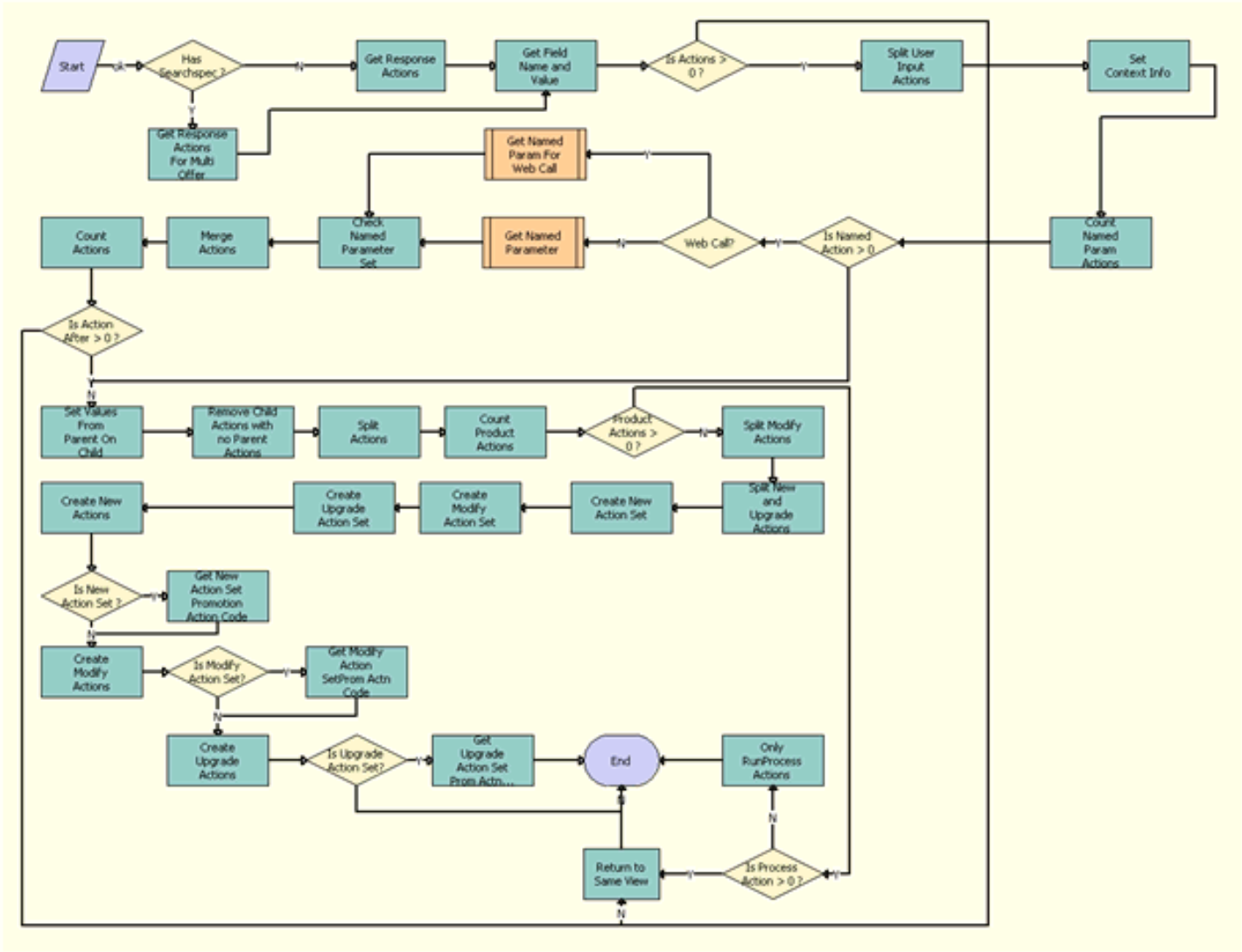
Workflow Description

This workflow performs the following actions:

- **Use Input Instance ?** This decision step checks whether an input document must be used while processing the order or whether a new quote or order must be generated.
- **Has Input DocId?** This decision step checks whether the input includes a DocId. If so, the specified quote or order is used as the active quote or order.
- **Get Active Doc Id.** This step calls the subprocess RAF Get Active Document Workflow to get the active DocId, which is the row_id of the latest pending quote or order.
- **Active Doc Id IS NULL?** This decision step routes the flow to Create Header or Set Order Created flag based whether the Active Doc Id is NULL.
- **Create Header.** This step calls the Conditional Action Transform method of the Row Set Transformation Toolkit business service to create a quote or order header template with required fields, which is used in further processing.
- **Assign Doc Id.** This step calls the Echo method of Workflow Utilities business service to assign the value of DocId from the input as the Active Doc Id.
- **Set Active Doc Id.** This step calls the Set Profile Attribute method of the SIS OM PMT Service business service to set the Order Profile Attribute Name or Quote Profile Attribute Name to the Active Doc Id.
- **Set Order Created Flag.** This Step calls the Echo method of the Workflow Utilities business service to set the Order Created Flag to Y.

RAF Prepare Actions Sub Process Workflow

RAF Prepare Actions Sub Process Workflow, shown in the following figure, queries for all the actions defined for the particular product offer and sorts them into different property sets depending on the action type. It is called as a subprocess from the Prepare Actions step of the parent workflow RAF Create Document Workflow.



Workflow description

This workflow performs the following actions:

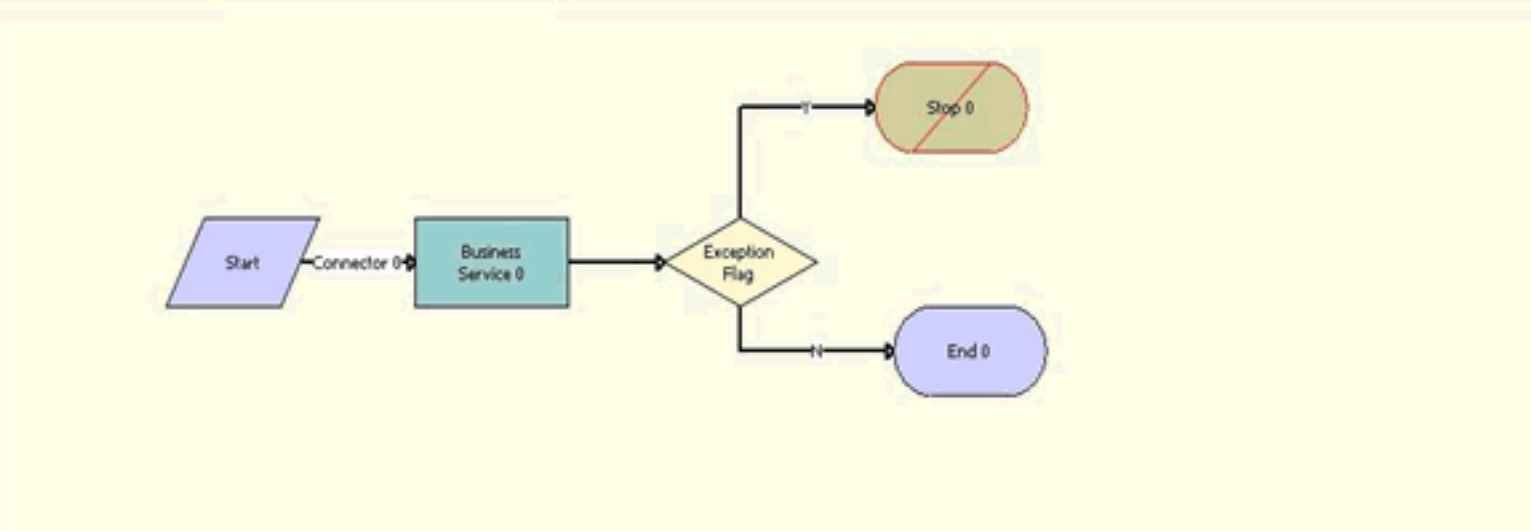
- **Has Searchspec ?** This decision step checks whether the input has value populated for OfferSerachSpec populated. If so, it directs the flow to Get Response Actions for Multi Offer (for more than one offer). If not, it directs the flow to Get Response Actions (for one offer).
- **Get Response Actions For Multi Offer.** This step calls Query Transform method of the Row Set Transformation Toolkit business service using the value of OfferSearchSpec to query actions for multiple offers.
- **Get Response Actions.** This step queries the RAF Product Offer Action business component to get all the actions defined for the product offer and the response which are not inactive. It gets the count of actions as well as building an actions property set which has details of each action.
- **Get Field Name and Value.** This step calls the method GetFieldValueCMU of the business service External Integration Service to get the value of the field Prod Prom Name.
- **Is Actions > 0?** This decision step routes the flow to Split User Input Actions if the actions count is greater than 0. Otherwise, it routes the flow to Return to Same View.

- **Split User Input Actions.** This step calls the method Split Transform of the business service Row Set Transformation Toolkit to prepare two row sets, splitting the User Input Action row set into one row set with actions for which the Named Parameter Flag is Y, and one with actions for which the flag is N.
- **Set Context Info.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to set the Account Id from the context to the value in the User Input Action row set.
- **Count Named Param Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to count the number of actions in User Input Action row set.
- **Is Named Action > 0.** This decision step routes the flow to Web Call? if the named action count is greater than 0. Otherwise, it routes the flow to Is Process Action > 0?
- **Web Call?** This decision step routes flow to Get Named Param For Web Call if it is a Web service call. Otherwise, it routes the flow to Get Named Parameter.
- **Get Named Parameter.** This step calls the subprocess RAF Get Named Parameter Workflow to process the User Input Action row set.
- **Get Named Param For Web Call.** This step calls the subprocess RAF Get Named Parameter For Web Call Workflow to process the User Input Action row set in the Web service call.
- **Check Named Parameter Set.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to set the argument Named Parameter Set to Y, in context if it is Y in User Input Action row set. Otherwise, it sets the this argument to N.
- **Merge Actions.** This step calls the method Merge Transform of the business service Row Set Transformation Toolkit to merge the Response Actions and User Input Action row sets.
- **Count Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to count the number of actions in the Response Actions row set.
- **Is Action After > 0?** This decision step routes the flow to Set Values From Parent On Child if Action Count after is greater than 0. Otherwise, it routes the flow to Return to Same View.
- **Set Values From Parent On Child.** This step calls the method SetParentActionValuesInChild of the business service RAF Product Offers Service to set the values of the child actions field based on the parent actions field values, if the child action field value is empty. In other words, if the based product is null in child, then it will get the value from the parent action.
- **Remove Child Actions with no Parent Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to remove actions which do not have parent action id entered from the response actions.
- **Split Actions.** This step calls the method Split Transform of the business service Row Set Transformation Toolkit to prepare two row sets, the RunProcess row set with actions for which the Action Type is Runprocess, and the Actions row set for other actions.
- **Count Product Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to count the number of actions of type other than Run Process.
- **Product Actions > 0?** This decision step routes the flow to Split Modify Actions if the product action count is greater than 0. Otherwise, it routes the flow to Is Process Action > 0?
- **Split Modify Actions.** This step calls the method Split Transform of the business service Row Set Transformation Toolkit to prepare the row set Modify Actions for actions where the action type is Modify.
- **Split New and Upgrade Actions.** This step calls the method Split Transform of the business service Row Set Transformation Toolkit to prepare the row set New Actions for actions where action type is New and the row set Upgrade Actions for other actions.
- **Create New Action Set.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to create the New Actions row set.

- **Create Modify Action Set.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to create the Modify Actions row set.
- **Create Upgrade Action Set.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to create the Upgrade Actions row set.
- **Create New Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to fill the New Actions row set with actions of the type New.
- **Is New Action Set?** This decision step routes the flow to Get New Action Set Promotion Action Code if the New Actions row set exists. Otherwise, it routes the flow to Create Modify Actions.
- **Get New Action Set Promotion Action Code.** This step calls the method GetPromotionActionCode of the business service ABO Bulk Request Validation Service to get promotion action codes for the New Actions row set.
- **Create Modify Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to fill the Modify Actions row set with actions of the type Modify.
- **Is Modify Action Set?** This decision step routes the flow to Get Modify Action SetProm Actn Code if the Modify Actions row set exists. Otherwise, it routes the flow to Create Upgrade Actions.
- **Get Modify Action SetProm Actn Code.** This step calls the method GetPromotionActionCode of the business service ABO Bulk Request Validation Service to get promotion action codes for the Modify Actions row set.
- **Create Upgrade Actions.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to fill the Upgrade Actions row set with actions of the type Upgrade.
- **Is Upgrade Action Set?** This decision step routes the flow to Get Upgrade Action Set Prom Actn Code if the Upgrade Actions row set exists. Otherwise, it routes the flow to End.
- **Get Upgrade Action Set Prom Actn Code.** This step calls the method GetPromotionActionCode of the business service ABO Bulk Request Validation Service to get promotion action codes for the Upgrade Actions row set.
- **Is Process Action > 0?** This decision step routes the flow to Only RunProcess Actions if the action count is greater than 0. Otherwise, it routes the flow to Return to Same View.
- **Return to Same View.** This step calls the method Echo of the business service Workflow Utilities to set return to same view to Y.
- **Only RunProcess Actions.** This step calls the method Echo of the business service Workflow Utilities to set Only RunProcess Actions to Y.

RAF Check Prepaid Balance Workflow

RAF Check Prepaid Balance Workflow, shown in the following figure, is called when the action type is Run Process and the workflow name is SIS OM Submit Order Process.



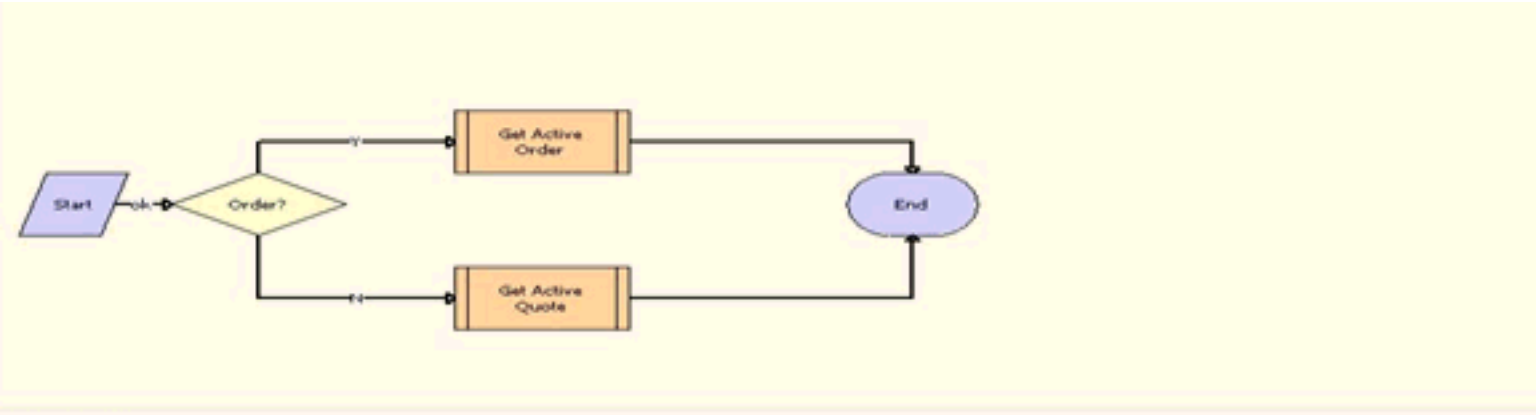
Workflow Description

This workflow performs the following actions:

- **Business Service 0.** This step calls the method GetFieldValue of the business service CMU External Integration Service to get values for fields Minimum Balance Required and Exception Flag from the context.
- **Exception Flag.** This decision step routes the flow to Stop 0 if Exception Flag is Y. Otherwise, it routes the flow to End.
- **Stop 0.** This step displays the error message: Exception: Order failed. Offer requires customer to have a minimum balance of [&Minimum Balance Required].

RAF Get Active Document Workflow

RAF Get Active Document Workflow, shown in the following figure, gets the ID of the active document. (the row_id of the latest pending order or latest open quote). It is called as a subprocess from the Get Active Doc Id step of the workflow RAF Prepare Header Sub Process.



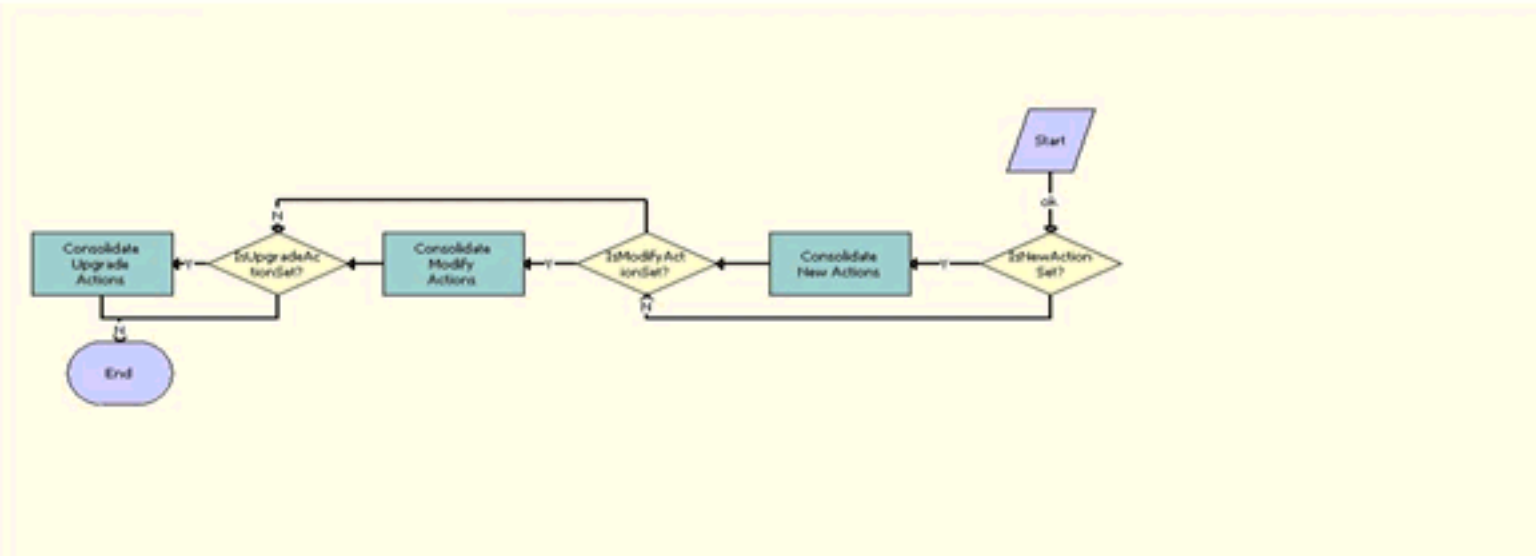
Workflow Description

This workflow performs the following actions:

- **Order?** This decision step routes the flow to Get Active Order if Account Type is Residential. Otherwise, it routes the flow to Get Active Quote.
- **Get Active Order.** This step calls the subprocess SIS OM Active Order Sub-Process and passes it the values of Account Id and Is RAF flag.
- **Get Active Quote.** This step calls the subprocess SIS OM Active Quote Sub-process and passes it the values of Account Id and Is RAF flag.

RAF Consolidate Actions Sub Process Workflow

RAF Consolidate Actions Sub Process Workflow, shown in the following figure, consolidates all actions based on the action type (other than the action type Run Process), and it groups them under one base product. It is called as a subprocess from the Consolidate Actions step of the RAF Create Document Workflow.



Workflow Description

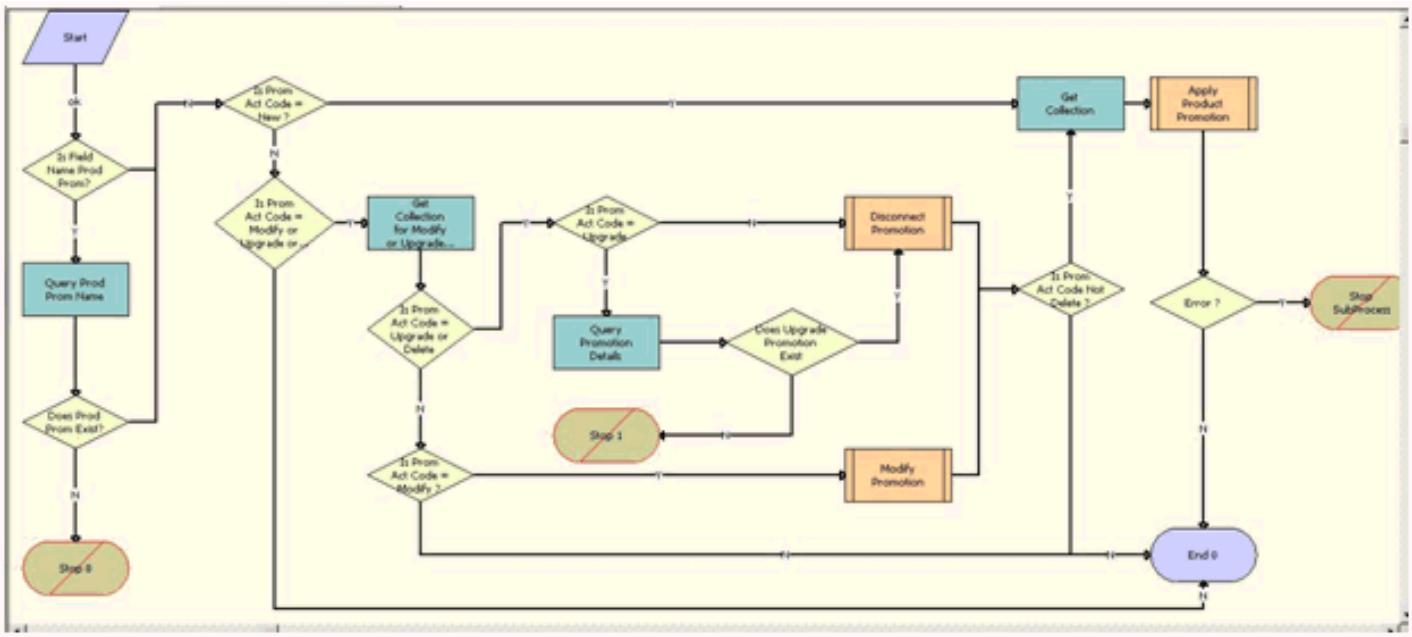
This workflow performs the following actions:

- **IsNewActionSet?** This decision step routes the flow to Consolidate New Actions if IsNewActionSet is not NULL. Otherwise, it routes the flow to IsModifyActionSet?
- **Consolidate New Actions.** This step calls the method ConsolidateAllActions of the business service ABO Bulk Request Processing Service to consolidate all actions in new action set under one Base product.
- **IsModifyActionSet?** This decision step routes the flow to Consolidate Modify Actions if IsModifyActionSet is not NULL. Otherwise, it routes the flow to IsUpgradeActionSet?
- **Consolidate Modify Actions.** This step calls the method ConsolidateAllActions of the business service ABO Bulk Request Processing Service to consolidate all actions in Modify action set under one Base product.
- **IsUpgradeActionSet?** This decision step routes the flow to Consolidate Upgrade Actions if IsUpgradeActionSet is not NULL. Otherwise, it routes the flow to End.

- **Consolidate Upgrade Actions.** This step calls the method ConsolidateAllActions of the business service ABO Bulk Request Processing Service to consolidate all actions in Upgrade action set under one Base product.

RAF Check For Promotion Sub Process Workflow

RAF Check For Promotion Sub Process Workflow, shown in the following figure, checks whether the promotion Name is correct and applies the Apply, Disconnect, Modify or Upgrade promotion. It is called as a subprocess from the Check For Promotion and Apply Promotion steps of the RAF Create Document Workflow.



Workflow Description

This workflow performs the following actions:

- **Is Field Name Prod Prom?** This decision step routes the flow to Query Prod Prom Name if the field name is Prod Prom Name. Otherwise, it routes the flow to Is Prom Act Code = New?
- **Query Prod Prom Name.** This step queries the business component ISS Promotion for the field name to see whether this promotion exists.
- **Does Prod Prom Exist?** This decision step routes the flow to Is Prom Act Code = New? if the promotion name does not exist in the business component. Otherwise, it routes the flow to Stop 0.
- **Stop 0.** This error step displays the error message, “You selected a Invalid value/values for the Promotion Creation for <prom name>. Please change your selection and submit your request again."
- **Is Prom Act Code = New?** This decision step routes the flow to Get Collection if

```
NewActnSetPromActnCode='New' OR ModifyActnSetPromActnCode='New'
```

Otherwise, it routes the flow to Is Prom Act Code = Modify or Upgrade or Delete.

- **Is Prom Act Code = Modify or Upgrade or Delete?** This decision step routes the flow to Get Collection for Modify or Upgrade or Delete if

```
ModifyActnSetPromActnCode='Modify' OR ModifyActnSetPromActnCode='Delete' OR
UpgrdActnSetPromActnCode='Upgrade'.
```

  Otherwise, it routes the flow to End.
- **Get Collection for Modify or Upgrade or Delete.** This step calls the method GetCollection of the business service ABO Bulk Request Processing Service to get details of the promotion to be deleted or, in case of upgrade, to get details of the promotion to be upgraded to.
- **Is Prom Act Code = Upgrade or Delete?** This decision step routes the flow to Is Prom Act Code = Upgrade? if

```
ModifyActnSetPromActnCode='Delete' OR UpgrdActnSetPromActnCode]='Upgrade'
```

  Otherwise, it routes the flow to Is Prom Act Code = Modify?
- **Is Prom Act Code = Modify?** This decision step routes the flow to Modify Promotion if

```
ModifyActnSetPromActnCode='Modify '.
```

  Otherwise, it routes the flow to End.
- **Is Prom Act Code = Upgrade.** This decision step routes the flow to Query Promotion Details if

```
UpgrdActnSetPromActnCode]='Upgrade'
```

  Otherwise, it routes the flow to Disconnect Promotion.
- **Query Promotion Details.** This step queries business component ISS Promotion Upgrade To to get target promotion details.
- **Does Upgrade Promotion Exist.** This decision step routes the flow to Disconnect Promotion if

```
[&Target Promotion Id] = [&Upgrade Promotion Id].
```

  Otherwise, it routes the flow to Stop 1.
- **Stop 1.** This error step displays the error message “You selected a Invalid value/values for the Promotion Creation for %1.Please change your selection and submit your request again."
- **Disconnect Promotion.** This step calls the subprocess ABO Bulk Request - Disconnect Products & Services Process to disconnect the existing promotion from the account’s assets.
- **Modify Promotion.** This step calls the subprocess ABO Bulk Request - ModifyWebService-Order to modify the existing promotion.
- **Is Prom Act Code Not Delete?** This decision step routes the flow to Get Collection if

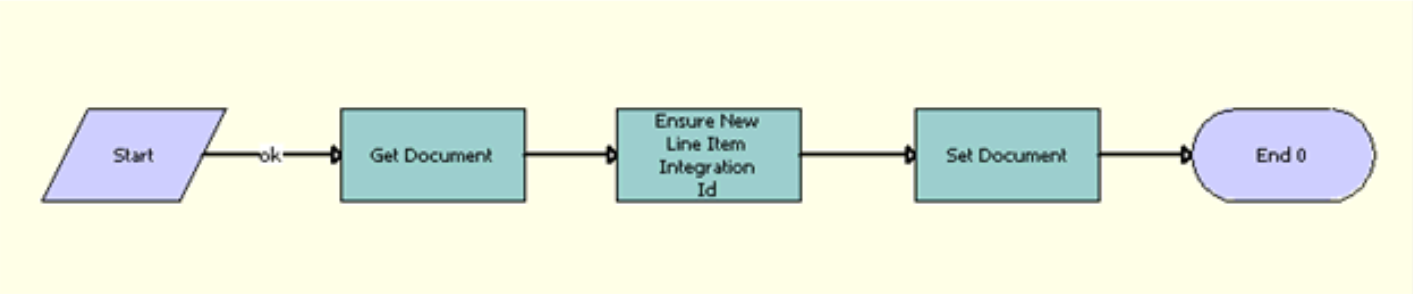
```
ModifyActnSetPromActnCode='Delete '
```

Otherwise, it routes the flow to End.

- **Get Collection.** This step calls the method GetCollection of the business service ABO Bulk Request Processing Service to get details of the promotion, such as Prom Inst Id, Prod Prom Name, from the current instance.
- **Apply Product Promotion.** This step calls the subprocess ABO Bulk Request - ApplyProductPromotion to apply the promotion change to the asset.
- **Error?** This decision step routes the flow to Stop SubProcess if Err Flag is Y. Otherwise, it routes the flow to End.
- **Stop SubProcess.** This error step displays the error message from the step Apply Product Promotion.

RAF Ensure New Line Item Workflow

RAF Ensure New Line Item Workflow, shown in the following figure, is a sub process that is invoked from RAF Create Document workflow. It extracts the Siebel message from the collection, stamps the Integration Id, and adds the Siebel message back to collection that would be used in the subsequent step to synchronize the row set data.



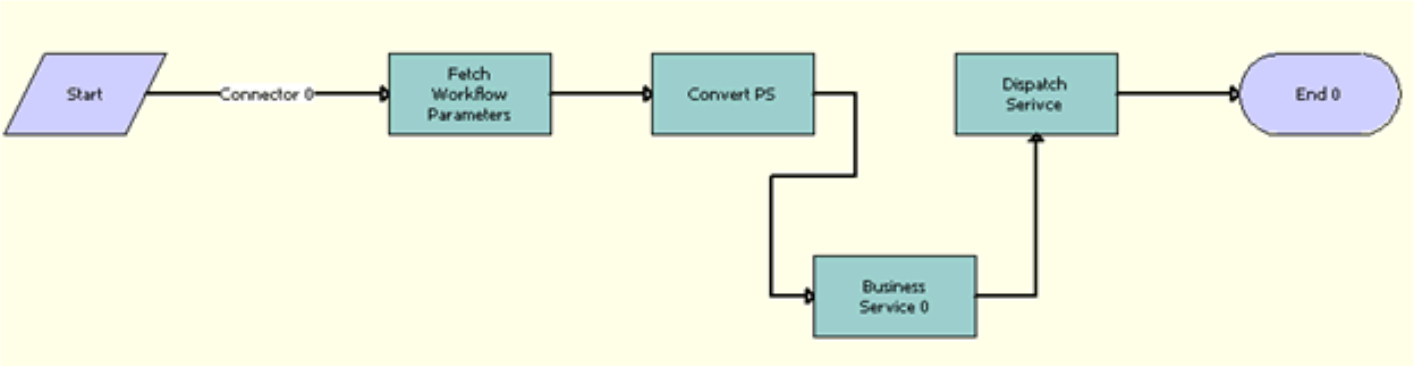
Workflow Description

This workflow performs the following actions:

- **Get Document.** This step calls the Get Collection method of the ABO Bulk Request Processing Service business service to extract the Siebel message from the input property set which is used during the synchronization of the document.
- **Ensure New Line Item Integration Id.** This step calls the Set Field Value method of the SIS OM PMT business service to add the Integration Id field to the Siebel message which is the output of the previous step.
- **Set Document.** This step calls the AddItemtoCollection method of the ABO Bulk Request Processing business service, which adds the Siebel message from the previous step back to the collection.

RAF RunProcess Driver Workflow

RAF RunProcess Driver Workflow, shown in the following figure, is invoked whenever there is a Run Process Action Type. It takes Context, Row Set, Active Doc Id, Asset Id and Account Id as input arguments.



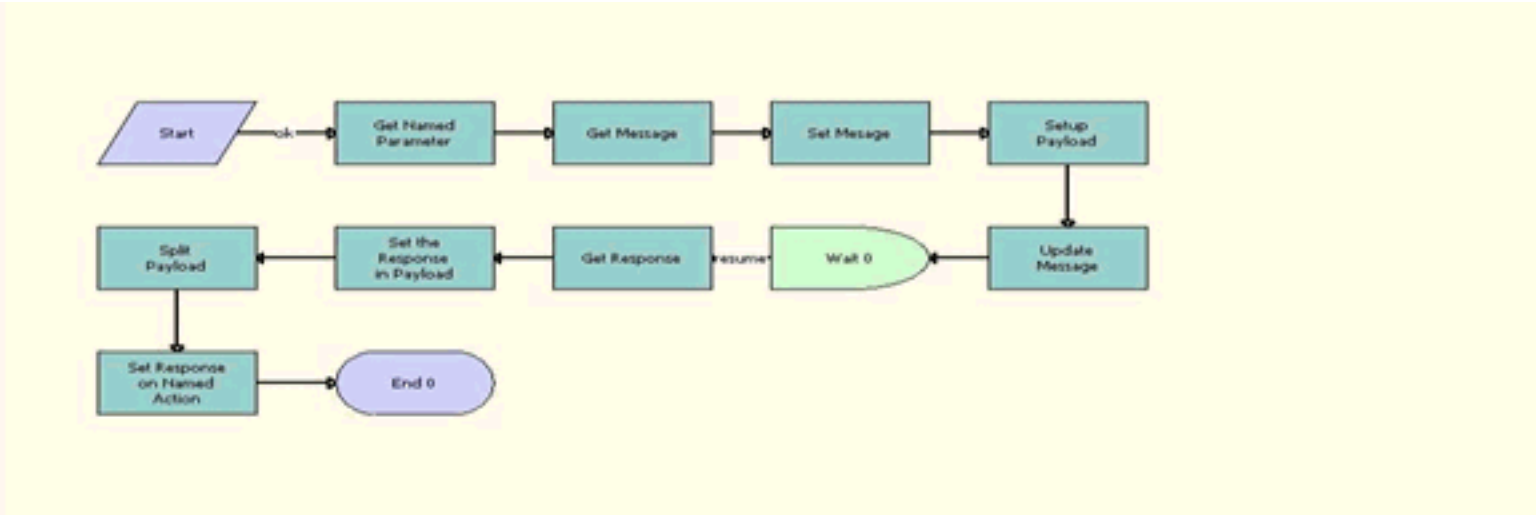
Workflow Description

This workflow performs the following actions:

- **Fetch Workflow Parameters.** This step calls the Query Transform method of the Row Set Transformation Toolkit business service to fetch the row set related to the Run Process action, querying the business component RAF Product Offer Workflow Parameters using the Action Id as the search specification.
- **Convert PS.** This step calls the ConvertPS method of the RAF Product Offers business service, which is a specialized method that takes Context and the output of the previous step (that is, the Workflow Parameters row set) and merges the workflow parameters into the context which is the output.
- **Business Service 0.** This step calls the GetFieldValue method of the CMU External Integration business service to extract Activity Description and Type from the context.
- **Dispatch Service.** This step calls the CallPSPWorkflow method of the ISS PSP Dispatch business service, which takes Context, Row Set, Instance Name and PSPWorkflowName, which is the Workflow name defined in the Run Process Action, thus invoking the required workflow.

RAF Get Named Parameter Workflow

RAF Get Named Parameter Workflow, shown in the following figure, is called as a subprocess from the Get Named Parameter step of the RAF Prepare Actions Sub Process workflow to process the User Input Action row set.

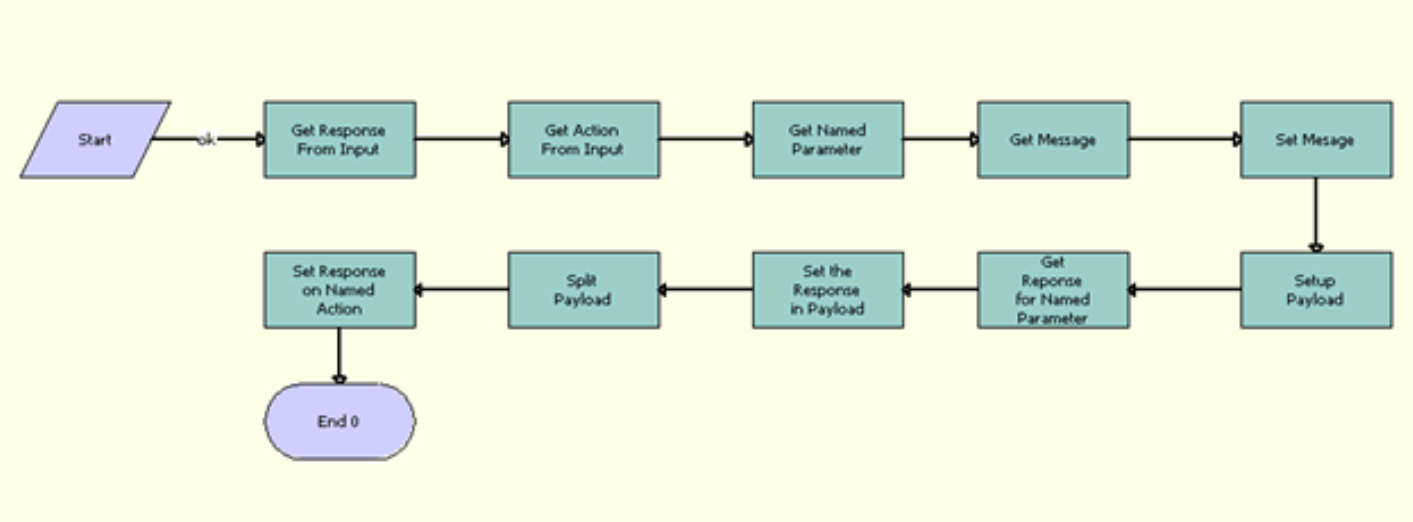


Workflow Description

- **Get Named Parameter.** This step calls the method Query Transform of the business service Row Set Transformation Toolkit to get the named parameters from the business component RAF Product Offer Parameters defined under the current action.
- **Get Message.** This step calls the method Query Transform of the business service Row Set Transformation Toolkit to get the message from the business component UMS Type and save it in the Msg row set.
- **Set Message.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to set message details on the Named Param row set from the Msg row set.
- **Setup Payload.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to set up the payload for the accept and reject responses.
- **UpdateMessage.** This step calls the method UpdateMessages of the business service Unified Messaging Service to the push message list to UMF Active Messages popup list applet.
- **Wait 0.** This step calls the method Sleep of the business service Workflow Utilities to wait until the user clicks OK or Cancel.
- **Get Response.** This step calls the GetMsgResponse method of the business service RAF Product Offers Service to get the response for the selected message in the process property Msg Response.
- **Set the Response in Payload.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to the set response for each message in the response row set.
- **Split Payload.** This step calls the method Split Transform of the business service Row Set Transformation Toolkit to create the Payload row set from the rows of the Response row set where the action Id is not null.
- **Set Response on Named Action.** This step calls the method Conditional Action Transform of the business service Row Set Transformation Toolkit to set details of the selected message in the User Input Action row set.

RAF Get Named Parameter For Web Call Workflow

RAF Get Named Parameter For Web Call Workflow, shown in the following figure, is the subprocess call for named parameters that use Web services.



Workflow Description

This workflow performs the following actions:

- **Get Response From Input.** This step uses the GetFieldValue method of the CMU External Integration business service to extract Action Name and Action Id from the Siebel message that is the input parameter.
- **Get Action From Input.** This step uses the GetFieldValue method of the CMU External Integration business service to extract Named Parameter Id, Upgrade Product, Component Product, Field Name, Attribute Value from the Siebel message that is the input parameter.
- **Get Named Parameter.** This step uses the Query Transform method of the Row Set Transformation Toolkit to query the RAF Product Offer Parameters business component to get the Named Parameter and User Input row sets.
- **Get Message.** This step uses the Query Transform method of the Row Set Transformation Toolkit to query the UMS Message business component to get the Msg row set.
- **Set Message.** This step uses the Conditional Action Transform method of the Row Set Transformation Toolkit to query based on the condition `{Row.Name} IS NOT NULL` from the output of the previous step to set the Named Parameter row set and the Msg row set as Row Set and Context respectively.
- **Set Up Payload.** To set the context and row set as the output, this step uses the Conditional Action Transform method of the Row Set Transformation Toolkit to query based on condition

```
 {Row} += New(Payload", [Product Type Code] = {Row.Product Type Code} , [Account
Id] = {Context.Account Id} ,[Reject Action Type] = "RuntimeEvent" , [Accept Action
Type] = "RuntimeEvent" , [Reject Action Name] = "EventMethodResumeParamWflw" ,
[Accept Action Name] = "EventMethodResumeParamWflw" , [Base Product Name] =
{Context.Base Product Name} , [Action Id] = {Context.Id} , [Named Parameter Flag]
= {Context.Named Parameter Flag} , [Product Path] = {Row.Component Product Path}
, [Named Parameter Type] ={Context.Named Parameter Type} , [Base Product Id] =
{Context.Base Product Id} , [Field Value] = {Row.Field Value} , [Product Name]=
{Row.Product Name} , [Product Id] = {Row.Product Id} , [Component Product] =
{Row.Component Product} , [Name] = {Row.Name} , [Attribute Value] = {Row.Attribute
Value} )
```

- **Get Response for Named Parameter.** To obtain the Response row set, this step uses the Query Transform method of the Row Set Transformation Toolkit to query RAF Product Offer Parameters business component based on the condition

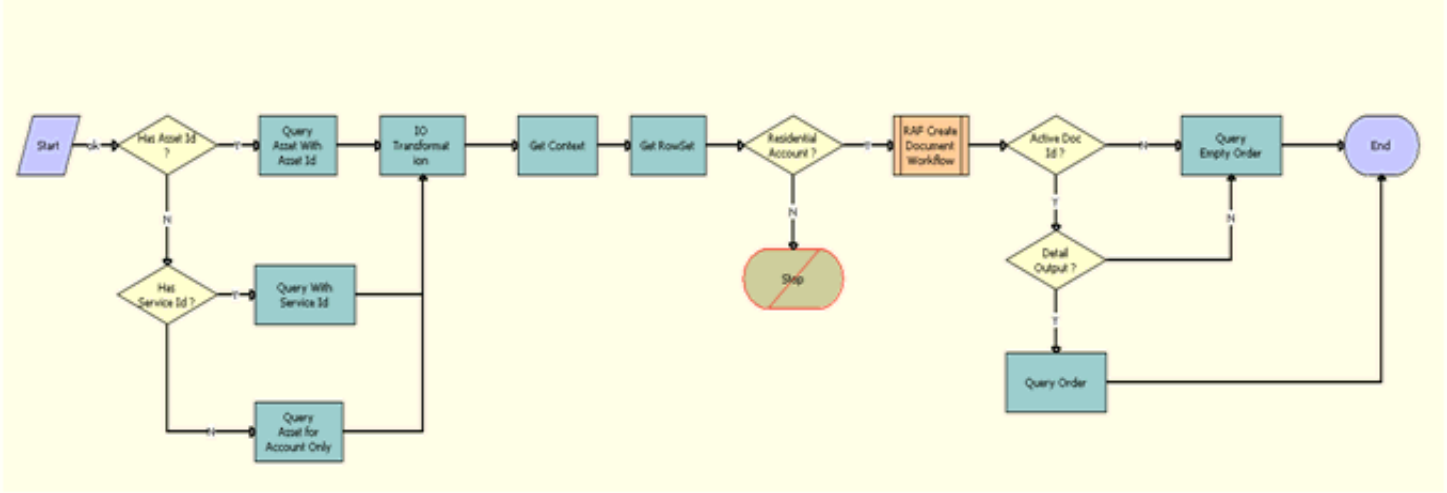
```
 '[Product Offer Action Id] = ' + '"' + [&ActionId] + '" AND [Id] = ' + '"' +
[&ParameterId] + '"'
```

- **Set the Response in Payload.** This step uses Conditional Action Transform method of the Row Set Transformation Toolkit to query based on the condition `{Row.Action Id} IS NOT NULL` to get the Response row set and Child Response row set.
- **Split Payload.** This step uses the Split Transform method of the Row Set Transformation Toolkit to split the output row set into Payload row set and Response row set based on the condition `{Row.Action Id} IS NOT NULL`.
- **Set Response on Named Action.** To arrive at the row sets pertaining to conditions satisfied having User Input Action Rowset and Payload Rowset respectively, this step uses the Conditional Action Transform method of the Row Set Transformation Toolkit to query based on the conditions

```
 {Row.Named Parameter Type} = LookupValue("RAF_NAMED_PARM_TYPE","Component
Product") or {Row.Named Parameter Type} = LookupValue("RAF_NAMED_PARM_TYPE","Base
Product") or {Row.Named Parameter Type} =
LookupValue("RAF_NAMED_PARM_TYPE","Upgrade Product") or {Row.Named Parameter
Type} = LookupValue("RAF_NAMED_PARM_TYPE","Field Value") or {Row.Named Parameter
Type} = LookupValue("RAF_NAMED_PARM_TYPE","Domain Value").
```

RAF Apply Offer To Create Order Web Workflow

RAF Apply Offer To Create Order Web Workflow, shown in the following figure, takes account, asset, and offer information and applies the offers for a residential customer to create an order. It also sends an error message as part of its output if there is an error while processing the offer. It is called while calling method ApplyOfferToCreateOrder of the Web service Product Offer Web Service.



Workflow Description

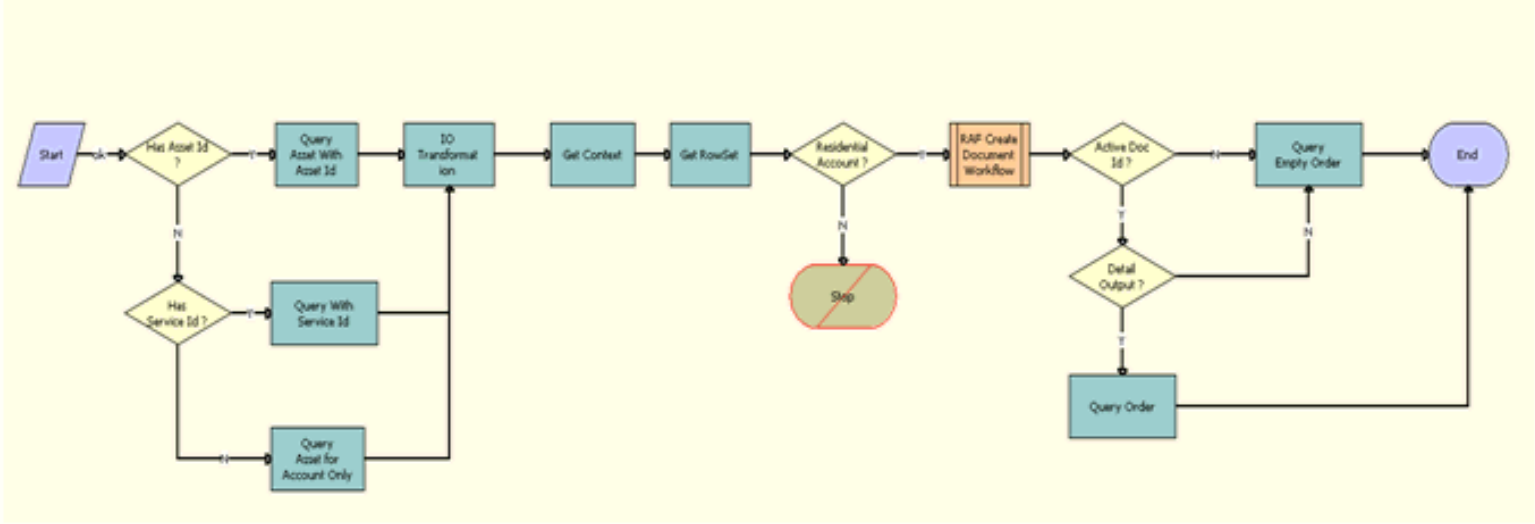
This workflow performs the following actions:

- **Has Asset Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateOrder contains AssetId. If so, it routes the flow to the step Query Asset With Asset Id. Otherwise, it routes the flow to the step Has Service Id?
- **Has Service Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateOrder contains ServiceId. If so, it routes the flow to the step Query With Service Id. Otherwise, it routes the flow to the step Query Asset for Account Only.
- **Query Asset With Asset Id.** This step calls Query method of the business service EAI Siebel Adapter, using the Account Id and Asset Id to retrieve all the asset information for the specified account and asset.
- **Query With Service Id.** This step calls the Query method of the business service EAI Siebel Adapter, using the Account Id and Service Id to retrieve all the asset information for the specified account and service Id.
- **Query Asset for Account Only.** This step calls the Query method of business service EAI Siebel Adapter, using the Account Id to retrieve all the asset information for the specified account.
- **IO Transformation.** This step calls the method IntObjHierToPSHier of the business service IntObjHierToPSHier to transform the Input from Integration Object format to Property Set Format.

- **Get Context.** This step calls the method GetRowSetData of Context Service business service. It uses Variable Map Default Product Offer Recommendation Variable Map - Context to get the account header information.
- **Get Rowset.** This step calls the method GetRowSetData of Context Service business service. It uses Variable Map Default Product Offer Recommendation Variable Map - Row Set to get all asset information for the account.
- **Residential Account?** This decision step checks whether the account in the input of the Web service operation ApplyOfferToCreateOrder is of the type Residential. If so, it routes the flow to the step RAF Create Document Workflow. Otherwise, it routes the flow to the Stop step to set the error message.
- **Stop.** This stop step sets an error message as part of output message of the Web service operation ApplyOfferToCreateOrder, if the account is not a residential account.
- **RAF Create Document Workflow.** This step calls the subprocess RAF Create Document Workflow to apply the offer on the account, and to return the Order Id.
- **Active Doc Id?** This step verifies if the offer is successfully applied by creating an order with a non-null value for the DocId. If DocId (Order Id) is NULL, it routes the flow to the step Query Empty Order. Otherwise, it routes the flow to the step Detail Output?
- **Detail Output?** This step verifies if DetailOutput is set as Y in the input, to determine whether only the order Id or the full order structure is part of output message. If it is evaluated to TRUE, it routes the flow to the step Query Order. Otherwise, it routes the flow to Query Empty Order.
- **Query Order.** This step calls the Query method of the business service EAI Siebel Adapter by using the newly created order after applying the offer. It takes the DocId (Order Id) as input and returns the full order structure as the output that would be part of the response message to the Web service call.
- **Query Empty Order.** This step calls the Query method of the business service EAI Siebel Adapter, querying an invalid order to populate the empty value for the order message as part of response message the Web service call.

RAF Apply Offer To Create Quote Web Service Workflow

RAF Apply Offer To Create Quote Web Service Workflow, shown in the following figure, takes account, asset, and offer information and applies the offers for a nonresidential customer to create a quote. It also sends an error message as part of the output if there is an error while processing the offer. It is called while calling the method ApplyOfferToCreateQuote of the Web service Product Offer Web Service.



Workflow Description

This workflow performs the following actions:

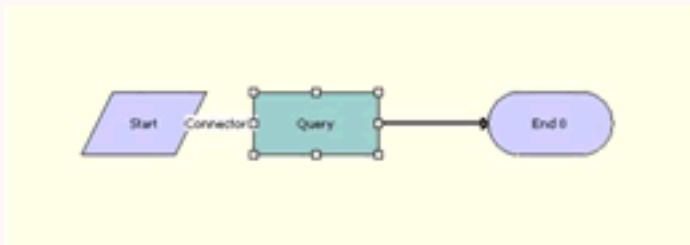
- **Has Asset Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateQuote contains the asset Id. If so, it routes the flow to the step Query Asset With Asset Id. Otherwise, it routes the flow to the step Has Service Id?
- **Has Service Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateQuote contains the service Id. If so, it routes the flow to the step Query With Service Id. Otherwise, it routes the flow to the step Query Asset for Account Only.
- **Query Asset With Asset Id.** This step calls the Query method of the business service EAI Siebel Adapter by using the Account Id and Asset Id to retrieve all the asset information for the specified account and asset.
- **Query With Service Id.** This step calls the Query method of the business service EAI Siebel Adapter by using the Account Id and Service Id to retrieve all the asset information for the specified account and service.
- **Query Asset for Account Only.** This step calls the Query method of business service EAI Siebel Adapter by using the Account Id to retrieve all the asset Information for the specified account.
- **IO Transformation.** This step calls the method IntObjHierToPSHier of the business service IntObjHierToPSHier to transform the Input from Integration Object format to Property Set Format.
- **Get Context.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Context to get the account header information.
- **Get Rowset.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Row Set to get all asset information for the account.
- **Residential Account?** This decision step checks whether the account in the input of the Web service operation ApplyOfferToCreateQuote is of the type Residential. If not, it routes the flow to the step RAF Create Document Workflow. Otherwise, it routes the flow route to the Stop step to set the error message.
- **Stop.** This stop step sets the error message as part of output message of the Web service operation ApplyOfferToCreateQuote if the account is a residential account.
- **RAF Create Document Workflow.** This step calls the subprocess RAF Create Document Workflow to apply the offer on the account, and it returns the Quote Id.

- **Active Doc Id?** This step verifies if the offer is successfully applied by creating a quote with a non-null value for the DocId. If DocId (Quote Id) is NULL, it routes the flow to the step Query Empty Quote. Otherwise, it routes the flow to the step Detail Output?
- **Detail Output?** This step verifies if DetailOutput is set to Y in the input to determine whether to set only the quote Id or the full quote structure as part of output message. If it is evaluated to TRUE, it routes the flow to the step Query Quote. Otherwise, it routes the flow to the step Query Empty Quote.
- **Query Quote.** This step calls the Query method of the business service EAI Siebel Adapter by using the newly created quote after applying the offer. It takes the DocId (Quote Id) as input and returns the full quote structure as the output that will be part of the response message to the Web service call.
- **Query Empty Quote.** This step calls the Query method of the business service EAI Siebel Adapter by querying a invalid quote to populate the empty value for the quote message which will be used as part of response message the Web service call.

Product Offer GetProductOffer Workflow

Product Offer GetProductOffer Workflow, shown in the following figure, queries the product offer business component on ProductOfferId. It is called when calling the method GetProductOffer of the Web service Product Offer Web Service.

> **Note:** For more information about the request and response messages for this workflows, see *Siebel CRM Web Services Reference* .



Workflow Description

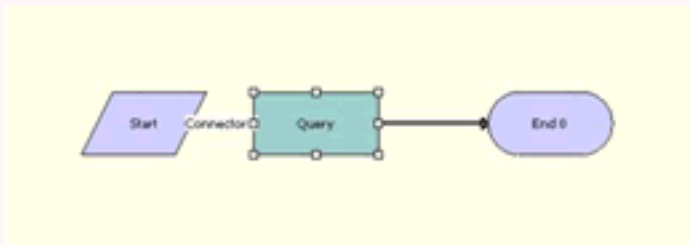
This workflow performs the following action:

- **Query.** This step calls the Query method of the business service EAI Siebel Adapter, using the ProductOfferId to retrieve the product offer from RAFProductOfferListIO. If the search specification is the asterisk wildcard, it returns all the offers.

Product Offer GetProductOfferStructure Workflow

Product Offer GetProductOfferStructure Workflow, shown in the following figure, queries the product offer business component on ProductOfferId to get the product offer structure. It is called when calling the method GetProductOfferStructure of the Web service Product Offer Web Service.

> **Note:** For more information about the request and response messages for this workflows, see *Siebel CRM Web Services Reference* .



Workflow Description

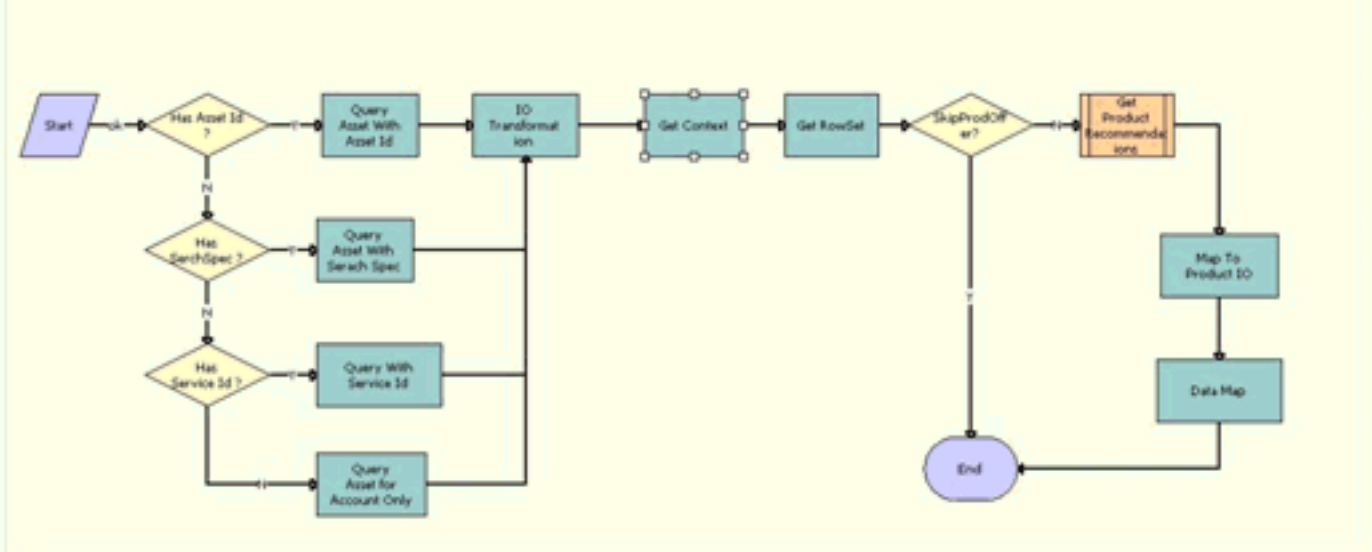
This workflow performs the following actions:

- **Query Product Offer.** This step calls the Query method of the business service EAI Siebel Adapter by using the ProductOfferId to retrieve the product offer structure from RAFProductOfferStructureIO.

Product Offer GetRecommendedProductOffers Workflow

Product Offer GetRecommendedProductOffers Workflow, shown in the following figure, is a wrapper workflow which calls Product Recommendation Get Recommended Products. It is called when calling the method GetRecommendedProductOffers of the Web service Product Offer Web Service.

Note: For more information about the request and response messages for this workflows, see *Siebel CRM Web Services Reference* .



Workflow Description

This workflow performs the following actions:

- **Has Asset Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateQuote contains the asset Id. If so, then it routes the flow to the step Query Asset With Asset Id. Otherwise, it routes the flow to the step Has Searchspec?

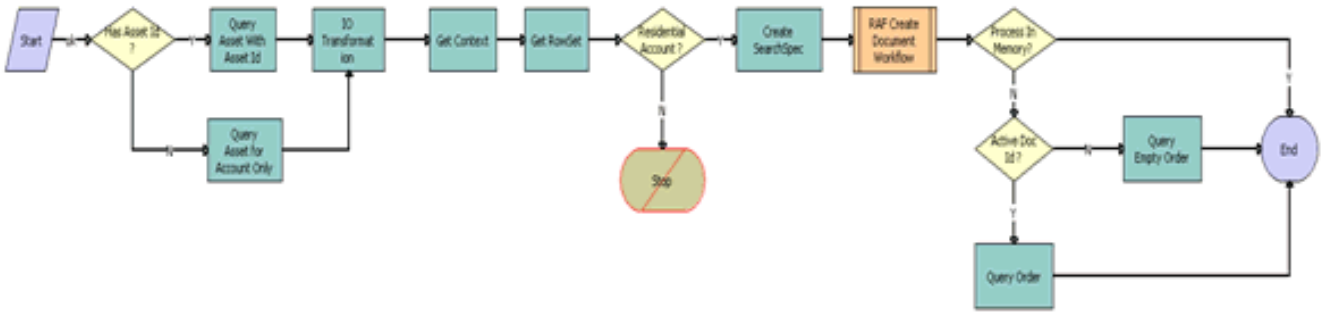
- **Has Searchspec?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateQuote contains searchspec. If so, then it routes the flow to the step Query Asset With Search Spec. Otherwise, it routes the flow to the step Has Service Id?
- **Has Service Id?** This decision step checks whether the input of the Web service operation ApplyOfferToCreateQuote contains the service Id. If so, then it routes the flow to the step Query With Service Id. Otherwise, it routes the flow to the step Query Asset for Account Only.
- **Query Asset With Asset Id.** This step calls the Query method of the business service EAI Siebel Adapter by using the account Id and asset Id to retrieve all the asset information for the specified account and asset.
- **Query Asset with Search spec.** This step calls the Query method of the business service EAI Siebel Adapter by using the searchspec to retrieve all the asset information for the specified account and asset.
- **Query With Service Id.** This step calls the Query method of the business service EAI Siebel Adapter, using the account Id and service Id to retrieve all the asset information for the specified account and service.
- **Query Asset for Account Only.** This step calls the Query method of business service EAI Siebel Adapter, using the account Id to retrieve all the asset information for the specified account.
- **IO Transformation.** This step calls the method IntObjHierToPSHier of the business service IntObjHierToPSHier to transform the input from integration object format to property set format.
- **Get Context.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Context to get the account header information.
- **Get Rowset.** This step calls the method GetRowSetData of the Context Service business service. It uses the variable map Default Product Offer Recommendation Variable Map - Row Set to get all asset information for the account.
- **SkipProdOffer?** This decision step routes the flow to End if skip prod offer is Y. Otherwise it routes the flow to Get Product Recommendations.
- **Get Product Recommendations.** This step calls the subprocess Product Recommendation Get Recommended Products by passing context and rowset.
- **Map To Product IO.** This step calls the method PSHierToIntObjHier of the business service EAI Integration Object to XML Hierarchy Converter to transform the Input from Property Set Format to Integration Object format.
- **Data Map.** This step calls the method Execute of the business service EAI Data Transformation Engine to get the output in RAF Offer Asset Data Map, which will be sent as output of the web service.

RAF Apply Multiple Offer To Create Order Web Service Workflow

RAF Apply Multiple Offer To Create Order Workflow, shown in the following figure, takes the account and a list of offer IDs to generate an order for a residential customer. The order can be generated in-memory or saved to the database based on the SaveInstance input argument.

This workflow is called by the ApplyMultipleOffersforOrder method of Apply Multiple Product Offers Web service. Optionally, this workflow accepts the order structure as an input argument and applies offer actions to this order structure instead of creating a new order. For existing customers, the workflow takes the Asset Id as an input argument and offers' are applied to the customer's asset.

Note: For more information about the Web service that calls this workflow, see *Siebel CRM Web Services Reference* .



Workflow Description

This workflow performs the following actions:

1. **Has Asset Id?** This decision step checks whether the input of the Web service operation ApplyMultipleOfferToCreateOrder contains AssetId. If so, it routes the flow to the step Query Asset With Asset Id. Otherwise, it routes the flow to the step Query Asset for Account Only.
2. **Query Asset With Asset Id.** This step calls Query method of the business service EAI Siebel Adapter, using the Account Id and Asset Id to retrieve the asset Information for the specified account and asset.
3. **Query Asset for Account Only.** This step calls the Query method of business service EAI Siebel Adapter, using the Account Id to retrieve the Account details from Integration object ISS Account Instance. This step is called when Asset Id is not part of the input message of the Web service.
4. **IO Transformation.** This step calls the method IntObjHierToPSHier of the business service EAI Integration Object to XML Hierarchy Converter to transform the queried message from the previous step from Integration Object format to Property Set Format.
5. **Get Context.** This step calls the GetRowSetData method of the Context Service business service. It uses the Variable Map Default Product Offer Recommendation Variable Map - Context to get the account header information.
6. **Get Rowset.** This step calls the GetRowSetData method of the Context Service business service. It uses the Variable Map Default Product Offer Recommendation Variable Map - Row Set to get the asset information for the account.
7. **Residential Account?** This decision step checks whether the account in the input of the Web service is of the type Residential. If so, it routes the flow to the step Create SearchSpec. Otherwise, it routes the flow to the Stop step to set the error message.
8. **Stop.** This stop step sets an error message as part of output message of the Web service operation ApplyMultipleOfferToCreateOrder, if the account is not a residential account. It aborts further processing and exits the workflow.
9. **Create SearchSpec.** This step calls the CreateSearchSpec method of the RAF Product Offers Service business service, which traverses the list of offers from the input message of the web service and prepares a search expression with the offer IDs.
10. **RAF Create Document Workflow.** This step calls the subprocess RAF Create Document Workflow to process the actions for all the offers provided in the input message. If the order is generated in-memory, the subprocess returns the Order structure. If the order is saved to the database, the subprocess returns the Order ID.
11. **Process In Memory?** This step checks if the order processing is done in memory. If so, it returns the order structure as the output argument of the web service. Otherwise, it routes the flow to Active Doc Id.
12. **Active Doc Id?** This step verifies whether the offer is successfully applied by creating an order with a non-null value for the DocId. If DocId (Order Id) is NULL, it routes the flow to the step Query Empty Order. Otherwise, it routes the flow to the step Query Order.

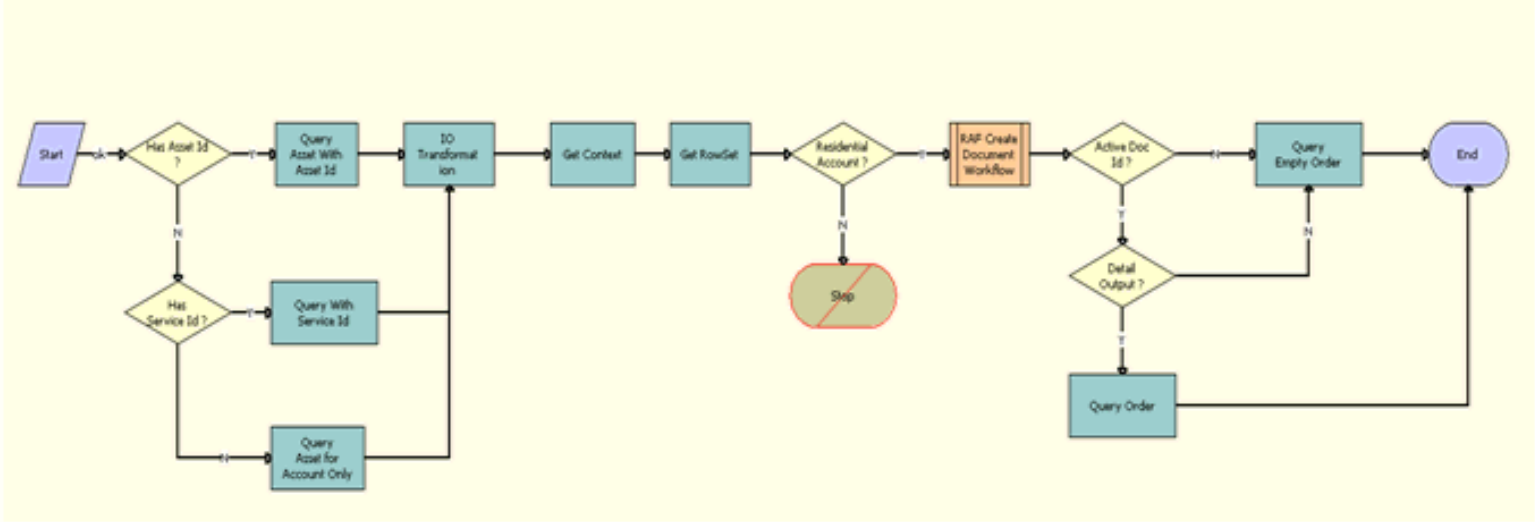
13. **Query Empty Order.** This step calls the Query method of the EAI Siebel Adapter business service, querying an invalid order to populate the empty value for the order message as part of response message the Web service call.
14. **Query Order.** This step calls the Query method of the EAI Siebel Adapter business service to retrieve the order in SIS OM Order integration object format. It takes the DocId (Order ID) as input and returns the full order structure as the output that would be part of the response message to the Web service call.

RAF Apply Multiple Offer To Create Quote Web Service Workflow

RAF Apply Multiple Offer To Create Quote Workflow, shown in the following figure, takes the account and a list of offer IDs to generate a quote for a non-residential customer. This quote can be generated in-memory or saved to the database, based on the SaveInstance input argument.

This workflow is called by the ApplyMultipleOffersforQuote method of the Apply Multiple Product Offers Web service. Optionally, this workflow accepts the quote structure as an input argument, and offer actions are applied to the quote structure instead of creating a new quote. For existing customers, this workflow takes the Asset ID as an input argument and applies offers to the customer's asset.

Note: For more information about the Web service that calls this workflows, see *Siebel CRM Web Services Reference* .



Workflow Description

This workflow performs the following actions:

1. **Has Asset Id ?** This decision step checks whether the input of the Web service operation ApplyMultipleOfferToCreateQuote contains AssetId. If so, it routes the flow to the step Query Asset With Asset Id. Otherwise, it routes the flow to the step Query Asset for Account Only.
2. **Query Asset With Asset Id.** This step calls Query method of the EAI Siebel Adapter business service, using the Account Id and Asset Id to retrieve the asset information for the specified account and asset.
3. **Query Asset for Account Only.** This step calls the Query method of the EAI Siebel Adapter business service, using the Account Id to retrieve the Account details from Integration object ISS Account Instance. This step is called when Asset Id is not part of the input message of the web service.

4. **IO Transformation.** This step calls the IntObjHierToPSHier method of the EAI Integration Object to XML Hierarchy Converter business service to transform the queried message from the previous step from Integration Object format to Property Set Format.
5. **Get Context.** This step calls the GetRowSetData method of the Context Service business service. It uses the Default Product Offer Recommendation Variable Map - Context variable map to get the account header information.
6. **Get Rowset.** This step calls the GetRowSetData method of the Context Service business service. It uses Default Product Offer Recommendation Variable Map - Row Set variable map to get the asset information for the account.
7. **Residential Account?** This decision step checks whether the account in the input of the Web service is of the type Residential. If not, it routes the flow to the step Create SearchSpec. Otherwise, it routes the flow to the Stop step to set the error message.
8. **Stop.** If the account is a residential account, this stop step sets an error message as part of output message of the ApplyMultipleOfferToCreateQuote Web service operation. It aborts further processing and exits the workflow.
9. **Create SearchSpec.** This step calls the CreateSearchSpec method of the RAF Product Offers Service business service, which traverses the list of offers from the input message of the Web service and prepares a search expression with offer IDs.
10. **RAF Create Document Workflow.** This step calls the subprocess RAF Create Document Workflow to process the actions for all the offers provided in the input message. If the quote is generated in-memory, the subprocess returns the Quote structure. If the quote is saved to the database, the subprocess returns the Quote ID.
11. **Process In Memory?** This step checks if the quote processing is done in memory. If so, it returns the quote structure as the output argument of the web service. Otherwise, it routes the flow to Active Doc Id.
12. **Active Doc Id?** This step verifies whether the offer is successfully applied by creating an quote with a non-null value for the DocId. If DocId (Quote Id) is NULL, it routes the flow to the step Query Empty Quote. Otherwise, it routes the flow to the step Query Quote.
13. **Query Empty Quote.** This step calls the Query method of the EAI Siebel Adapter business service, querying an invalid quote to populate the empty value for the quote message as part of response message the Web service call.
14. **Query Quote.** This step calls the Query method of the EAI Siebel Adapter business service to retrieve the order in SIS OM Quote integration object format. It takes the DocId (Quote Id) as input and returns the full quote structure as the output that would be part of the response message to the Web service call.

10 Creating Discount Matrices

Creating Discount Matrices

This chapter covers discount matrices and includes the following topics:

- *About Discount Matrices*
- *About Discount Matrix Dimensions*
- *About Discount Matrix Types*
- *Defining a Discount Matrix*
- *Entering Discount Matrix Details*
- *Discount Matrix Integration*
- *About Configuring Discount Matrices*

About Discount Matrices

In many industries, prices and price adjustments often depend on both the product that is being sold and other information relevant to the sale. Often, customer and market conditions affect the price for which a product is sold.

Discount matrices allow you to tailor prices and adjustments to specific customer and market segments. They allow you to create and manage prices and adjustments for products in conjunction with other customer and market factors. The discount matrix administrative framework allows pricing administrators to logically group factors that result in a price or adjustment, working in the Siebel client without changing the Siebel repository.

For example, a company that sells laptop computers can create a discount matrix that allows it to charge a different price for laptops based on customer account type and sales region. In this example, customer account type, product, and sales region are the criteria for the price or discount. Therefore, the columns of the matrix would be Product, Account Type, Sales Region, Adjustment Type, Adjustment Amount, and a few additional columns that will be covered later in this chapter. Each row of the matrix defines the price or discount that will be applied when a given product is sold to a given account type in a given sales region, as shown in the following table.

Product	Account Type	Sales Region	Adjustment Type	Adjustment Amount
Laptop 2000	Platinum	Europe	% Discount	10
Laptop 2000	Gold	Europe	% Discount	5
Laptop 2000	Platinum	Asia	% Discount	11
Laptop 2000	Gold	Asia	% Discount	6

Discount matrices allow you to maintain all of this critical pricing information in a single matrix, instead of dispersing it across numerous Siebel views.

Discount matrices are recommended in cases where:

- There is a high volume of prices and adjustments.
- The criteria for prices and adjustments are known ahead of time and do not change frequently.

Because prices and adjustments are frequently stored in disparate applications, discount matrices:

- Support integration with SAP condition tables.
- Can be loaded using Enterprise Integration Manager (EIM) or Enterprise Application Integration (EAI) integration objects.
- Can be integrated with legacy applications using other XML-based integration technologies.

About Discount Matrix Dimensions

A discount matrix is a logical grouping of prices or adjustments that apply when specific combinations of criteria for prices or adjustments are met. For example, a discount matrix might contain all the adjustments for products when sold to specific account types, or all the adjustments for products when sold in specific regions.

Before you create discount matrices, you must determine the criteria for the prices or adjustments, based on your company's business model. After identifying criteria, you may begin creating the discount matrix. The criteria for the prices or adjustments will be the unique dimensions (or columns) of the discount matrix.

Along with the criteria dimensions, your discount matrix will include other dimensions. To continue the example, assume your company provides different discounts for each product based on the account type it is sold to and the region where it is sold. It would use the following dimensions:

- **Criteria Dimensions.** These are the fields that uniquely identify the adjustment. In the example, these dimensions would be Product, Account Type, and Region. For each record, all of the criteria dimensions must be populated.
- **Calculation Dimensions.** These fields allow administrators to enter various adjustments for the unique combinations of the discount criteria across various effective dates. These fields include Adjustment Type, Adjustment Amount, Currency Code, Effective Start Date, and Effective End Date.
- **Result Dimensions.** These fields allow administrators to associate other information to the adjustment record that may be needed in the quote, order, or agreement later in the pricing procedure, or by back-end applications. These fields include Volume Discount Schedule, Attribute Adjustment Matrix, Pricing Procedure, Aggregate Discount, and Accounting Code.

The discount details list will include the criteria dimensions that were selected when you created the discount matrix, as well as the default calculation and result dimensions. Each set of dimensions is configurable and extensible.

About Discount Matrix Types

The following topics describes the three types of discount matrices, and the criteria, calculation, and result dimensions that are available for each type:

- *Product-Based Adjustment Discount Matrices*

- *Entitlement-Based Adjustment Discount Matrices*
- *Training-Based Adjustment Discount Matrices*

The fields that are available by default are the fields that are most commonly used with each type of adjustment. In case your business model demands different criteria dimensions, each set is configurable and extensible, as described in *About Configuring Discount Matrices*.

Product-Based Adjustment Discount Matrices

Product-Based Adjustment Discount Matrices allow administrators to define adjustments based on product, customer, and market information.

By default, the administrator can choose from the following dimensions:

- Criteria dimensions to identify product:
  - Product
  - Parent Product
  - Root Product
  - Product Line
  - Product Class
  - Price Type
- Criteria dimensions to identify customer:
  - Account
  - Account Type
- Criteria dimensions to identify market:
  - Region
  - Sales Organization
  - Division Code
  - Distribution Channel
- Dimensions to identify transaction:
  - Action
  - Approval Status
- Calculation dimensions:
  - Adjustment Type
  - Adjustment Amount
  - Currency Code
  - Effective Start Date
  - Effective End Date
- Result dimensions:

- Volume discount
- Attribute Adjustment
- Pricing Procedure
- Aggregate Discount Sequence
- Accounting Code

Entitlement-Based Adjustment Discount Matrices

Entitlement-Based Adjustment Discount Matrices allow administrators to define adjustments based on product, entitlement, and transactional information.

By default, the administrator can choose from the following criteria dimensions:

- Criteria dimensions to identify product:
  - Product
  - Parent Product
  - Root Product
  - Product Line
  - Price Type
- Criteria dimensions to identify transaction:
  - Entitlement
  - Action
  - Qty from
  - Qty to
  - Reason Code
- Calculation dimensions:
  - Adjustment Type
  - Adjustment Amount
  - Currency Code
  - Effective Start Date
  - Effective End Date
- Result dimensions:
  - Volume discount
  - Attribute Adjustment
  - Pricing Procedure
  - Aggregate Discount Sequence

- Accounting Code

Training-Based Adjustment Discount Matrices

Training-Based Adjustment Discount Matrices allow administrators to define adjustments based on account, transaction, and training event information.

By default, the administrator can choose from the following dimensions:

- Criteria dimensions to identify training:
  - Training Course
  - Event
  - Event Type
  - Event Location
  - Price Type
- Criteria dimensions to identify customer:
  - Account
  - Contact
- Criteria dimensions to identify transaction:
  - Action
  - Reason Code
  - Approval Status
- Calculation dimensions:
  - Adjustment Type
  - Adjustment Amount
  - Currency Code
  - Effective Start Date
  - Effective End Date
- Result dimensions:
  - Volume discount
  - Attribute Adjustment
  - Pricing Procedure
  - Aggregate Discount Sequence
  - Accounting Code

Process of Creating a Discount Matrix

To create a discount matrix, perform the following steps:

Defining a Discount Matrix

When you create a discount matrix, you specify what type of discount matrix it is, and which criteria dimensions will be shown in the Discount Details list applet.

To define a discount matrix

1. Navigate to the Administration - Pricing screen, then Discount Matrices View.
2. Add a new record to the Discount Matrices list and complete the necessary fields, as described in the following table.

Field	Comments
Name	Enter a name for the discount matrix.
Status	This field displays the status of the discount matrix. When you click the Complete Definition button, the value changes from In Progress to Active. You cannot enter any discount details until the status is Active.
Discount Matrix Type	Select one of the discount matrix types. The options are as follows: o **Product-Based Adjustment.** For more information, see *Product-Based Adjustment Discount Matrices*. o **Entitlement-Based Adjustment.** For more information, see *Entitlement-Based Adjustment Discount Matrices*. o **Training-Based Adjustment.** For more information, see *Training-Based Adjustment Discount Matrices*.

3. In the Discount Criteria list, add records for all the criteria dimensions that you want exposed in the discount matrix.

   The criteria dimensions that you can add depend on the value selected in the Discount Matrix Type field in Step 2.
4. After all the necessary criteria dimensions are added, click Complete Definition in the Discount Matrix list applet.

   Clicking this button freezes the definition of the discount matrix. After this step, you cannot change the discount matrix type or the criteria dimensions.
5. In the Discount Matrix list, click the name of the new discount matrix.

   The Discount Details list appears. It includes the criteria dimensions that you selected in Step 3 and the default calculation and result dimensions.

Entering Discount Matrix Details

After a discount matrix has been defined, you enter details of each price adjustment in the matrix.

You must make entries in all the criteria dimension columns that have been defined.

Enter effective dates in the same way as price lists, as described in *Giving a Product Multiple Prices with Different Effective Dates*. If you create records with overlapping effective dates, then the product resolves the issues in the same way as price lists, as described in *About Effective Dates for Pricing Line Items*.

When you enter Discount Detail records, enter a value in the Currency Code field if the Adjustment Type is Price Override, Discount Amount, or Markup Amount. By default, the Currency Code field value is the system currency.

To enter details in a discount matrix

1. Navigate to the Administration - Pricing screen, then Discount Matrices View.
2. In the Discount Matrices list, click the name of the matrix to which you want to add details.

   The dimensions in the Discount Details list applet are as follows:

   - The criteria dimensions specified on the discount matrix
   - The default calculation dimensions
   - The default result dimensions
3. In the Discount Details list, add a new record for each combination of criteria dimension values for which you want to define a price adjustment.

   In each record, you must enter values for all the criteria dimensions.

Modifying the Pricing Procedure - Calculate Net Price Workflow to Include a Discount Matrix

When you use a discount matrix, you must added an additional step to the workflow Pricing Procedure - Calculate Net Price to apply the discount matrix to the pricing. For more information about this workflow, see *Pricing Procedure - Calculate Net Price Workflow*.

To modifying the Pricing Procedure - Calculate Net Price workflow to include a discount matrix

1. In Web Tools, select the Pricing Procedure - Calculate Net Price workflow process. Revise and edit this workflow process.
2. After the Tiered Volume Discount Upsell step, add a new business service step that queries the appropriate discount matrix business component for any discounts associated with each account type.

   a. Define the step as shown in the following table.

Name	Type	Business Service Name	Business Service Method
Account Type Adjustments	Business Service	Row Set Transformation Toolkit	Simple Look-Up Transform

b. Define the input arguments as shown in the following table.

Input Argument	Type	Value	Property Name
Context	Process Property		Context
Row Set	Process Property		Row Set
Business Object	Literal	Adjustment Group	
Business Component	Literal	Product-Based Adjustment	
Process Condition	Literal	{Row.Product Id} IS NOT NULL AND {Context.Account Type} IS NOT NULL	
Search Specification	Literal	[Adjustment Group Name] = 'Account Type Discount' AND [Account Type] = {Context.Account Type}	
On First Match 1	Literal	{Row.Net Price} = AdjustPrice({Row.Net Price}, {Row.Currency Code}, LookupName("PRICER_CALCULATION_SIMPLE_TYPE", {Match.Adjustment Type}), {Match.Adjustment Amount}, {Match.Currency Code}, {Row.Exchange Date})	
On First Match 2	Literal	{Row}.{Net Price Waterfall} += New("Waterfall", [Adjustment Amount] = {Match.Adjustment Amount}, [Adjustment Type] = {Match.Adjustment Type}, [Price] = {Row.Net Price}, [Currency Code] = {Row.Currency Code}, [Text] = LookUpMessage ("Pricer - Account Type Discount",[Account Type] = {Match.Account Type} , [Discount Matrix] = {Match.Adjustment Group Name}))	

c. Define the output arguments as shown in the following table.

Output Argument	Property Name	Type
Context	Context	Output Argument
Row Set	Row Set	Output Argument

3. Deploy the updated workflow process.

Discount Matrix Integration

You can populate discount matrices with data from other applications, load them using EIM, and systematically update them using EAI. They are designed for integration with SAP condition tables.

Loading and Updating Data from Other Applications

Siebel Enterprise Integration Manager (EIM) tables are available for each discount matrix table.

To initially load data from another application, use EIM. For more information, see *Siebel Enterprise Integration Manager Administration Guide* .

Integration objects are available for each discount matrix table to enable integration using EAI.

For systematic incremental updates, use EAI. For more information, see *Overview: Siebel Enterprise Application Integration* .

Integration with SAP Condition Tables

The following SAP-specific dimensions columns are available for the product-based and entitlement-based discount matrix types:

- Sales Organization Code
- Division Code
- Distribution Channel Code

The following SAP-specific fields are available in the Discount Matrix table (S_ADJ_GROUP):

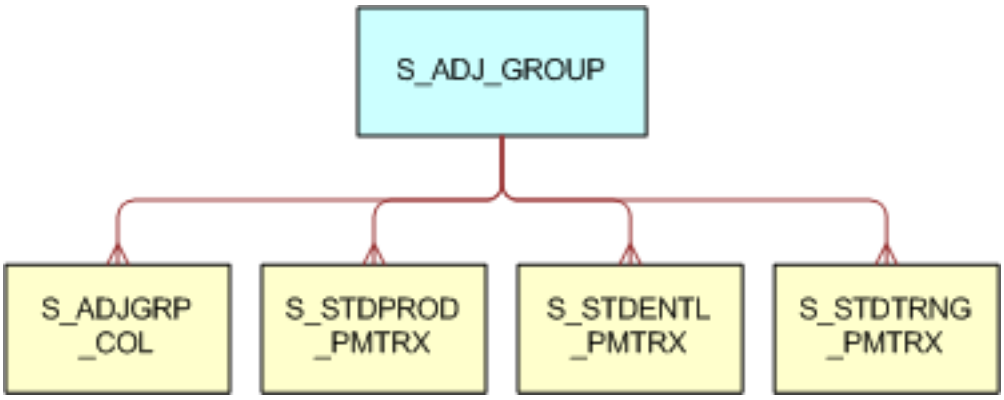
- SAP Condition Type: SAP_COND_NAME
- SAP Condition Table: SAP_TABLE_NAME

About Configuring Discount Matrices

The criteria, calculation, and result dimensions that are available for each discount matrix type are the ones that are most commonly used for that type of adjustment. However, your business model may require you to use additional dimensions that are not available by default.

To make these additional dimensions available, you can configure the application with Web Tools. For more information about Web Tools, see *Using Siebel Tools* .

To make additional dimensions available, you must understand the tables used to store the discount matrix data, which are shown in the following schema.



Discount matrix fields are stored in the following tables:

- The Discount Matrix Name, Status, and Type are stored in the S_ADJ_GROUP table.
- The criteria dimensions that the administrator has selected for the discount matrix are stored in the S_ADJ_GRP_COL table.
- The discount details are stored in the following tables:
  - S_STDPROD_PMTRX (if the discount matrix type is Product-Based Adjustment)
  - S_STDENTL_PMTRX (if the discount matrix type is Entitlement-Based Adjustment)
  - S_STDTRNG_PMTRX (if the discount matrix type is Training-Based Adjustment)

To extend the available criteria, calculation, or result dimensions, you may add columns to any or all of these tables.

The following topics include instructions for configuring discount matrices:

- *Adding New Dimensions to Existing Discount Matrix Types*
- *Creating New Discount Matrix Types for Custom Tables*

Adding New Dimensions to Existing Discount Matrix Types

> **Note:** This topic includes instructions that assume you are familiar with Web Tools. For more information, see *Using Siebel Tools* . These instructions also assume that you have made the appropriate data model extensions to the appropriate discount details table.

You must use Web Tools to add new dimensions to existing discount matrix types.

To add new dimensions to existing discount matrix types using Web Tools

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. In the Object Explorer, select the Business Component Object.
3. Query to find the appropriate discount matrix Business Component (Product-Based Adjustment, Entitlement-Based Adjustment, or Training-Based Adjustment).
4. Add a new Business Component field based on the newly created extension column.
5. In the Object Explorer, select the Business Component User Prop object under the Business Component object.
6. Add the new field name to the appropriate user property.

   > **Note:** If the field will be used as a criteria dimension, then add it to the Dimension Columns user property. If the field will be used as a calculation or result dimension, then add it to the Default Columns user property. The appropriate syntax for the user properties is as follows: #FieldName#FieldName#FieldName#
7. In the Object Explorer, select the Applet object.
8. Query to find the appropriate discount details applet (Product-Based Adjustment list applet, Entitlement-Based Adjustment list applet, or Training-Based Adjustment list applet).
9. Create a new list column for the new field.
10. Bind the list column to the Edit List Web template.
11. Deliver the applet changes to the Siebel runtime repository.

If the new field was added as a criteria dimension, then it now appears as a selection in the discount criteria pick applet for the appropriate discount matrix type. If the new field was added as a calculation or result dimension, then it now appears in the Discount Details applet when the user drills down on a discount matrix with the appropriate discount matrix type.

Creating New Discount Matrix Types for Custom Tables

> **Note:** This topic includes instructions that assume you are familiar with Web Tools. For more information, see *Using Siebel Tools* .

You must use Web Tools to create new discount matrix types for custom tables.

To create a new discount matrix based on a custom table using Web Tools

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. Create a Business Component based on the custom table.

   For example create the Default Columns user property:

   Name: Default Columns

   Value: <buscomp field names separated by # sign>

This user property controls which columns always display in the Discount Matrix Details List applet for the new Discount Matrix Type.

3. Create the Dimension Columns user property:

   Name: Dimension Columns

   Value: <buscomp field names separated by # sign>

   This user property controls which criteria dimensions will appear in the discount criteria pick applet for the new Discount Matrix Type.

4. Add the new matrix type to the Discount Matrix Type LOV:

   Type: ADJ_GROUP_TYPE

   Display Value: <the new matrix type>

   LIC: <the new matrix type>

5. Modify the Discount Matrix Business Component by adding the following Business Component User Property:

   Name: <Display Value of New Matrix Type from LOV>

   Value: Adjustment Group#New Matrix BC Name#

   This user property inserts the Business Object and Business Component values into the appropriate fields on the discount matrix header when the discount matrix type is selected.

6. Create a new Discount Matrix Details list applet for the new Business Component:

   Name: <any name>

   Business Component: <new matrix buscomp>

   Class: CSSSWEFrameListPSPAdjRule

   List Columns: <all fields from the new matrix Business Component that could possibly appear as criteria or default dimensions>

   > **Note:** Specialized applet class will only show dimensions defined and default columns from the Business Component user property.

7. Create a new Discount Matrix Details view as follows:

   Name: <any name>

   Business Object: Adjustment Group

   View Web Template Items:

   Adjustment Group form applet

   New Applet from Above

8. Modify the Adjustment Group list applet as follows:

   Create a New Drilldown Object:

   Name: <new matrix type>

Hyperlink Field: Name

Destination View: <name of new details view>

Create a Dynamic Drilldown Destination for the new Drilldown Object:

Name: <new matrix type>

Field: Type - LIC

Value: <new matrix type LIC value from LOV>

Destination Drilldown Object: <name of drilldown object created in previous step>

9. Add the new view to the Administration - Pricing screen:

Name: <name of the new view>

View: <name of the new view>

Type: Details view

Parent Category: Adjustment Group Details

11 Creating Attribute Adjustments

Creating Attribute Adjustments

This chapter describes how to create attribute pricing adjustments and includes the following topics:

- *About Attribute Pricing*
- *Attribute Adjustments Design*
- *About the Attribute Adjustments View*
- *Using Variable Maps in Attribute Adjustments*
- *How Attribute Adjustments Apply Rules*
- *Single and Multiple Type Attribute Adjustments*
- *Process of Creating an Attribute Adjustment*
- *Example of Creating an Attribute Adjustment Based on Product Attributes*
- *Example of Creating an Attribute Adjustment with Nonproduct Attributes*
- *Guidelines for Creating Attribute Adjustments*
- *How the Pricing Procedure Uses Attribute Adjustments*
- *Upgrade Considerations for Attribute Adjustments*

About Attribute Pricing

Users may customize products by selecting attributes of the product, such as size and color. Attribute-based pricing adjusts the price based on the attributes chosen, and displays the adjusted price in the quote or order.

For example, a sales agent might create a quote for a car and select the leather upholstery attribute. After selecting the leather upholstery attribute, the agent sees the net price increase by $1000, the cost of this attribute. A home shopper using a Web site might select the same car model, but select the plush interior attribute. This shopper sees the net price increase by $500, the cost of this attribute. These two different users might be using different price lists, so their attribute adjustments might be different.

In earlier versions, you used attribute-based pricing only to define pricing based on attributes of a product. Effective in version 7.8, the Attribute Adjustments view extends and generalizes the earlier attribute-based pricing feature, so it can also be used for defining price adjustments based on other criteria (such as Status of an Account, Age of an Asset, and so on) in addition to product attributes.

To create attribute adjustments, first you define a list of attributes that affect the price and the adjustments needed for each. Then, you associate the attribute adjustments with the product in a price list. Because the product can be in multiple price lists, you can create multiple attribute adjustments for the product that are used in different price lists.

This chapter assumes that you have general background knowledge about attribute-based products, classes, and attribute values. For more information, see *Siebel Product Administration Guide* . This chapter also assumes that you have general background knowledge about how the PSP engine works. For more information, see *Siebel Order Management Infrastructure Guide* .

Note: If you use attribute-based pricing with other types of price adjustments, then the final price depends on the order in which Siebel Pricer applies the pricing adjustments. For more information, see *About the Processing Order of Price Adjustments*.

Attribute Adjustments Design

To plan attribute adjustments, begin by identifying the products, the product attributes that affect prices, and any other attributes (such as Status of an Account) for which you need to define price adjustments.

Determine which attributes you want to use for pricing. Be aware that you can use nonproduct attributes, such as Status of an Account, to set up price adjustments. Some product attributes that are necessary for customization have no effect on pricing. For example, some products may have a color attribute, because the user can select the color, but the color does not affect the price. You would not use the color attribute in the attribute adjustments table.

About the Attribute Adjustments View

You use the Attribute Adjustments view to create a list of attribute adjustments, which can be based on both product attributes and nonproduct attributes.

For example, the following table shows a simple list of attribute adjustments that adjust shirt prices based on size and color of the shirt. Any shirt ordered that is XL size and not colored red is marked up by 10%. Any shirt that is L size and not colored white is marked up by 8%. Any shirt that is M size and not colored blue is marked up by 10%.

Size	Color	Adjustment Type	Adjustment Amount
= XL	<> Red	% Markup	10
= L	<> White	% Markup	8
= M	<> Blue	% Markup	5

The following table shows a simple example that adjusts shirt prices based on the size of the shirt and the status of the account. If the shirt ordered is XL size and the customer is a Gold account, then there is a 10% discount. If the shirt ordered is L size and the customer is a Silver account, then there is a 5% discount. If the shirt ordered is M size and the customer is a Bronze account, then there is a 3% discount.

Size	Customer Status	Adjustment Type	Adjustment Amount
= XL	= Gold	% Discount	10
= L	= Silver	% Discount	5

Size	Customer Status	Adjustment Type	Adjustment Amount
= M	= Bronze	% Discount	3

Dimensions in Attribute Adjustments

The columns of these tables are called the *dimensions* of the attribute adjustment. The examples show that attribute adjustments have two types of dimensions:

- **Conditions.** These dimensions affect the price of the product and determine whether the rule is applied. In the example, the condition dimensions are the columns Size, Color, and Status. Conditions can be based on any text, number, or date field in any Siebel business object, but if a rule uses the greater than (>) or less than (<) comparison operator, then its condition must be a number or date field (otherwise, the results will be unpredictable).
- **Results.** These dimensions determine the result when the rule is applied. In the example, the results dimensions are the Adjustment Type and Adjustment Amount columns.

Navigate to the Attribute Adjustments screen, then the Dimensions view to specify these dimensions.

Variable Maps in Attribute Adjustments

Variable maps are used to map the dimensions of attribute adjustments to fields in Siebel business objects. Every condition dimension of an attribute adjustment table has to be mapped to a variable map. Variable maps are provided for use with attribute adjustments. You can modify these variable maps or set up new variable maps. For information about working with variable maps, see *Siebel Order Management Infrastructure Guide* .

The variable maps are transparent to the pricing administrator who defines the price adjustments based on attributes. The order management administrator sets up all the appropriate variable maps for use in pricing.

Without configuration, the view to associate the variable maps with an attribute adjustment table is hidden. You can display the variable maps associated to an attribute adjustment table only by using the site map to navigate to the Administration - Pricing, Attribute Adjustments, then Variable Maps view. For more information about how to use variable maps in attribute adjustments, see *Using Variable Maps in Attribute Adjustments*.

Rules in Attribute Adjustments

The Rules view contains a list of conditions and results similar to the samples shown in the tables in the topic *About the Attribute Adjustments View*. You can enter rules in this view in two ways:

- **Enter rules manually.** Add a new record and enter the information for each rule manually.
- **Generate rules.** Click the Generate Rules button. The rules list is populated with records for every possible combination of condition dimensions and domains entered in the Dimensions view. You enter the results dimensions for each record. If you do not need all the combinations that are generated, then you can edit or delete some of the generated records.

You enter the list of rules in the Attribute Adjustments, Rules view.

Variable Maps in Attribute Adjustments

You use variable maps to map the dimensions of attribute adjustments to fields in Siebel business objects. For information about variable maps, see *Siebel Order Management Infrastructure Guide* .

There are three types of variable maps:

- **Context.** This type maps the variable to header-level information, for example, to fields in the Order Header applet such as Account, Order Status, and so on.
- **Row set.** This type maps the variable to line item-level information, for example, to fields in the line items of an order such as Quantity, Product Line, Price Type, and so on.
- **XA.** This type maps the variable to the attribute names and values of a line item.

For attribute pricing, you need only the standard variable maps that are provided with the product.

The Siebel application provides standard variable maps for all of these variable map types. The variable maps that attribute adjustments use are declared in the Systems Preferences view as shown in the following table.

System Preference Variable	Value
PSP Pricing Var Map - Context	Default Pricing Variable Map - Context
PSP Pricing Var Map - Row Set	Default Pricing Variable Map - Row Set
PSP Pricing Var Map - XA	Default Pricing Variable Map - XA

While setting up an Attribute Adjustment table, the pricing administrators can associate the variable maps shown in the previous table by clicking the Create Result Dimensions button. The three variable maps are created for the attribute adjustment in the Attribute Adjustments, Variable Maps view along with the two pricing results dimensions, Adjustment Type and Adjustment Amount. The system preferences, as shown in the previous table, are read and a record is inserted for each variable map type in the System Preferences view.

> **CAUTION:** If the system preference value is either empty, nonexistent, or invalid, then that variable map record is not created. No error message appears. If a user has already created the variable map records, then the application does not overwrite them.

The use of variable maps for attribute adjustments is transparent to the pricing administrator. The Variable Maps view tab does not appear in the user interface, but you can still access it using the Site Map to navigate to the Administration - Pricing screen, Attribute Adjustments, then Variable Map view.

You can set up variable maps in either of the following ways:

- *Using System Preferences to Set Up Variable Maps for Pricing Adjustments*
- *Using the Variable Maps View to Set Up Variable Maps for Pricing Adjustments*

Using System Preferences to Set Up Variable Maps for Pricing Adjustments

Use the following procedure to set up variable maps for pricing adjustments by modifying system preferences

To set up variable maps for pricing adjustments using the System Preferences view

1. Navigate to the Administration - Application screen, then System Preferences view.
2. In the System Preferences list, query for: PSP Pricing Var Map*

   Three records appear, as detailed in the table in *Using Variable Maps in Attribute Adjustments*.
3. Set the appropriate variable map names in the System Preference Value field.
4. Restart the server.

   New attribute adjustments created after the server is restarted use the new variable maps declared in system preferences.

Using the Variable Maps View to Set Up Variable Maps for Pricing Adjustments

Use the following procedure to set up variable maps for pricing adjustments by overwriting the variable maps in the Variable Map view.

To set up variable maps for pricing adjustments using the Variable Maps view

1. Navigate to the Administration - Pricing screen , Attribute Adjustments, then Variable Map view.
2. Modify variable maps by selecting different variable maps in the picklists.

The first approach is a global level change, and any new attribute adjustment table created after the server is restarted uses the new variable maps. The second approach is a local level change, and only the attribute adjustment that is modified uses the new variable maps.

How Attribute Adjustments Apply Rules

When you set up attribute adjustments, a search specification template is dynamically generated based on the definition of the condition dimensions of the attribute adjustment.

The following properties of a condition dimension are important for generating the search specification template:

- **Operator.** This property specifies the type of comparison performed against the column during run time.
- **Sort Order.** This property specifies the sorting priority of the columns in the search specification.
- **Sort Method.** This property specifies ascending or descending ordering of the search specification.

At run time, this search specification template, along with run-time contextual values, executes a lookup against the rules of the attribute adjustments, as defined in the Conditions view.

For example, shirts have two attributes that affect their price, color and size. You enter the condition dimensions shown in the following table.

Name	Operator	Variable Map	Data Type	Sort Order	Sort Method
Size	<	XA	Number	2	Descending
Color	=	XA	Text	1	Ascending

Based on these dimensions, at run time, the application generates and caches the search specification:

```
[Matrix.Color] = {Row.Color} AND [Matrix.Size] < {Row.Length}, Order by
[Matrix.Color] Ascending, [Matrix.Size] Descending
```

The notation [Matrix.<Column>] refers to a condition dimension of an attribute adjustment. {Row.<Var Name>} is a contextual value, generated at run time. You also enter the rules shown in the following table.

Sequence	Color	Size	Adjustment Type	Adjustment Amount
1	Red	15	% Discount	15
2	Blue	15	% Discount	15
3	Green	15	% Discount	15
4	Red	10	% Discount	25
5	Blue	10	% Discount	25
6	Green	10	% Discount	25

In run time, when there are incoming transactions, the application uses the search specification template with the contextual values from the specific transaction to see which rule applies. For example:

- If the transaction has Color =Red and Size = 11, then the application generates the search specification

  ```
  [Matrix.Color] = "Red" AND [Matrix.Size] < 11.
  ```

  This search specification matches the rule for Color = Red and Size = 10, so the discount of 25% is applied to the transaction.
- If the transaction has Color =Red and Size = 16, then the application generates the search specification

  ```
  [Matrix.Color] = "Red" AND [Matrix.Size] < 16.
  ```

This search specification matches the rule for Color = Red and Size = 15, so the discount of 15% is applied to the transaction.

- If the transaction has Color =Red and Size = 8, then the application generates the search specification

```
[Matrix.Color] = "Red" AND [Matrix.Size] < 8.
```

This search specification does not match any rule, so no discount is applied to the transaction.

Single and Multiple Type Attribute Adjustments

After determining which attributes affect pricing, you must decide whether you need a single-attribute or multiple-attribute pricing adjustment table.

When you create an Attribute Adjustment record, you use the Type field to specify that it is a single-condition or a multiple-condition adjustment as follows:

- **Single condition.** Use this type if the price adjustments are based on condition dimensions that apply in isolation from each other. For example, a product with the attributes Size and Color could have a price adjustment for each of these individual attributes and not for a combination of both. In this case, set up a single type attribute adjustment table.
- **Multiple condition**. Use this type if the combination of condition dimensions affect pricing, for example, if the product size and its color together determine the price adjustment.

Single-condition attribute adjustments used for attribute pricing are relatively simple to create if you need them. For information, see *Process of Creating an Attribute Adjustment*.

> **Note:** The Price Override adjustment type is not available for single dimension attribute adjustments, so use the multiple dimension attribute adjustment option to apply just one adjustment as a price override.

Before you begin creating multiple-condition attribute adjustments, decide on the following fields:

- **Sort order.** You must use this field to specify the order in which attributes (condition dimensions) are processed, because the order affects the adjusted price. For more information, see *Order of Conditions in Multiple-Condition Pricing Tables*.
- **Allow any.** You must decide whether to use this field to allow you to insert null value wildcards into the pricing table, so one record represents many values of an attribute (condition dimension). For more information, see *About Wildcards in Attribute Adjustments*.
- **Display error.** You must decide whether to use this field to prevent the user from choosing an invalid combination of condition dimensions. For more information, see *Preventing Users from Selecting Invalid Combinations of Attributes*.

Order of Conditions in Multiple-Condition Pricing Tables

If you set up a multiple type Attribute Adjustment record, then you must consider the processing order of the attributes (condition dimensions).

You can use the Sort Order and Sort Method fields of the Dimensions record to specify the order in which attributes (condition dimensions) are processed. The application assigns the dimension with the least numerical sort order value

the highest processing priority while performing a lookup against the rules (the Conditions view). The sort method determines whether the values in the dimension are processed in ascending or descending order.

Processing order is important because:

- The order of attributes (condition dimensions) affects the lookup of the price adjustment.
- If there are multiple matches of rules (Conditions view), then the order of the attributes (condition dimensions) determines which rule is applied.
- The order of attributes (condition dimensions) affects processing speed.

Order of Conditions and Pricing Calculations

In a single-type attribute adjustment, each attribute (condition dimension) contributes to the price adjustment in isolation from the rest of the other attributes (condition dimensions). This means that the sort order of the attributes does not impact the price in any way.

Assume you are using a single-type attribute adjustment for attribute pricing. A product has a list price of $100 with a 50 percent markup on the large size and a $10 markup for the color gold. The application calculates the price of the product by considering each attribute adjustment in isolation, so the price of the product is $100 + (50% of $100 for the size large) + $10 (markup for gold color) = $160. The application does not consider the sequence in which the adjustments need to be applied.

Order of Conditions and Precedence

In addition, the selection order also determines the relative importance of attributes in case of ties.

If two or more rules in an attribute adjustment are both valid, then the rule with the more explicitly defined attribute values (not blanks) in the first few columns takes precedence. For example, if there are four fields for each record, and the first two fields of both records match what has been selected, then the engine returns the record that matches the first condition going from first to last. The dimension with least numerical sort order value is processed first.

Order of Conditions and Processing Speed

The left-most condition in each rule has the greatest impact on processing speed. If it does not make a difference in the final price, then follow these general rules to speed processing:

- Give the lowest sort orders to conditions with the most enumerated values.
- Do not give a condition a low sort order if it is flagged as Allow Any.
- Do not give conditions a low sort order if you allow wildcards in the condition.

CAUTION: Pricing administrators are responsible for correctly setting up the sort orders of the dimensions. Any duplicate or blank sort order values for condition dimensions with the Allow Any flag turned on can give unpredictable results.

About Wildcards in Attribute Adjustments

Selecting the Allow Any check box in a Dimension record lets you enter a blank instead of entering every different domain value for that dimension. This blank is called a *null value wildcard.* It represents all possible values in the given attribute's domain.

For example, you sell T-shirts that come in the colors white, gray, and red, and also come in the sizes small, medium, large, and extra-large, with the pricing shown in the following table.

Color	Size	Adjustment Type	Adjustment Amount
Red	Small	Markup Amount	$2.00
Red	Medium	Markup Amount	$2.00
Red	Large	Markup Amount	$2.00
Red	Extra-Large	Markup Amount	$5.00

There is no markup for any color except red. The red T-shirts are marked up $2, except for the extra-large size, which is marked up $5. Instead of the long list shown in the previous table, you can represent this with the short list shown in the following table.

Color	Size	Adjustment Type	Adjustment Amount
Red	Not applicable	Markup Amount	$2.00
Red	Extra Large	Markup Amount	$5.00

In the previous table, the first row represents all the colors except red, and the second row represents all the sizes of red except extra-large.

The null value wildcard can reduce the work of creating a large attribute pricing table, but be careful to avoid the problems that it can create:

- If you are using the attribute adjustment for attribute pricing, then you may mistakenly enter wildcards that allow the user to choose invalid combinations of products.
- You may also mistakenly create several rules that represent the same combination of condition dimensions. Make sure that every combination of condition tables is unique.

If you do create several rules that represent the same combination of condition dimensions, then the application uses the rule with the most exact matches (not blanks) in the condition dimensions as the highest priority for processing. Priority is specified using the Sort Order field of the dimension.

Avoid using null value wildcards in the columns with the highest priority for processing. This increases the processing task for Siebel Pricer and causes the processing to take longer.

Note: If you use wildcards, then it is especially important to test your results after creating the attribute pricing table.

Preventing Users from Selecting Invalid Combinations of Attributes

You can select the Display Error check box in the Attribute Adjustment record to prevent users from selecting combinations of attributes that are not valid. Use this field judiciously, and not as a mechanism to enforce product configuration or other business rules. Attribute adjustments are primarily meant for pricing.

For example, if a product's large version comes in every color except green, you could use this option to display an error message if the user chooses both large and green.

If you select this option, then you must create a Rule record in the attribute adjustment representing every valid combination of attributes, even if the combination has zero effect on pricing. The application displays an error message if the user selects a combination of attributes that is not represented by a rule. For example, even if gold is the only color that requires a price markup, you will have to add Rule records for all the colors that the user can select. If there are 100 combinations of other attributes, and 10 possible colors, then you may have to create 1000 Rule records, even if only one color affects pricing (though you may be able to reduce the number using wildcards).

Note: If your deployment does not use the Display Error option, then you can remove the steps in the Dynamic Pricing Procedure that enforce the validity of the attribute combinations. These are the Invalid Combination Decision step and the Display Invalid Combination Error step. Removing these steps reduces processing time for the pricing. For more information about this pricing procedure, see *Pricing Procedures and Workflow References*.

To decide whether to use this option, weigh its benefits against its costs. There is a benefit to preventing users from choosing invalid combinations of attributes. However, the added administrative cost is high because there are many combinations of attributes that do not affect pricing.

If you select the Display Error option, then you must create a pricing adjustment item representing every valid combination of attribute values, but you must not create a pricing adjustment item for any invalid combination of attribute values. To create the correct pricing adjustment items, use one of the following two features of Siebel Pricer:

- **Wildcards.** If you select the Allow Any check box, then you can create adjustment items with null value wildcards that accommodate many dimension domain combinations, so you can use fewer adjustment items to represent all valid combinations. Wildcards may allow you to create rules representing every valid combination of attributes without entering a separate rule for each combination of attributes. For more information, see *About Wildcards in Attribute Adjustments*.
- **Generate Rules button in the Conditions view.** Use this button to automatically generate a full set of records for all possible combinations of values of attributes that affect pricing. You can delete the records that do not impact pricing and those that represent an invalid combination of attributes. After Rule records are generated, you must enter the adjustments. If any combinations of attributes are invalid, then you must delete those rules. For more information, see *Adding Attribute Adjustment Conditions*.

Note: If you select the Display Error option, then make sure that the default attributes for the product are a valid attribute pricing combination. If it is not a valid combination, then your Siebel application will not add the product when users click the Add Item button to add this product to a quote. Users will think that the Add Item button is not working.

It may be easier to use the Adjustment Item Generator to generate all possible attribute value combinations, each in its own adjustment item, than to use null value wildcards. To help make the decision, consider the following:

- If you use the Adjustment Item Generator, then you must do the extra administrative work of entering adjustment types and adjustment values in each record. This could involve much more work than using null value wildcards, if many combinations of attributes have the same price adjustment.
- If you use null value wildcards, then you may have a difficult conceptual task of analyzing the combinations of attributes to decide where you can use wildcards.
- If you choose the Display Error option, then you cannot use a null value wildcard for an attribute if that attribute is not always available. For example, if the extra-large size is available in all colors except gray, then you cannot create an adjustment item with the null value wildcard for both color and size (as in the previous example). If you create this adjustment value, then an error message is not generated when the user chooses extra-large and gray.

Zero-Effect Adjustment Items

If you choose the Display Error option, then you must create Rule records representing every valid combination of attributes, even if they have no effect on pricing. Even if you do not choose this option, you may want to create zero-effect rule records for the following reasons:

- To make the attribute adjustments easier to maintain. If a pricing attribute is not represented in the Rules list, then it can be difficult or impossible to add it later and integrate it with the rules already in the list.
- To create a comprehensive, well-structured set of rules. If a Rules list has a structured place for everything, then it can be easier to work with, even if it has items with zero effect.

Process of Creating an Attribute Adjustment

To create attribute adjustments, perform the following tasks:

1. *Adding an Attribute Adjustment Record*
2. *Specifying Attribute Adjustment Dimensions and Domains*
3. *Adding Attribute Adjustment Conditions*
4. *Associating Attribute Adjustments with Price List Line Items*

Note: Before you can perform this process, you must set up the appropriate variable maps, as described in *Using Variable Maps in Attribute Adjustments*.

Adding an Attribute Adjustment Record

This task is a step in *Process of Creating an Attribute Adjustment*.

The Attribute Adjustment record holds general information about the attribute adjustment.

To add an attribute adjustment record

1. Navigate to the Administration - Pricing screen, then Attribute Adjustments view.
2. In the Attribute Adjustments list, add a new record and complete the necessary fields as described in the following table.

Field	Comments
Name	Enter a name for the attribute adjustment.
Class	If product attributes affect the price, then select the class where these attributes are defined. **Note:** If you are setting up attribute adjustments based on nonproduct attributes such as Account Type, Region, and so on, then leave this field blank.
Type	Select the type of the adjustment. options are as follows: o **Single.** Each attribute (condition dimension) affects the price in isolation from the others. o **Multiple.** The price adjustment depends on a combination of attributes (condition dimensions). For more information, see *Single and Multiple Type Attribute Adjustments*.
Display Error	Select this check box if you want the application to display an error message when a user selects an invalid combination of attributes for the attribute pricing. The combination is considered invalid if it is not included in the list of rules. For more information about how to use this field, see *Preventing Users from Selecting Invalid Combinations of Attributes*. **Note:** This field is used only for attribute pricing. Do not select it if you are using attribute adjustment for some advanced use instead of for attribute pricing.
Effective From and Effective To	Enter the start date and end date of the time period when this attribute adjustment is active.

Specifying Attribute Adjustment Dimensions and Domains

This task is a step in *Process of Creating an Attribute Adjustment*.

You must enter the dimensions of the attribute adjustment, which are used as the columns of the rules tables. There are two types of dimensions:

- **Condition dimensions.** These are the dimensions that affect the price of a product. They determine the conditions under which the adjustment is applied. Create one condition dimension for each attribute that impacts the price of a product.

- **Result dimensions.** These are the outputs of the attribute adjustment tables. When you click the Create Result Dimensions button, the application automatically creates the two result dimensions, Adjustment Type and Adjustment Amount.

Optionally, if you want to automatically generate rule records, then you can enter the domain of each dimension, which is all the possible values of that dimension.

Dimension domains can have multiple sources:

- **List of values (LOV).** Enter the LOV Name.
- **Business component field values.** Enter the Business Object, Business Component, Field, and Search Specification.
- **User defined.** Manually enter all the values in the domain.
- **Attribute values.** Enter the name of the attribute.

When you define domains, you also specify whether values other than those included in the domain can be added to the rules list.

CAUTION: If you change the source of the dimension by modifying the variable map type, then the Domain Type field, or making modifications in some other way, you must click the Create Domain button to regenerate the domain, and you must modify the rules (Conditions view) that have been set up so that the old domain values for a dimension are no longer used.

To specify attribute adjustment dimensions and their domains

1. Navigate to the Administration - Pricing screen, then Attribute Adjustments view.
2. In the Attribute Adjustments list, select the attribute adjustment whose dimensions you are specifying.
3. In the More Info view, click Create Result Dimensions.
4. Click the Dimensions view tab.
5. Add a record for each dimension that affects the price and complete the necessary fields, as described in the following table.

   Note: The Adjustment Amount and Adjustment Type dimensions were added automatically when you clicked Create Result Dimensions.

Field	Comments
Dimension Type	Select the type of the dimension. The options are Condition and Result, as described in *Dimensions in Attribute Adjustments*.
Name	Enter the name of the dimension to specify the column name in the Rules list. **Note:** The name of the dimension can be different from the field name it maps to. For example, the dimension Product Color can map to the product attribute Color.
Operator	If this is a condition dimension, then select the operator that you want to use to compare it with a value.

Field	Comments
	For example, if you want to create the rule Size = XL, and if this record is the Size dimension, then you must select the equal operator (=). For more information, see *How Attribute Adjustments Apply Rules*.
Variable Map Type	Use this field for condition dimensions only. Select the type of the variable map, which determines the hierarchical level (header, line item or XA) of the contextual variable: o To map a dimension based on a field from Order Header, such as Order Status, use Context as the type. o To map a dimension based on a Line Item such as Product Line, use Row as the type. o To map dimensions based on product attributes, use XA as the type. For more information, see *Using Variable Maps in Attribute Adjustments*.
Variable Name	For dimensions with Row or Context variable map type, select the name of the variable from the variable map to map the dimension to it. Do not use for dimensions that are mapped to product attributes (variable map type - XA). For more information, see *Using Variable Maps in Attribute Adjustments*.
Data Type	Select the data type of this dimension.
Domain Type	Select the source of the domain for a dimension. The options are as follows: o **Attribute.** These are the values of an attribute that you want to use as the dimension domain. Any dimension that is associated with an XA variable map type automatically has Attribute as its domain type. o **Business component.** These are the values in a business component field that you want to use as the dimension domain. o **LOV.** These are the values in a list of values that you want to use as the dimension domain. o **User defined.** You will determine the dimension domain manually by adding new records to the Domain list.
Bounded	Select this check box to restrict the values used for this dimension in the rules. If you select this check box, then only the values listed in the domain of this dimension can be used in rules. If you do not select the check box, then any values can be used in rules.
Allow Any	Select this check box to allow any values to be entered for this dimension in rules.
Business Object, Business Component, Field, Search Spec	This field applies only if you selected Business Component in the Domain Type field. To create the domain for the dimension, specify the business component field you must use along with the Business Object, Business Component, and a Search Spec. The Business Component has the field whose values are used for the domain. The Search Spec is used to limit which values from the field you want to use as the domain. After

Field	Comments
	you enter values in these fields, click the Create Domain button to enter the values from the field in the Domain list. **Note:** To avoid poor performance, be sure to enter a Search Spec or the application may take a long time to generate all the domain records.
Attribute	If you want to base the domain on the values of an attribute, then select the attribute. After you enter an attribute here, you can click the Create Domain button to enter all the values of this attribute in the Domain list.
LOV	If you want to base the domain on the values in a list of values (LOV), then select the LOV. After you select an LOV here, you can click the Create Domain button to enter all the values of this LOV in the Domain list.
Sort Order	Enter the number specifying where this dimension appears in the rules list. For example, if you want this to be the first column in the rules list, then enter 1. This field determines the priority of this rule when the application performs a lookup against the rules table. For more information, see *How Attribute Adjustments Apply Rules*.
Sort Method	Select Ascending to display the dimensions in the order specified in the Sort Order field, or select Descending to display the dimensions in the reverse order. For more information, see *How Attribute Adjustments Apply Rules*.
Translate	This boolean flag is used for multi-lingual deployments. Out of the box, this flag is hidden in the Dimensions list and needs to be exposed if it is required for usage. This flag is used only for dimensions of the data type String that are based on domains Attribute and LOV. When checked, the application stores internally the Language Independent Code of the domain value. **Note:** Use the Translate flag only in multi-lingual deployments. Do not use for deployments that are not multi-lingual.

6. If you selected User Defined in the Domain Type field, then you must enter the domain for this dimension manually as follows:
    a. In the Dimensions list, select the record.
    b. In the Domain list, add records and enter all possible values for the selected dimension.
7. If you selected Attribute, Business Component or LOV in the Domain Type field, then to generate values for domains, click Create Domain.

Adding Attribute Adjustment Conditions

This task is a step in *Process of Creating an Attribute Adjustment*.

After you have set up the dimensions of the attribute adjustment, you can add conditions, which are the rules that state the result of the adjustment for each combination of condition dimensions.

To add attribute adjustment conditions

1. Navigate to the Administration - Pricing screen, then Attribute Adjustments view.
2. In the Attribute Adjustments list, select the attribute adjustment whose rules you are specifying.
3. Click the Conditions view tab.

   The conditions list appears, with a Name column for the name of each condition.
4. If all domain values of each dimension impact price, then you can generate rules as follows:

   a. In the Attribute Adjustment form, click Generate Rules.

      Records are added to the Rules list with every possible combination of domain values of the condition dimensions domains.
   b. Enter the Adjustment Type and Adjustment Value for each record in the Rules list.
5. If you are entering rules manually, then:

   a. In the Attribute Adjustment form, click New.
   b. In the new record, enter one possible value of the condition dimensions, and enter values in the Adjustment Type and Adjustment Amount columns to specify the result for that combination of conditions.
   c. Continue to add new records until you have entered all the necessary rules.

Associating Attribute Adjustments with Price List Line Items

This task is a step in *Process of Creating an Attribute Adjustment*.

To apply adjustments to a product, you associate the attribute adjustment with the product in a price list line item. The attribute adjustments apply to all customers who use this price list.

To associate attribute adjustments with price list line items

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the appropriate price list.
3. Click the Price List Line Items view tab.
4. In the Price List Line Items list, find the record for the appropriate price list and in the Attribute Adjustments field, select the attribute adjustment.

Example of Creating an Attribute Adjustment Based on Product Attributes

This topic gives one example of how to create attribute-based pricing based on product attributes. You may use this feature differently, depending on your business model.

In this example, adjustments are applied to the price of automobiles based on two attributes:

- Interior color has the possible values red, green, and blue.
- Transmission has the possible values automatic and manual.

Because of current market and inventory conditions, you want to create the three price adjustments shown in the following table. These price adjustments will apply only to vans sold to retail customers.

Interior Color	Transmission	Adjustment Type	Adjustment Amount
Green	Automatic	% Discount	$10
Red	Manual	% Markup	$10
Blue	Manual	Price Override	$22,000

This example assumes that you have already created the class Horizon Autos, the product Van, and the price list Retail. For information about creating classes and products, see *Siebel Product Administration Guide* . For information about creating price lists, see *Creating and Assigning Price Lists* This example also assumes that you have set up variable maps, as described in *Using Variable Maps in Attribute Adjustments*.

Creating the Attribute Adjustment Record

First, create the Attribute Adjustment record. The type must be multiple, because the price adjustments depend on more than one attribute. It must be associated with the class Horizon Autos, which includes these attributes.

To create the attribute adjustment record

1. Navigate to the Administration - Pricing screen, then Attribute Adjustments view.
2. In the Attribute Adjustments list, add a new record and complete the necessary fields as described in the following table.

Field	Value
Name	Horizon Autos Retail Van Adjustments
Class	Horizon Autos

Field	Value
Type	Multiple Dimension

Specifying Dimensions and Domains

Next, set up the dimensions that you want to use in the attribute adjustments. You must add two condition dimensions for the attributes that are the basis of price adjustments (Interior Color and Transmission).

You must also add two result dimensions that determine the amount of the price adjustment (Adjustment Amount and Adjustment Type), which are added automatically when you click Create Result Dimensions.

You also enter the domains for each of these dimensions, and you select the Bounded check box in the Dimension records. This controls data entry in the Rules list, so only valid values can be entered in the columns for these dimensions.

Use the following procedure to specify the result dimensions and their domains.

To specify the result dimensions and their domains

1. In the Attribute Adjustments list, click Create Result Dimensions.
2. Click the Pricing Dimensions view tab.

   Records have already been added to the Dimensions list for the two result dimensions, Adjustment Amount and Adjustment Type.
3. In the Pricing Dimensions list, select the record for adjustment type, and click Create Domain.

   Records are added to the Domain list for the five possible values of adjustment type.

Specifying Condition Dimensions and Domains

Use the following procedure to specify the condition dimensions and their domains.

To specify the condition dimensions and their domains

1. Add a record to the Dimensions list to represent the Interior Color attribute and complete the necessary fields, as described in the following table.

Field	Value
Dimension Type	Condition
Name	Int Color
Operator	=

Field	Value
Variable Map Type	XA
Data Type	Text
Domain Type	Attribute
Bounded	Y
Attribute	Interior Color
Sort Order	1
Sort Method	Ascending

2. Click Create Domain.

   The three values for this domain (red, blue, and green) appear in the Domain list.
3. Add a record to the Dimensions list to represent the Transmission attribute and complete the necessary fields, as described in the following table.

Field	Value
Dimension Type	Condition
Name	Transmission
Operator	=
Variable Map Type	XA
Data Type	Text
Domain Type	Attribute
Bounded	Y

Field	Value
Attribute	Transmission
Sort Order	2
Sort Method	Ascending

4. Click Create Domain.

   The two values for this domain (automatic and manual) appear in the Domain list.

Adding Pricing Conditions

After you have set up the dimensions of the attribute adjustment, you can add conditions. The Conditions list has a column for each of the dimensions that you created.

Because you are creating conditions for only three price adjustments, you add the Conditions records manually instead of using the Generate Rules button.

To add pricing conditions

1. Click the Conditions view tab.
2. Add a record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	1
Int Color	Green
Transmission	Automatic
Adjustment Type	% Discount
Adjustment Amount	10

3. Add a second record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	2
Int Color	Red
Transmission	Manual
Adjustment Type	% Markup
Adjustment Amount	10

4. Add a third record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	3
Int Color	Blue
Transmission	Manual
Adjustment Type	Price Override
Adjustment Amount	22,000

Associating the Attribute Adjustment with a Product and Price List

After you have created the attribute pricing conditions, you must associate them with a product on a price list to make them effective. These conditions apply to vans sold to retail customers.

To associate the attribute adjustment with a product and price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the record for the price list named Retail.
3. Click the Price List Line Items view tab.

4. In the Price List Line Items list, in the record for the product named Van, in the Attribute Adjustment field, select Horizon Autos Retail Van Adjustments.

Example of Creating an Attribute Adjustment with Nonproduct Attributes

This topic gives one example of how to create attribute-based pricing with nonproduct attributes. You may use this feature differently, depending on your business model.

In this example, adjustments are applied to the price of automobiles based on three attributes in three different business components:

- **Transmission.** This is a product attribute that has the possible values Automatic and Manual.
- **Account Type.** This is a header field that has the possible values Customer, Commercial, and Partner.
- **Quantity**. This is a line item field that can have any numeric value.

This example creates three possible adjustments based on these attributes:

- All vans (a product) ordered for a commercial account with manual transmission and a quantity greater than 20 will be priced at $18,000 (price override).
- All vans ordered for a customer account with manual transmission and a quantity greater than 15 will be marked up by 10% (% Markup).
- All vans ordered for a partner account with automatic transmission and a quantity greater than 10 will be discounted by 10% (% Discount).

All these discounts apply to the price list named Americas Price List.

This example assumes that you have already created the class Horizon Autos, the product Van, and the price list Americas Price List. For information about creating classes and products, see *Siebel Product Administration Guide* . For information about creating price lists, see *Creating and Assigning Price Lists*.

Modifying the Variable Map

First, you must modify the variable map Default Pricing Variable Map - Context, so Account Type is defined as a variable in this map. Do this only if Account Type is not already defined as a variable in the map.

To modify the variable map

1. Navigate to the Administration - Order Management screen, then Variable Maps view.
2. Select and lock the variable map Default Pricing Variable Map - Context.
3. In the Versions list, click Workspace.
   Details of the new version of the variable map appear.
4. In the Variable Definitions list, add a new record and complete the necessary fields as described in the following table.

Field	Value
Variable Name	Account Type

Field	Value
In/Out	In
Type	Text
On Null	Ignore

5. In the Variable Sources list, specify the source of the new variable, as shown in the following table.

Field	Value
Mode	Any
Path	$Current/Header/Account Type
Source Type	Instance

6. Restart the client application to make sure that the variable map changes are available.

Creating the Attribute Adjustment Record

The type of the Attribute Adjustment record must be multiple, because the price adjustments depend on more than one attribute. The type must be associated with the class Horizon Autos, which includes these attributes.

To create the attribute adjustment record

1. Navigate to the Administration - Pricing screen, then Attribute Adjustments view.
2. In the Attribute Adjustments list, add a new record and complete the necessary fields, as described in the following table.

Field	Value
Name	Horizon Autos
Class	Horizon Autos
Type	Multiple Dimension

Specifying Dimensions and Domains

Next, set up the dimensions that you want to use in the attribute adjustments. You must add three condition dimensions for the attributes that are the basis of price adjustments (Transmission, Account Type, and Quantity).

Also, you must add two result dimensions that determine the amount of the price adjustment (Adjustment Amount and Adjustment Type), which are added automatically when you click Create Result Dimensions.

You also enter the domains for each of these dimensions. Select the Bounded check box in the Dimension records to control data entry in the Rules list, so only valid values can be entered in the columns for these dimensions.

To specify the result dimensions and their domains

1. In the Attribute Adjustments list, click Create Result Dimensions.
2. Click the Pricing Dimensions view tab.

   Records have already been added to the Dimensions list for the two result dimensions, Adjustment Amount and Adjustment Type.
3. Add a record to the Dimensions list to represent the Transmission attribute and complete the necessary fields, as described in the following table.

Field	Value
Dimension Type	Condition
Name	Transmission
Operator	=
Variable Map Type	XA
Data Type	Text
Domain Type	Attribute
Bounded	Y
Attribute	Transmission
Sort Order	1
Sort Method	Ascending

Field	Value

4. Click Create Domain.

   The two values for this domain (automatic and manual) appear in the Domain list.
5. Add a record to the Dimensions list to represent the Account Type attribute and complete the necessary fields, as described in the following table.

Field	Value
Dimension Type	Condition
Name	Act Type
Operator	=
Variable Map Type	Context
Data Type	Text
Domain Type	Business Component
Bounded	Y
Business Object	List of Values
Sort Order	2
Sort Method	Ascending
Business Component	List of Values
Field	Value
Search Spec	[Type] = “ACCOUNT_TYPE"

> **Note:** In this example, Business Component is used as the domain for setting up the Act Type dimension. However, a better way to define the domain in practice would be to use LOV with the value ACCOUNT_TYPE in the LOV field.

6. Click Create Domain.

   The values for this domain appear in the Domain list.
7. Add a record to the Dimensions list to represent the Quantity attribute and complete the necessary fields, as described in the following table.

Field	Value
Dimension Type	Condition
Name	Quantity
Operator	<
Variable Map Type	Row
Data Type	Number
Domain Type	User Defined
Bounded	Y
Sort Order	3
Sort Method	Ascending

8. Because Quantity has a user-defined domain, create its domain manually by adding records to the Domains list as follows:
   a. Add a record to the Domains list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	1
Domain	10

b. Add another record to the Domains list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	2
Domain	15

c. Add another record to the Domains list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	3
Domain	20

Adding Pricing Conditions

After you have set up the dimensions of the attribute adjustment, you can add conditions. The Conditions list has a column for each of the dimensions that you created.

In this example, you add the Conditions records manually instead of using the Generate Rules button.

To add pricing conditions

1. Click the Conditions view tab.
2. Add a record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	1
Transmission	Manual
Act Type [=]	Commercial
Quantity [<]	20

Field	Value
Adjustment Type	Price Override
Adjustment Amount	18,000

3. Add a second record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	2
Transmission	Manual
Act Type [=]	Customer
Quantity [<]	15
Adjustment Type	% Markup
Adjustment Amount	10

4. Add a third record to the Conditions list and complete the necessary fields, as described in the following table.

Field	Value
Sequence	3
Transmission	Automatic
Act Type [=]	Partner
Quantity [<]	10
Adjustment Type	% Discount
Adjustment Amount	10

Associating the Attribute Adjustment with a Product and Price List

After you have completed setting up the attribute adjustments, you must associate them with a product on a price list to make them effective. These conditions apply to the product named Van and to the price list named Americas Price List.

To associate the attribute adjustment with a product and price list

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the record for the price list named Americas Price List.
3. Click the Price List Line Items view tab.
4. In the Price List Line Items list, find the record for the product named Van and in the Attribute Adjustment field, select Horizon Autos.

Testing the Attribute Adjustments

You can test the attribute adjustments for three different account types. This example assumes that you have created the following accounts:

- **ABC.** Account Type = Customer.
- **XYZ.** Account Type = Commercial.
- **123.** Account Type = Partner.

To test the attribute adjustments

1. In the Quotes screen, create a new quote with price list Americas Price List and account ABC (Account Type = Customer).
2. Add the product Van as a line item of the quote.
3. Set the attribute Transmission to Manual as follows:
    a. Click Customize to launch Configurator.
    b. In Configurator, set the attribute Transmission to Manual.
    c. Click Done to exit from Configurator.
4. For the quantity of the product Van, enter 16.

    The net price of the product Van is marked up by 10%.
5. Click the Net Price field to display the pricing waterfall, to view this adjustment.
6. Test the second attribute adjustment as follows:
    a. In the Quote, change the account to XYZ (Account Type = Commercial).
    b. In the line item, change the product quantity to 21.

        The net price of the product Van is $18,000.
7. Test the third attribute adjustment as follows:
    a. In the Quote, change the account to 123 (Account Type = Partner).

b. Click Customize and use Configurator to change the value of the attribute Transmission to Automatic.
c. In the line item, change the product quantity to 11.

The net price of the product Van is discounted by 10%.

Guidelines for Creating Attribute Adjustments

This topic provides guidelines for creating attribute adjustments. Use attribute adjustments in cases where:

- The values of the variables that are used to determine the results are not known ahead of time.
- Integrations to external applications are secondary criteria.
- There is a small number of total rules, in the thousands at the most.

Use Sort Order to Improve Performance

You can use the Sort Order and Sort Method fields of the Dimensions record to specify the order in which attributes are processed. For information about how this affects performance, see *Order of Conditions in Multiple-Condition Pricing Tables*.

Use Nonequality Operators to Improve Performance

You can reduce the number of rules required by using the nonequality comparison operators to limit the size of the rules table, when possible. For example, if any shirt that is of length 40 or smaller receives a 10% discount, then you can set up a condition dimension for the Length attribute with greater than (>) operator to set up the price adjustment rule. For more information, see *Adding Attribute Adjustment Conditions*.

> **CAUTION:** Make sure that the attribute adjustment is set up so that the attribute adjustment returns only one record. When nonequality operators are used, the attribute adjustment may return more than one record. Pricing administrators are responsible for setting up the dimensions in the appropriate way or setting up the pricing procedure to handle multiple matches.

Using wildcards does not improve performance, but it does decrease the number of records in the rules table.

Numeric Dimensions and Operators

If a condition dimension is compared with a numeric value using the equal (=) operator, then you can improve performance by setting its data type to text. Property sets can store only text values, and when you use a numeric comparison, there is an overhead to convert the text to numeric values at run time. You must use numeric values for less than (<) or greater than (>) comparisons, but if you are using an equal (=) comparison operator, then you will get the same result whether the data is numeric or text.

Comparison Operators

When the rules in attribute adjustments use comparison operators such as the greater than (>) and less than (<) operators:

- The data types of the condition dimensions must be number or date.
- Pay special attention to the following:
- Whether the rule must act on all the rows of data when you enter the values that the conditions must match
- The Sort Order properties of the dimensions. Blanking out or duplicating the sort order will give undefined results.
- The Sort Order and Sort Method fields of the dimensions. These fields determine the order in which the rules are processed, and you must select values that minimize the number of records that must be processed.

How the Pricing Procedure Uses Attribute Adjustments

After an attribute adjustment is set up, the pricing procedure must perform a lookup against it in the pricing procedure so that adjustments can be applied at run time. For more information about the pricing procedure, see *Pricing Procedures and Workflow References*.

The pricing procedure does this lookup using the Attribute Adjustment step, which invokes the Dynamic Lookup Transform. For more information about the Dynamic Lookup Transform, see *Siebel Order Management Infrastructure Guide* .

Upgrade Considerations for Attribute Adjustments

When you upgrade from earlier versions of Siebel applications to version 7.8 or later, use the On First Match (Action) argument of the Dynamic Lookup Transform. This argument makes attribute pricing that is defined in the Attribute Adjustments list in version 7.8 or later work the same way as attribute-based pricing feature in earlier versions.

Without configuration, the pricing procedure provides the same functionality as in the earlier release.

The Attribute Adjustments list can return multiple rows. The first row it returns will always be consistent with attribute pricing behavior in earlier versions. Using On Match arguments can return results that are not consistent with the behavior of versions earlier than 7.8.

12 Setting Up Pricing for Products with Components

Setting Up Pricing for Products with Components

This chapter describes pricing for products with components. It instructs you on viewing a product's components and setting up component-based pricing. It also covers the processing order of price adjustments for products with components. This chapter covers the following topics:

- *About Component-Based Pricing*
- *Impact of Product Versions and Price List Line Item Effectivity on Component-Based Pricing*
- *Viewing a Product's Components in the Pricing Designer*
- *About Setting Up Component-Based Pricing*
- *Viewing and Changing the Reference Price*
- *Defining a Relationship-Based Price Adjustment*
- *Verifying Component Prices Across Time Intervals*

About Component-Based Pricing

Products with components include components that are themselves products in the product table. For example, if you sell computer monitors, CD-ROMs, and disk drives, then you also sell several models of desktop computer that include these products as components.

This chapter explains how to set up pricing for products with components, as follows:

- **View a product in the Pricing Designer.** You can use the Pricing Designer to view a product's components and for the other functionality described in this chapter.
- **Set up component-based pricing.** You set up pricing so the total price of a product either equals a base price plus the price of optional components, or equals the total price of all components.
- **Create relation-based price adjustments.** You can give customers discounts if they buy components as part of a specific product package. These are called relation-based price adjustments.

This chapter assumes that you have background knowledge about products with components. For more information about products with components, see *Siebel Product Administration Guide* .

If you create several types of price adjustments for products with components, then the order in which price adjustments are processed can affect the final price. For more information, see *About the Processing Order of Price Adjustments*.

Impact of Product Versions and Price List Line Item Effectivity on Component-Based Pricing

A product can have multiple released versions with future dates. In addition, price list line items can have effective dates. As a result, the structure of a product can be dynamic over time, because multiple versions of it can exist.

Component-based pricing takes the following into account:

- Both root customizable product and its components can have multiple versions across a time interval.
- Both root customizable product and its components can have multiple entries on a price list for different effective dates.
- The Customizable Product Adjustments (The CP Pricing Designer) can have multiple versions for different effective dates.

This topic discusses some of the ways that multiple product versions and multiple effective dates affect component-based pricing.

Component Product Must Exist on the Price List of a Component-Based Product

If a component-based product does not exist on a price list as of a particular date, then the adjustments to its components are not applied as of that date. For the adjustment to apply to the component product, it must exist on the price list of the component-based product in the same time interval as the adjustment. Consider this example:

- A component-based product, product P, has only one component product, product P1.
- P and P1 have entries in the Price List Line Item list that are shown in the following table.

Product	Start Date	End Date	List Price
P	12/31/2004	04/01/2004	1000
P1	12/31/2003	01/15/2004	100

- Adjustments have been defined for P1 when it is a component of P that are shown in the following table.

Adjustment Type	Adjustment Value	Start Date	End Date
Price Override	104	02/05/2004	02/29/2004
Percent Discount	10	03/01/2004	03/31/2004

- In this case, if the product P1 is ordered in the context of the root product P on 02/15/2004, then it will have a zero price. You might expect its price to be 104, based on the price override that is effective on that date, but because product P1 does not have an entry in the price list for the interval 02/05/2004 through 02/29/2004, no price appears on the transaction for it.

This behavior is consistent with the prior releases of Customer Order Management.

Note: If a root product with components does not exist on a price list on a given date, then the adjustments to its components are not applied for that date. For example, the adjustments in the CP Pricing Designer might be effective for a specific time interval during which the root product with components might not have an entry in the price list itself. The CP adjustments to the components of the root product with components do not apply during that time interval.

Time Intervals When a Product Is Not Customizable

The price list for a product with components can include a time interval when the product is not customizable. This is possible because a product might not have any structure on a given date, but might have a structure on some other date (specified in a released version with a future date).

In this case, if the user drills down from the price list line item to display the product, then the application displays an error message saying that the customizable product does not have versions in the date range, and that the user must choose an appropriate time interval to set up price adjustments for the product. This means that the customizable product does not have structure at that time. In other words, it does not have component products.

Viewing a Product's Components in the Pricing Designer

You can use the Pricing Designer to display the relationships and components of a product with components. You can query by relationships to look at all the component products available within a specific relationship.

The CP Pricing Designer displays a superset structure of the root product with components. It displays a consolidated view of all the versions of the root product with components that might exist in the price list during a given time interval. Thus, you might see products in a relationship in Pricing Designer that do not exist in the latest released version of the root product with components. They are visible in the pricing designer for the root product with components because they exist in released version of the product with a future date. Therefore, the structure of the root product in the pricing designer may not be identical to the latest released version of the root product. The Pricing Designer is sensitive to the start and end date of the root product with components on the price list, so the content of the Pricing Designer may vary for different time intervals.

To view a product's components in the pricing designer

1. Navigate to the Administration - Pricing screen, then Price Lists view.
2. In the Price Lists list, select the price list that contains the product.
3. Click the Price List Line Items view tab to display the products in the price list.
4. In the Price List Line Items list, click the name of the root product with components.

> **Note:** The root product with components might have multiple entries on the price list for the different time intervals, so you must pick the root product with components with the appropriate time interval.

The Pricing Designer view appears.

> **Note:** Using the New Effective Date button on the Price List Line Item applet for a product with components causes a deep copy that also copies any customizable product adjustments defined for the product. Delete the adjustments that do not apply in the new time interval.

About Setting Up Component-Based Pricing

There are two ways of setting up component-based pricing:

- *Setting Up Base Pricing for Products With Components* Use this when products with components share a base configuration. Customers who purchase the base product can customize it by selecting options that are added to the base product.
- *Setting Up Summation or Rollup Pricing for Products With Components* Use this when products with components do not share a common base configuration.

Setting Up Base Pricing for Products With Components

Use this pricing method when the user purchases a base product and customizes it by adding options. The total price is the sum of the base price plus the adjusted price of the components. For example, if a car has a base price of $25,000, then the user can also purchase additional options such as air conditioning, sun roof, and so on. Use the following procedure to set up base pricing for a $25,000 car.

To set up base pricing

1. Set the list price of the car to $25,000.
2. Set the price for all other options and, if necessary, specify that some are default components.

   When users configure the product, the price they see is the sum of the base price plus the adjusted price of the default options. If the user adds other options, then the price increases. For example, assume a car has a base price of $25,000 and includes the following options:

- Chrome Wheels (Default) for $2000
- Gold Wheels for $3000

  The user sees an initial price of $27,000 for the car. This is the sum of the base price of the car plus the price of the default chrome wheels. If the user chooses the gold wheels, then the sum is recomputed and the price of the car becomes $28,000.

  An alternate method for setting up base pricing is to:

- Include options in the base price.
- Create price override adjustments for all default options included in the base price that sets their price to zero (0).

- Create a price override adjustment for related nondefault options that sets their price to the difference between the default option and the nondefault option.

  In the example, you can use this alternative method to include chrome wheels in the list price of the car by doing the following:

- Set the list price of the car to $27,000.
- Create a price override to zero (0) for the chrome wheels.
- Create a price override to $1000 for the gold wheels. This is the price difference between the gold and chrome wheels.

Setting Up Base Pricing for Products with Components

Use the following procedure to set up base pricing for products with components.

To set up base pricing for products with components

1. Display the components of the product with components in the Pricing Designer, as described in *Viewing a Product's Components in the Pricing Designer*.
2. Define relationship-based pricing adjustments, if they are needed, as described in *Viewing and Changing the Reference Price*.
3. In the Price List Item form, go to the List Price field and enter the base price.
4. In the Pricing Designer list, select the components. In the Adjustments list, enter price adjustments for each component.
5. If you are using the alternative method of setting up base pricing, then enter the necessary price overrides for default components.

Setting Up Summation or Rollup Pricing for Products With Components

Use this pricing method when all of a product's components are configurable. Although each product has a default configuration, users can select nondefault components instead of the default components.

When you use the Pricing Designer, you must set a list price for the base product of a component-based product and define pricing adjustments on components. The price that is displayed for the user is the sum of the list price and the adjusted price of the default components. This price may display to the user in a quote, agreement, order, the selection pages in a configuration session, or in a catalog.

When you are using summation (or rollup) pricing, there is no base product, so you set the list price of the base product to zero (0). This means that the price the user sees is the sum of the prices of its default components (Quote, Agreement, Order, and configuration selection pages). In an eSales catalog, the user sees a price of 0.

An example would be a desktop computer made up entirely of components that you also sell separately. To set up rollup pricing for quote users you would do the following:

- Set the list price of the desktop computer to 0.
- Set a price for each of the components, and specify which ones are default components.

Because the list price is 0, the default price of the desktop computer is the sum of the prices of its default components. If the user who is configuring the product picks a nondefault component, then the sum is recomputed to include the

price of the nondefault component. For example, for a desktop computer, you have entered a list price of 0 and the component pricing adjustments as shown in the following table.

Item	List Price	Adjustment Type	Adjusted Price
1.0 Chassis (default)	$1200	Price override	$1000
2.0 Monitor	Not applicable	Not applicable	Not applicable
2.1 Small monitor (default)	$300	Price override	$200
2.2 Large monitor	$400	Price override	$300

While configuring a quote, users see an initial price of $1200 for the desktop computer (the sum of the prices of the default chassis and monitor). If the user chooses the large monitor, then the sum is recomputed, and the price of the desktop computer becomes $1300.

Note: The price that is displayed includes any relationship-based pricing adjustments and other adjustments to the price of the components.

To set up rollup pricing for a product with components

1. Display the components of the product with components in the Pricing Designer, as described in *Viewing a Product's Components in the Pricing Designer*.
2. Define relationship-based pricing adjustments, if they are needed, as described in *Viewing and Changing the Reference Price*.
3. Do not enter a price for the root product with components in the Price List line items view.
4. Enter prices for all the components.

Viewing and Changing the Reference Price

To give customers an accurate idea of the price they might expect to pay for a component-based product, Siebel Pricer calculates a reference price, which is the price of the product with its default options. The reference price is displayed to the customer in the Starting At Price field in the product catalog.

For the Starting At Price to function as expected, child components of the product must have a minimum cardinality greater than zero(0) to be reflected in the reference price. If the minimum cardinality is not set for a child component, then its price is not reflected in the reference price. Any component that is part of the default instance is used to evaluate the Starting At Price. Out of the box, only products with the price type one-time are included in the evaluation of the Starting At Price. Products with price types recurring and usage are not included.

Note: The default instance used for evaluation is based on the most effective applicable version of the component pricing.

Siebel Pricer displays both the list price and the reference price for the component-based product. The list price is the price for the root component. If you click Check Starting At Price, then Siebel Pricer adds the list price and the price of all the default components. This calculation does not include any attribute pricing adjustments that would come from default attributes.

In the catalog, this value is displayed to customers in the Starting At field. This value is displayed only for component-based products.

To change default the reference price, click the Edit button in the first applet. You can then edit the reference price manually.

The value displayed to customers does not change unless the pricing administrator clicks the Check Starting At Price again or overwrites the reference price manually.

> **Note:** The reference price is used for products with components, including component-based products and bundles. It does not apply to attribute-based products.

Without configuration, the application executes the Basic Pricing Procedure. However, the Basic Pricing Procedure does not execute CP pricing. To be consistent, the Check Starting At Price button calls the same procedure as the run-time application (therefore, without configuration, the Basic Pricing Procedure) to determine the starting price for a customizable product. This ensures that the administrative user sees the same price as the end user.

Customers who have the Dynamic Pricer license and want to use CP Pricing need to enable the Dynamic Pricing Procedure, as detailed in the *Dynamic Pricing Procedure Workflow*. In this procedure, the customer changes the CalculatePrice_Configurator signal to call the Dynamic Pricing Procedure.

This change ensures that the administrative user will see the effect of CP Price Adjustments in the starting price that is generated by clicking the Check Starting At Price button, and that the end user will see the effect of CP Price Adjustments in the starting price of the CP and its components in the Configurator's running time.

Viewing the Reference Price

Use the following procedure to view the reference price.

To view the reference price

1. Navigate to the Administration - Pricing screen, then Price List view.
2. In the Names field, select the price list that contains the component-based products.

   The Price List Line Item list appears.
3. Click on a customizable product name.

   A customizable product has a check in the Customizable field.
4. In the Price List Item form, click Check Starting At Price.

   The reference price is recalculated based on the default components of the product, and the result displays in the Reference Price field.

Exposing the Starting Price At in the Product Catalog

Use the following enhancements to expose the Starting Price At field in the product catalog UI.

1. Workflow: In the Basic Pricing Procedure workflow and the Dynamic Pricing Procedure workflow, add the input argument from the following table to the Get List Price step.

Input Argument	Value
On First Match 15	{Row. Reference Price} = {Match. Reference Price}

   Then, activate the workflow after the change. For more information about using these workflows, see *Pricing Procedures and Workflow References*.
2. Variable Map or Variable Source: In Product Pricing Variable Map - Row Set, and Product Pricing Variable Map - Row Set-Basic, use the variable definition in the following table.

Variable Name	In/Out	Type	On Null
Reference Price	Out	Currency	Ignore

   With the new definition, use the variable sources described in the following table:

Mode	Path	Source Type
Any	$Current/Product/Reference Price	Instance
Favorite	$Current/Favorite/Reference Price	Instance

   For more information about using variable maps, see *Siebel Order Management Infrastructure Guide* .
3. Business Components: Modify the business components (buscomps) described in the following table.

Business Component	User Property	Change This Value
Quote Catalog Internal Product by Price List Option	PSP: Price Fields	From: List Price, Net Price, Deals To: List Price, Net Price, Reference Price
Quote Catalog Internal Product by Price List Optional 2	PSP:Price Fields	From: List Price, Net Price, Deals To: List Price, Net Price, Reference Price

Business Component	User Property	Change This Value
Quote Catalog Internal Product by Price List Optional - PSS	PSP:Price Fields	From: List Price, Net Price, Deals To: List Price, Net Price, Reference Price

For more information about business components, see *Siebel Order Management Infrastructure Guide* .

4. Applets: Update the applets described in the following table.

Applet	Change
Quote Catalog Product Form	Change field to Reference Price for control PriceStartingAt
Quote Catalog Product List - Add to Cart	Change field to Reference Price for list column Price Starting At
Sales Order Catalog Product List - Add to Cart	Change field to Reference Price for list column Price Starting At
Service Order Catalog Product List - Add to Cart	Change field to Reference Price for list column Price Starting At
Quote Catalog Product List - MultiSelect Checkbox	Change field to Reference Price for list column Reference Price
Sales Order Catalog Product List - MultiSelect Checkbox	Change field to Reference Price for list column Reference Price
Service Order Catalog Product List - MultiSelect Checkbox	Change field to Reference Price for list column Reference Price

For more information about working with applets, see *Using Siebel Tools* .

5. Integration Objects: Update the following objects:
   a. Integration Object: ISS Quote
      - Integration Component: Internal Product PSP Integration
      - Add Integration Component Field:
        - Name: Reference Price
        - Data Type: DTYPE_CURRENCY

- External Name: Reference Price
- XML Tag: ReferencePrice
- Integration Component: Quote Catalog PSP Favorite Item

- Add Integration Component Field:
  - Name: Reference Price
  - Data Type: DTYPE_CURRENCY
  - External Name: Reference Price
  - XML Tag: ReferencePrice

b. Integration Object: ISS Order

- Integration Component: Internal Product PSP Integration
- Add Integration Component Field:
  - Name: Reference Price
  - Data Type: DTYPE_CURRENCY
  - External Name: Reference Price
  - XML Tag: ReferencePrice
  - Integration Component: Quote Catalog PSP Favorite Item
- Add Integration Component Field:
  - Name: Reference Price
  - Data Type: DTYPE_CURRENCY
  - External Name: Reference Price
  - XML Tag: ReferencePrice

6. Business Components (Buscomp): Update the buscomps with the information provided in the following table.

Business Component	Change
Quote Catalog PSP Favorite Item	Add a calculated field Reference Price with no Calculated Value and type of DTYPE_CURRENCY.
Internal Product PSP Integration	Add a calculated field Reference Price with no Calculated Value and type of DTYPE_CURRENCY.

Defining a Relationship-Based Price Adjustment

Relationship-based price adjustments let you adjust the price of a product when it is a component of a product with components. For example, you may sell a model of computer monitor for $300 if the customer buys only the monitor.

However, if the customer who is buying one of your desktop computers chooses this monitor model as an option, then you may reduce its price 10 percent from this list price, or you may sell it for $250.

To create the relationship-based price adjustment, use the Adjustment Type and Adjustment Amount fields for the component.

To define a relationship-based pricing adjustment

1. Display the components of the product with components in the Pricing Designer, as described in *Viewing a Product's Components in the Pricing Designer*.
2. Select the records for any components that require a relationship-based pricing adjustment, add a new record to the Adjustments applet, and enter the values for the fields as described in the following table.

Field	Comments
Adjustment Type	Select the type of price adjustment applied to this product. Options are Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.
Adjustment Value	Enter the value of the Adjustment. How this value is used depends on the selection in the Adjustment Type field: o **Discount amount.** The adjustment value is subtracted from the price. o **% discount.** The adjustment value is treated as a percentage, and that percentage of the price is subtracted from the price. o **Markup amount.** The adjustment value is added to the price. o **% markup.** The adjustment value is treated as a percentage, and that percentage of the price is added to the price. o **Price override.** The adjustment value is used as the price, overriding the original price.
Min Price	Enter the minimum price for this product as a component of this customizable product. This may be different from the minimum price of the product when it is purchased separately.
Max Price	Enter the maximum price for this product as a component of this customizable product. This may be different from the maximum price of the product when it is purchased separately.
Start Date	Enter the date when this adjustment becomes effective.
End Date	Enter the date after which this adjustment is no longer effective.

3. Click Verify Adjustments to verify that the component's Reference Price reflects the pricing adjustment.

Note: You can modify the relationship-based adjustment by editing the amounts in the Adjustment Type and Adjustment Value fields for the records in the Adjustments applet, and you can delete the relationship-based adjustment by deleting the records in the Adjustments applet.

Verifying Component Prices Across Time Intervals

Products can have multiple versions with different effective dates. For more information, see *Siebel Product Administration Guide* .

Price lists can have multiple line items for the same product, with different effective dates for the price in each line item. For more information, see *Giving a Product Multiple Prices with Different Effective Dates*.

The structure of a component-based product may change during a time interval, and the prices of the product and its components may change at different dates during this time interval. It may be necessary to apply the component price adjustments to future versions of the component-based product, which has a different structure. For more information, see *Impact of Product Versions and Price List Line Item Effectivity on Component-Based Pricing*.

When you create a new time interval for a product with components in the Administration - Product screen, the component-product price adjustments are copied to the new time interval. However, these adjustments will apply in run time only if the component-product and its components exist in the price list during that same time interval.

You can click the Verify Adjustments button to display a list of the time intervals that have been defined for products with component prices. Check this list to make sure that component prices have been defined for every time interval.

To verify component prices across time intervals

1. Display the components of the product with components in the Pricing Designer, as described in *Viewing a Product's Components in the Pricing Designer*.
2. Select a component in the Pricing Designer list.
3. Click Verify Adjustments.

   The Reference Prices dialog box appears, listing the Start Date and End Date for all the prices for this product with components. For each time period, it lists the base price and price adjustments, if any.
4. Check the Reference Prices list to make sure that there are no time gaps when this product does not have a defined price.

Sample Output of Verifying Component Prices Across Time Intervals

As an example of verifying component prices across time intervals, consider a simple product with components:

- Product A is a product with components.
- Product A1 is a component of product A.

  In the price list, Product A1 has the list prices shown in the following table.

Start Date	End Date	List Price
12/31/04	6/30/05	$100
7/15/04	12/31/05	$105

Start Date	End Date	List Price

Within Product A, there are the price adjustments for Product A1 shown in the following table.

Start Date	End Date	Adjustment
1/31/05	6/25/05	10% discount
7/15/04	12/31/05	10% markup

When you click the Verify Adjustments button, the pop-up applet contains the information shown in the following table.

Start Date	End Date	Adjustment	List Price	Item Price
12/31/04	1/31/05	Not applicable	$100	$100
1/31/05	6/25/05	10% discount	$100	$90
6/25/05	6/30/05	Not applicable	$100	$100
6/30/05	7/15/05	Not applicable	Not applicable	$0
7/15/05	12/31/05	10% markup	$105	$115.50

This pop-up applet shows the gaps in the pricing information. During several time periods, there is no component price adjustment. Between 6/30/05 and 7/15/05, there is no list price, so the item has a price of zero.

13 Creating Price Reports

Creating Price Reports

This chapter describes available pricing reports and provides procedures to run, create, and modify pricing reports. This chapter covers the following topics:

- *About Pricing Reports*
- *Running Reports*
- *About Creating and Modifying Reports*

About Pricing Reports

Siebel Pricer includes a standard set of predefined reports that supply pricing administrators with fundamental pricing information. You are not limited to using these basic reports. You can create additional reports or modify existing ones.

Running Reports

While you are in a view, click Reports in the toolbar to see a list of the available preconfigured reports for the data in that view. The list specifies which reports are available as Oracle Business Intelligence Publisher reports. From the list you can access individual reports. You can run all the reports in the same way.

To run the price lists report

1. Navigate to the screen and view from which you access the report, as specified in the report's description.
2. In the application-level menu, click the Reports icon and select the appropriate report.
3. In the Reports dialog box, select a report format from the drop-down menu and click Submit.

   The selected report appears in the format specified.
4. You can print a copy of the report.

About Creating and Modifying Reports

You are not limited to working with these standard, predefined reports. The Siebel Pricer reports are part of a large set of Siebel application reports. You can modify these reports or add new reports in the following location:

- **Web Tools.** Use this location to define the data exported from the Siebel application to the report, and to attach reports to the Reports menus of specific views.

For information about defining and working with reports, see *Siebel Reports Guide* .

14 Pricing in the Run-Time Application

Pricing in the Run-Time Application

This chapter covers the ways that pricing affects the run-time application. For more information about how pricing affects quotes and orders, see *Siebel Order Management Guide* . This chapter includes the following topics:

- *Pricing Waterfalls*
- *Contract-Based Pricing*
- *Multiple Price Types and Totals*
- *About Configuring Pricing Fields in Catalog, eSales, and Configurator*
- *Multiple Currencies in a Quote, Order, or Agreement*
- *Spread Discount Pricing*
- *Totaling and Rollups of Pricing*

Pricing Waterfalls

The final net price that is charged to a customer is usually the result of many pricing adjustments. It begins with the product's list price, and then adjusts the price because of volume discounts, attribute pricing, aggregate discount pricing, manual price adjustments entered by the sales representative, and many other possible types of price adjustments.

The pricing waterfall lets the user see all the steps that went into calculating the net price. It is a list of prices that begins with the list price showing the price adjustments, in the order they were applied, with the price after each adjustment, ending with the final net price.

Pricing waterfalls include any manual price adjustments that the sales representative entered at the time of sale, as well as price adjustments applied by the pricing engine.

The end user displays a pricing waterfall by clicking the net price in any line item of a quote, order, or agreement. A pop-up pricing waterfall applet appears that lists all the steps used to calculate the net price. The fields in each row of the pricing waterfall applet are described in the following table.

Field	Comment
Adjustment	This field describes the price adjustment for this line, such as manual line item adjustment or volume discount. Alternatively, this field gives the list price or other start price and notes the price list it came from.
Type	This field displays the type of the adjustment. Examples are % Discount, Discount Amount, or Override. The start price is considered an override.
Amount	This field displays the value used by the adjustment, whose use depends on the adjustment type. For example, if the adjustment type is % Discount, and the Adjustment Amount is 15, then the adjustment

Field	Comment
	is a 15 percent discount. If the adjustment type is Discount Amount, and the Adjustment Amount is 15, then the adjustment is a discount of $15 (assuming the currency is dollars). If the adjustment type is Override, and the Adjustment Amount is 15, then the adjustment is to replace the price with a price of $15 (assuming the currency is dollars).
Price	This field displays the current price in each step of the price calculation. In the first row, this is the start price. In the last row, it is the final net price.

Configuring Pricing Waterfalls

To configure the pricing waterfall, you modify the pricing procedure used to display the waterfall. For more information about this pricing procedure and the PSP waterfall business service that it calls, see *Siebel Pricer Technical Reference* For more information about working with pricing procedures and messages, see *Siebel Order Management Infrastructure Guide* .

This topic gives you general instructions about some common ways of configuring pricing waterfalls.

Configuring a New Pricing Waterfall

Use the following procedure to add a new pricing waterfall.

To configure a new pricing waterfall

1. Create a new message type in the Administration - Application screen, Message Type Administration View as follows:
    - You must enter the name and full text of the message.
    - You can add translations for the full text if needed.
2. Specify the payload variables that will be substituted in the full text in the Message Type Payload Administration view.
3. Add an action to a step in the pricing procedure that creates the waterfall record using the new message type, passing the required payload fields.
4. Test the revised pricing procedure.

Adding a New Field to the Pricing Waterfall

You can add a new field to the information in each row of a waterfall. For example, you might want each row to include the approval level that was needed to give the discount in that row.

Use the following procedure to add a new field to the pricing waterfall.

To add a new field to the pricing waterfall

1. Edit the pricing procedure and add the new field to actions that generate waterfall output.

For example, to add the approval level as the last column of the waterfall, add it at the end of the standard actions that generate the waterfall, as follows:

```
{Row}.{Net Price Waterfall} += New('Waterfall', [Text] = LookUpMessage('Pricer - Dynamic Matrix
Adjustment', [Price Book] = {Match.Price Book}), [Adjustment Type] = {Match.Adjustment Type},
[Adjustment Amount] = {Match.Adjustment Amount}, [Currency Code] = {Row.Currency Code}, [Price] =
{Row.Start Price}, [Approval Level] = {Match.Approval Level})
```

2. Add the new field to the Net Price Waterfall VBC.
3. Add the new field to the Net Price Waterfall Pop-up List applet.

Saving Pricing Waterfall Output

In general, the pricing waterfall is generated on demand when the user clicks the Show Details button, instead of being stored in the database. When the record is repriced, the waterfall displays.

You can save the waterfall to the database at any time by selecting a line item, and then clicking the Save Waterfall button in the Line Items list of a quote, order, or agreement. You can also configure the product to invoke the SyncToDB method of the PSP waterfall business service to save the waterfall for an entire document.

However, writing the waterfall to the database can slow performance, so do it only if it is necessary. For example, you might save waterfalls to analyze the discounts you give customers, or to calculate incentive compensation.

To save a pricing waterfall

1. Select a line item (for a quote, order, or an agreement).
2. Click the Save Waterfall button in the Line Items list.

Configuration for Waterfalls

You must configure the Siebel application to create a user interface for viewing a stored waterfall. For more information about configuring Siebel applications, see *Using Siebel Tools* . For more information about configuring Siebel Pricer, see *Siebel Order Management Infrastructure Guide* .

Contract-Based Pricing

Contract-based pricing is pricing based on an agreement with a customer to give that customer a special price. In Siebel applications, contracts are also called agreements and are managed in the Agreements screen. For general information about working with contracts, see the topic about agreements in *Siebel Field Service Guide* . For more information about contract-based pricing, see the following information:

- *About Hierarchical Contract-Based Pricing*
- *Defining Contract Pricing in the Agreements Screen*
- *Defining Contract Pricing in the Discount Matrices View*
- *Giving the Customer the Best Entitlement Price*

About Hierarchical Contract-Based Pricing

You can configure the product to create hierarchical contract-based pricing.

After you have set up hierarchical contract-based pricing, if you give a pricing entitlement to a parent account, then it will be inherited by all its children accounts. If you give a pricing entitlement to a child account, then it will be combined with any entitlements for its parent account.

For example, when you give a global account a 10 percent discount on all surgical products, you also give a local account that is a child of that global account a 10 percent discount on scalpels. When the child account buys scalpels, it has an entitlement for a 19 percent discount, because it inherits a 10 percent discount from its parent account. and also receives a 10 percent discount on the 90 percent of the list price that it must pay after the inherited discount. For more information about configuring the product to create hierarchical contract-based pricing, see *Siebel Order Management Infrastructure Guide* .

Defining Contract Pricing in the Agreements Screen

When you define contract pricing in the Agreements screen, the entitlement must be based on a customer account and on products. You create an agreement with an account, and you define the products for which that account will get special pricing. For more information about other fields in the Agreements screen, see the topic about agreements in *Siebel Field Service Guide* .

To define contract pricing in the Agreements screen

1. Navigate to the Agreements screen.
2. In the Agreements list, add a new record and complete the necessary fields.

   In the Account field, select the account for the customer who receives special pricing.
3. Click the Entitlements view tab.
4. In the Entitlements list, add a new record and complete the necessary fields.

   In the Type field, select Pricing.
5. Click the Price Details view tab, which is beneath the Entitlement list.
6. Add one or more new records to the Price Details list and complete the necessary fields as described in the following table.

Field	Comments
Product	Select the product for which this account gets a special price.
Target Price	Select the base price to which this special price is applied.
Type of Calculation	Select the type of special discount that this account receives. Options are Discount Amount, % Discount, Markup Amount, % Markup, and Price Override.
Adjustment Value	Select the value used by the calculation. How this value is used depends on the selection in the Type of Calculation field:

Field	Comments
	○ **Discount Amount.** The value is subtracted from the target price. ○ **% Discount.** The value is treated as a percentage, and that percentage is subtracted from the target price. ○ **Markup Amount.** The value is added to the target price. ○ **% Markup.** The value is treated as a percentage, and that percentage is added to the target price. ○ **Price Override.** The value replaces the target price.

Defining Contract Pricing in the Discount Matrices View

Discount matrices allow you to manage pricing entitlements for products or groups of products in one view.

For more detailed information about creating a discount matrix, see *Creating Discount Matrices*

To define contract pricing in the Discount Matrices view

1. Navigate to the Administration - Pricing screen, then Discount Matrices view.
2. Add a new record to the Discount Matrices list and complete the necessary fields.

   In the Discount Matrix Type field, select Entitlement-Based Adjustment.
3. In the Discount Criteria list, specify the fields that you want exposed in the discount matrix:

   a. Add a field to identify the entitlement, such as Entitlement.
   b. Add a field to identify the product or group of products, such as Product.
4. After all the fields are added, click Complete Definition.
5. In the Discount Matrix list, click the name of the new discount matrix.
6. In the Discount Details list, add one or more records.

   In each record:

   a. Select a value in the field that identifies the product.
   b. Select a value in the field that identifies the entitlement.
   c. Use the Adjustment Type and Adjustment Value to specify the adjustment that those products get when sold through that entitlement.

Giving the Customer the Best Entitlement Price

When a sales representative creates a quote for a customer who may have special contract-based pricing, the sales representative clicks the Entitle button to fill in the entitlements that apply to each line item.

You must modify the pricing procedure to apply the discount based on the entitlement returned. The best entitlement pricing works only off of the list price. For more information, see *Siebel Field Service Guide* .

To give the customer the best entitlement price

1. Create a quote or order, as described in *Siebel Order Management Guide* .
2. After adding all the line items, in the Quote Header form, click Entitle.

   In each line item where there is an entitlement that applies, the Entitlement Name field is filled in with the name of the entitlement. That entitlement is used to calculate pricing for that line item.

Multiple Price Types and Totals

Multiple price types allow you to create simple products that have one of three price types. You can also create customizable products with components and assign a price type to each component and to the product root. Price Type is a required field when you define a product.

The three price types are:

- **One-Time.** This price is charged only once. For example, you assign this price type to a piece of equipment that the customer can purchase.
- **Recurring.** This price is charged on a recurring basis. For example, you assign this price type to a service to which the customer subscribes.
- **Usage.** This price is charged based on customer consumption. For example, you assign this price type to an electric service that is billed by kilowatt hour of usage. The Usage price type is provided for information only and is not used in any price calculations.

When a product has multiple price types, quotes and orders must also have multiple price totals, one for charges of each price type and unit of measure. This is important because these different types of prices cannot be added up as one total, because they use different units of measure (each and per month).

About Configuring Pricing Fields in Catalog, eSales, and Configurator

In earlier versions, Siebel Catalog, eSales, and Configurator supported only two pricing fields. In version 7.8 or later, they can be configured to add additional pricing fields.

Without configuration, Siebel Catalog includes two pricing fields: List Price and Your Price for simple products, and Starting At Price and Your Price for customizable products. You can configure Siebel Catalog to add additional price fields. For more information about Siebel Catalog, see *Siebel Order Management Guide* .

Without configuration, the Siebel eSales catalog and shopping cart both include the same two pricing fields as catalog. For more information about eSales, see *Siebel eSales Administration Guide* .

Without configuration, Siebel Configurator includes two pricing fields: Starting At Price and Your Price. You can configure these by changing the configurator template. For more information about Configurator, see *Siebel Product Administration Guide* .

Multiple Currencies in a Quote, Order, or Agreement

One quote, order, or agreement can include prices from multiple price lists that are in different currencies, because the sales representative can use a different price list for an individual line item, if necessary.

In the header, when a user selects the price list, the currency of that price list is automatically entered in the Currency field. All totals for the quote, order, or agreement are in this currency.

In the line items, the default price list is the one selected in the header. However, users can select a currency in the line item, and the line item will use the price list for that currency. Users work with currencies by using the fields described in the following table.

Field	Comment
Currency Code	Enter the code for the currency.
Exchange Date	Enter the date used for currency conversion. The application uses the exchange rate on that day to convert currencies.

For more information, see the topic on creating quotes and orders in *Siebel Order Management Guide* .

Spread Discount Pricing

When sales representatives apply a manual discount to the total price of an order, they can spread that discount among the line items of the order. Because products can be defined with multiple price types, spread discount pricing must consider price types. Spread discount pricing spreads the discount to one-time charges only, not to recurring or usage charges.

Spread discounting performs any needed currency conversions. Users can specify the amount of the discount in a given currency. When it is spread among line items, the discount is converted to the currency in each line item. For more information about end users and spread discounting, see the topic about spreading a discount among line items in *Siebel Order Management Guide* . For more information useful to developers, see the following information:

- *About the Spread Discount Algorithm*
- *About Use of the Spread Discount Service with Pricing Procedures*
- *About Configuring the Spread Discount Source Field*
- *Displaying a Warning if the Exact Discount Cannot Be Spread*

About the Spread Discount Algorithm

This topic contains a high-level description of the algorithm used by the spread discount service:

- Get the BasePrice (List Price or Net Price).

- Convert all prices (BasePrice, Min, Max) to BaseCurrency. (Defaults to system currency, but user chooses from dialog box.)
- Add up all BasePrices up (multiplying each by Qty) to get TotalBasePrice.
- Get TargetPrice (calculated from AmtToDiscount or PercentToDiscount if necessary).
- Calculate total to discount: TotalToDiscount = TotalBasePrice – TargetPrice.
- Loop through the discounts one at a time:

  a. Loop through all Rows that have not reached min or max:

     - Discount each amount by (BasePrice/TotalBasePrice)*TotalToDiscount
     - If that discount pushes price beyond min or max, then:

       - Set the final price of item to its min or max

       - Remove that row from further discounts

  b. After round completes, make adjustment so maxed out rows are not taken into account in the next round's calculation. For each of these rows:

     - Its BaseValue*Qty will be subtracted from TotalBasePrice
     - Its DiscountAmt*Qty will be subtracted from TotalToDiscount

  c. Looping continues until:

     - TotalToDiscount can be spread evenly across all remaining line items, or
     - All items have been discounted to their respective min or max.

- Loop through all rows to convert each back to the line item currency and to round each to line item precision.

  Add up rounding remainders and amount discounted.

- The rounding remainder is added to element with highest BasePrice*Qty that has not already reached its min or max.

  This continues until the rounding remainder rounds to 0 or until no items can be discounted further.

About Use of the Spread Discount Service with Pricing Procedures

When an end user clicks the Spread button, the following occurs:

- One signal is raised to invoke the workflow process Spread Discount Driver Workflow process, which uses the business service ISS Spread Discount Service in one of its steps.
- After this is done, another signal is raised to invoke the workflow process Dynamic Pricing Procedure.

To use the spread discount service with custom pricing procedures, raise signals to call the workflows in this order.

About Configuring the Spread Discount Source Field

When an end user uses spread discount, the list of values for the source field has two values, list price and net price. List price is the default value.

If you configure the application to add a new value to this list of values, then you must make sure the new field is in the row set variable map for the spread discount Spread Discount Variable Map - Row Set. For more information about variable maps, see *Siebel Order Management Infrastructure Guide* .

Displaying a Warning if the Exact Discount Cannot Be Spread

In some cases, the spread discount algorithm might not be able to spread exact total discount over the line items. In this case, it adjusts the total so it can be spread over the line items.

There are two reasons why it might be impossible to spread the discount:

- **Decimal remainder inaccuracy.** For example, the total discount might be $100. After the spread discount process is run, the total might be adjusted to $100.02, because the process rounds prices to the nearest cent.
- **Maximum or minimum prices.** For example, the products that the discount is applied to might have minimum prices that are so high that it is impossible to apply the entire discount to them. By default, the Siebel application does not display a message telling the user that the total has been adjusted.

You can set up the application to display an error message if the total is adjusted.

- If there is a decimal remainder inaccuracy, it displays the message, “Discount cannot be spread completely. Remainder of [Currency Code] [OverflowAmount] will be added to total. This inaccuracy can be improved by increasing the number of line items or decreasing the Quantity of each line item."
- If there is an issue with minimum of maximum prices, it displays the message, “Discount cannot be spread completely. Remainder of [Currency Code] [OverflowAmount] will be added to total. All items have reached their minimum or maximum prices."

The user can either accept the adjustment or cancel it and change the records to make it possible to spread the exact discount.

To display a warning if the exact discount cannot be spread

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. In the Object Explorer, select the Business Component Object.
3. Query for the business service ISS Spread Discount Service.
4. Change the value of the business service user property Suppress Inaccuracy Warnings to N.

   Note: The default value is Y.
5. Deliver the changes to the Siebel runtime repository, and restart the server.

Totaling and Rollups of Pricing

Price totals and rollups consider price types:

- Totals and rollups include one-time and recurring prices.
- Totals and rollups do not include usage prices, because the price is not known until the customer's usage is known.

This information about totals is displayed in the one-time charge total and monthly charge total fields of quote, order, and agreement headers.

Users can specify the currency for the total or rollup, after which the application performs the currency conversions to calculate the total from line items in different currencies.

15 Siebel Pricer Technical Reference

Siebel Pricer Technical Reference

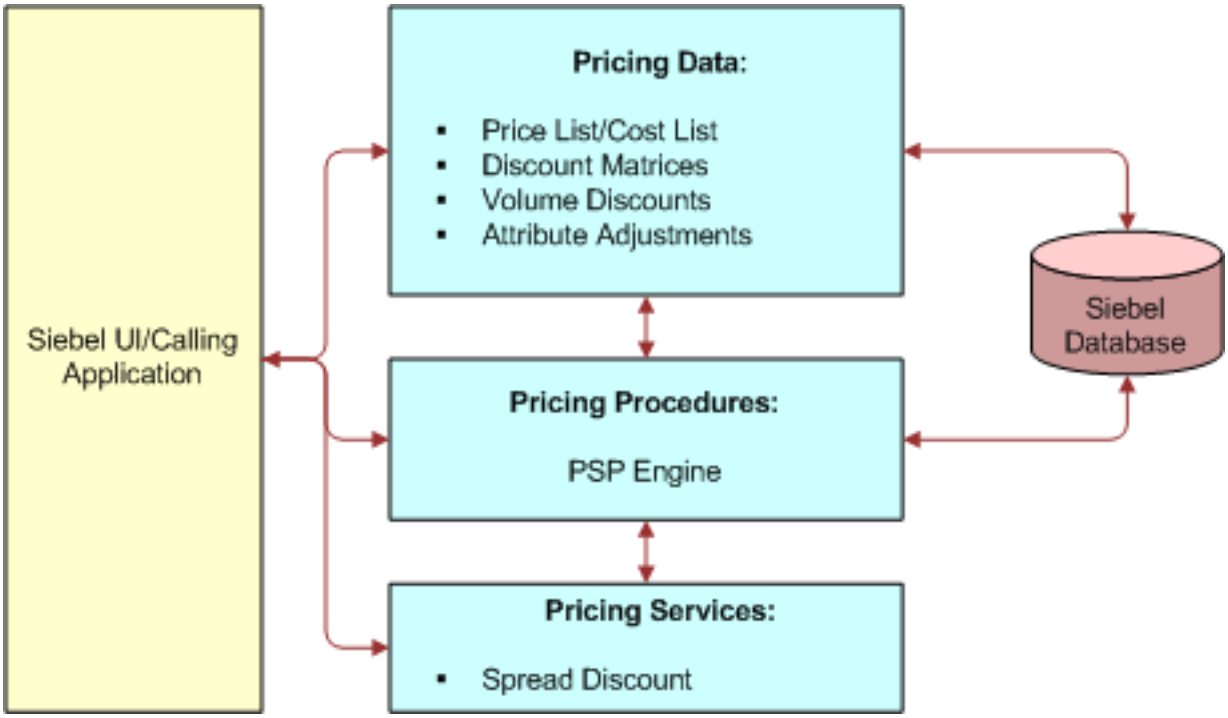
This chapter is a technical reference explaining how Siebel Pricer works. It includes the following topics:

- *Siebel Pricer Architecture*
- *Pricing Procedures and Workflow References*
- *Switching from Basic to Dynamic Pricing Procedure*
- *Modifying Signals from Basic Pricing Procedure to Dynamic Pricing Procedure*
- *Miscellaneous Pricing Workflows*

Siebel Pricer Architecture

A high-level representation of the Siebel Pricing Management Architecture is shown in the following figure. This diagram includes four key components:

- **Siebel UI** or **Calling Application.** This is the point from which the end user requests pricing (for example, quotes and orders).
- **Pricing Data.** This is the administrative business objects that store prices, costs, and adjustments.
- **Pricing Procedures.** This is the centralized location for pricing policies and logic. Pricing procedures use the PSP engine to locate and apply prices and adjustments, and you can customize them for your company's business model. For details about the PSP Engine, see *Siebel Order Management Infrastructure Guide* .
- **Pricing Services.** This is the specialized business services for generic, complex pricing algorithms (such as spread discount).

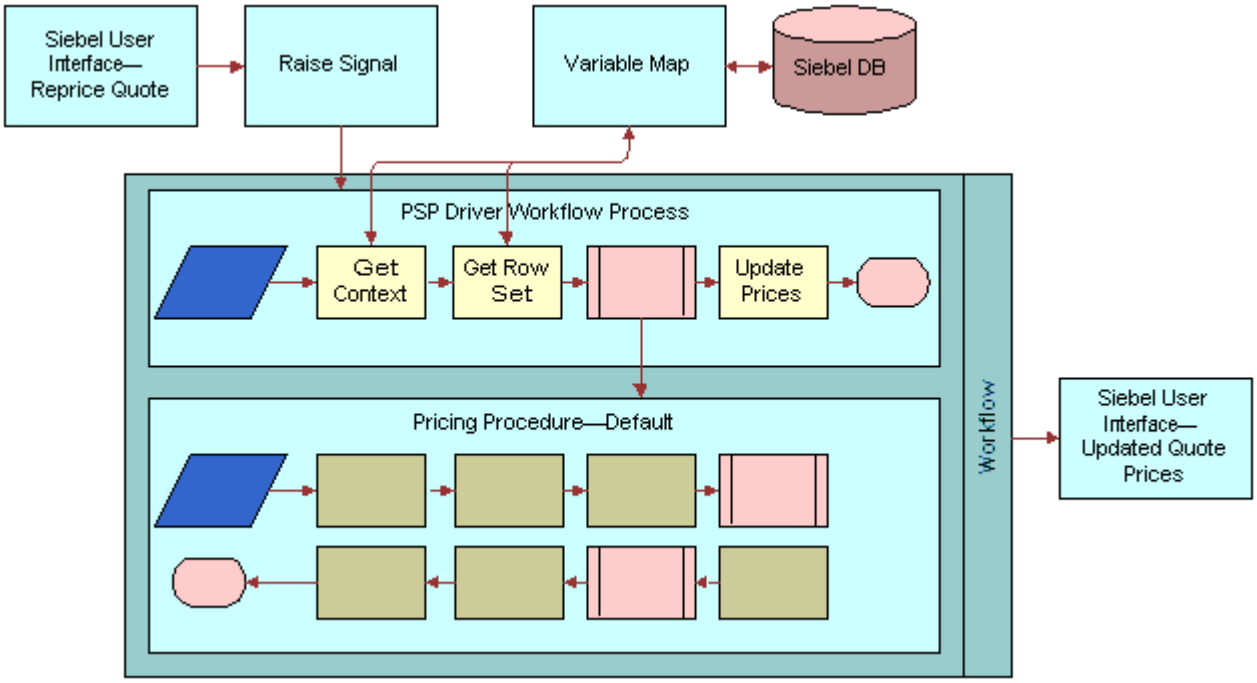


When there is a request for pricing (for example, when a user adds a product to a quote or order or selects Reprice All), the application performs the following steps, illustrated in the following figure:

- The end user action raises a signal, such as the CalculatePriceAll signal. A signal is a parameterized call to a business service or workflow process. For more information about signals, see *Siebel Order Management Infrastructure Guide* .
- The signal calls the controlling workflow process. To continue the previous example, the CalculatePriceAll signal calls the PSP Driver Workflow process. For more information about this workflow process, see *Pricing Procedures and Workflow References*.
- The controlling workflow process uses variable maps to build the Context Row Set (header information) and Line Item Row Set (line item information).
    - The variable maps that build the row sets needed for pricing are identified in the definition of the signal invoked by the end user's action. To continue the previous example, the Default Pricing Variable Map - Context and Default Pricing Variable Map - Row Set are specified as parameters to the PSP Driver Workflow process in the CalculatePriceAll Signal.
    - Variable maps build the row sets by mapping fields in the database to the variables for the row set. For more information about variable maps, see *Siebel Order Management Infrastructure Guide* .
- The controlling workflow process invokes a pricing procedure.
    - The pricing procedure invoked is identified in the definition of the signal invoked by the end user's action. To continue the previous example, the Dynamic Pricing Procedure workflow is specified in the SubPSPWFName parameter to the PSP Driver Workflow process in the CalculatePriceAll Signal.
    - The controlling workflow passes the Context Row Set and Line Item Row Set to the pricing procedure. For more information about controlling workflows and pricing procedures, see *Siebel Order Management Infrastructure Guide* .
- The pricing procedure applies pricing logic to the data in the row sets.

- To continue the previous example, for each line item in the line item row set, the Dynamic Pricing Procedure finds a starting price, executes a series of steps that may or may not adjust the net price, and outputs the final price.
- The Dynamic Pricing Procedure passes the transformed Row Set back to the PSP Driver Workflow process. For detailed information about pricing procedures, see *Pricing Procedures and Workflow References*.

- The Context Service updates the calling object (such as the quote or order) with the adjusted prices, and those prices appear in the end user interface.



Pricing Procedures and Workflow References

To price products, Siebel Pricer uses the following workflows and pricing procedures:

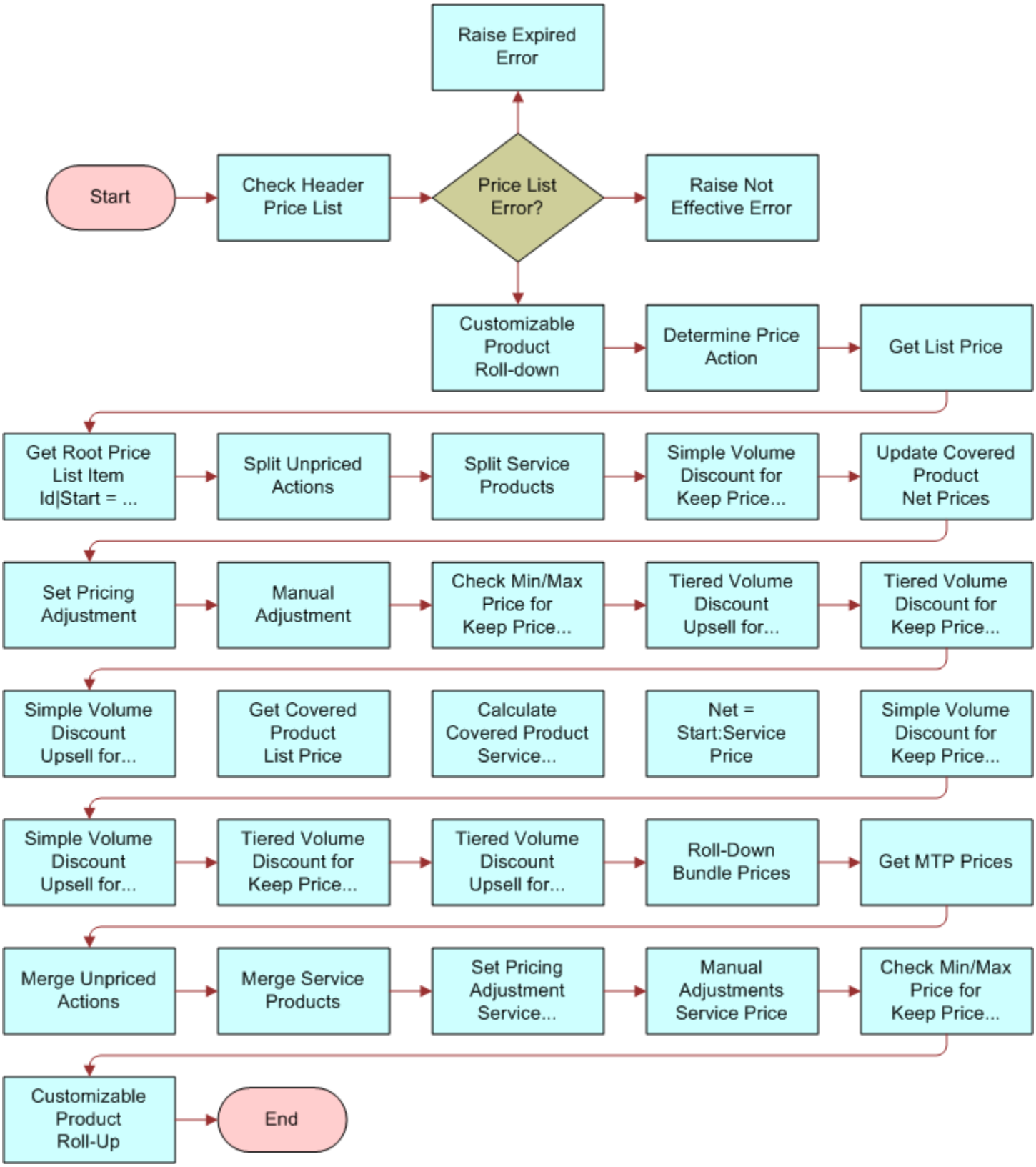
- *PSP Driver Workflow Process Workflow*
- *Basic Pricing Procedure Workflow*
- *Dynamic Pricing Procedure Workflow*
- *Pricing Procedure - Calculate Net Price Workflow*
- *Pricing Procedure - Aggregate Discounts Workflow*
- *Pricing Procedure - Service Workflow*

PSP Driver Workflow Process Workflow

The PSP Driver Workflow Process is the beginning of all pricing operations and other features. For information about this workflow, see *Siebel Order Management Infrastructure Guide* .

Basic Pricing Procedure Workflow

The Basic Pricing Procedure workflow, as shown in the following figure, provides basic pricer license functionality where the list price, volume discounts, and service pricing are supported.


Raise Expired Error
Start
Check Header Price List
Price List Error?
Raise Not Effective Error
Customizable Product Roll-down
Determine Price Action
Get List Price
Get Root Price List Item Id|Start = ...
Split Unpriced Actions
Split Service Products
Simple Volume Discount for Keep Price...
Update Covered Product Net Prices
Set Pricing Adjustment
Manual Adjustment
Check Min/Max Price for Keep Price...
Tiered Volume Discount Upsell for...
Tiered Volume Discount for Keep Price...
Simple Volume Discount Upsell for...
Get Covered Product List Price
Calculate Covered Product Service...
Net = Start:Service Price
Simple Volume Discount for Keep Price...
Simple Volume Discount Upsell for...
Tiered Volume Discount for Keep Price...
Tiered Volume Discount Upsell for...
Roll-Down Bundle Prices
Get MTP Prices
Merge Unpriced Actions
Merge Service Products
Set Pricing Adjustment Service...
Manual Adjustments Service Price
Check Min/Max Price for Keep Price...
Customizable Product Roll-Up
End

Workflow Description

This workflow does the following:

1. Stamps error codes by checking for various conditions on the Price list.
2. From the error codes stamped in step 2, branches out to the different actions listed in step 4.
3. Does the following:
   a. Raises Expired Error
   b. Raises Not Expired Error
   c. Calculates the quantity of each component in a customizable product instance, and rolls down the price list ID and promotion ID from the root.
4. Determines the price action code based on the item's Action Code and Price Type.
5. Using the Simple Look-up Transform business service, searches for the correct price list and stamps various price fields on the line item.
6. Gets Root Price List Item Id|Start = List|Net = Start.
7. Splits Unpriced Actions.

   Note: Do not price line items with Skip Pricing Flag = 'Y'.

8. Splits service and nonservice line items so they can be priced separately.

   Note: In this Pricing Procedure, products and service products need to be split and handled differently. Service procedures are services that are tied to products, for example, a maintenance service on a car. The pricing of a service is tied to the product it covers. Later in the this workflow, steps named " XXX service Pricing" will consume the Service Product Row Set.

9. Applies the volume discount adjustment that matches the quantity on the line item.
10. Finds out the upsell discount information.
11. Applies tiered volume discounts to the net price.
12. Finds out the tiered upsell discount information.
13. Forces the net price to be within the minimum and maximum values defined in the price list item.
14. Sets the manual adjustment and applies line item and header-level manual discounts.
15. Sets the pricing adjustment. Before adjustments and discounts, the Net Price = Start Price.
16. Updates the net price for any covered products that were repriced.
17. Looks up the list price information for covered products.
18. Calculates the start price for the service product by applying the percentage in the price list item to the net or start price of the covered product.
19. Before adjustments and discounts, ensures that the Net Price = Start Price.
20. Applies the volume discount adjustment that matches the quantity on the line item.
21. Determines the upsell discount information.
22. Applies tiered volume discounts to the net price.
23. Determines the tiered upsell discount information.
24. Applies line item and header-level manual discounts.
25. Merges service and nonservice line items.
26. Merges unpriced line items.
27. Gets the MPT Prices and the NRC and MRC price components.
28. Spreads discounts from bundle product root to subcomponents.
29. Calculates the total extended price of the customizable product, including all subcomponents.

    The following table lists the steps in the Basic Pricing Procedure with the business service and the method that is called by each step.

Name	Type	Business Service	Method	Sub Process	Description
Check Header Price List	Business Service	Row Set Transformation Toolkit Service	Query Transform	Not applicable	Stamps error codes by checking for various conditions on the Price list.
Price List Error	Not applicable	Not applicable	Not applicable	Not applicable	From the error codes stamped in step 1, branches out to different actions in the next step.
Raise Expired Error	Business Service	Pricing Manager	Raise Price List Expired Error	Not applicable	Halts execution and provides an error message to the user if the header Price List is expired.
Raise Not Effective Error	Business Service	Pricing Manager	Raise Price List Not Effective Error	Not applicable	Halts execution and provides an error message to the user if the header Price List is not yet effective.
Customizable Product Roll-Down	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Calculates the quantity of each component in a customizable product instance. Rolls down the price list ID and promotion ID from the root.
Determine Price Action	Not applicable	Not applicable	Not applicable	Not applicable	Determines the price action code based on the item Action Code and Price Type.
Get List Price	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Gets the list price for each row in the row set. Appends other values from the price list item Buscomp to the row.

Name	Type	Business Service	Method	Sub Process	Description
Get Root Price List Item Id|Start = List| Net = Start	Not applicable	Not applicable	Not applicable	Not applicable	Not applicable
Split Unpriced Actions	Business Service	Row Set Transformation Toolkit Service	Split	Not applicable	Moves service products from the row set into another row set for use in the service pricing subprocedure.
Split Service Products	Business Service	Row Set Transformation Toolkit Service	Not applicable	Not applicable	Splits service and nonservice line items so they can be priced separately. (Note: Later in the workflow, steps named " XXX service Pricing" would consume the Service Product Row Set).
Simple Volume Discount for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	Applies the volume discount adjustment that matches the quantity on the line item.
Split Service Products	Business Service	Row Set Transformation Toolkit Service	Not applicable	Not applicable	Splits service and nonservice line items so they can be priced separately. (Note: Later in the workflow, steps named " XXX service Pricing" would consume the Service Product Row Set).
Simple Volume Discount for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	Applies the volume discount adjustment that matches the quantity on the line item.

Name	Type	Business Service	Method	Sub Process	Description
Simple Volume Discount Upsell for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	This is a step to find out upsell discount information.
Tiered Volume Discount for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	Applies tiered volume discounts to the Net Price.
Tiered Volume Discount Upsell for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	This is a step to find out tiered upsell discount information.
Check Min or Max Price for Keep Price <> Y	Not applicable	Not applicable	Not applicable	Not applicable	Forces the Net Price to be within the Minimum and Maximum defined in the price list item.
Manual Adjustment	Not applicable	Not applicable	Not applicable	Not applicable	Applies line item and header-level manual discounts.
Set Pricing Adjustment	Business Service	Row Set Transformation Toolkit Service	Split Transform	Not applicable	Before adjustments and discounts, Net Price = Start Price.
Update Covered Product Net Prices	Business Service	Row Set Transformation Toolkit Service	Row Set Lookup Transform	Not applicable	Updates the net price for any covered products that were repriced.
Get Covered Product List Price	Not applicable	Not applicable	Not applicable	Not applicable	Looks up list price information for the covered product.
Update Covered Product Net Prices	Business Service	Row Set Transformation Toolkit Service	Row Set Lookup Transform	Not applicable	Updates the net price for any covered products that were repriced.

Name	Type	Business Service	Method	Sub Process	Description
Get Covered Product List Price	Not applicable	Not applicable	Not applicable	Not applicable	Looks up list price information for the covered product.
Calculate Covered Product Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Calculates the Start Price for a service product by applying the percentage in the price list item to the Net or Start Price of the covered product.
Net = Start:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Before adjustments and discounts, Net Price = Start Price.
Update Covered Product Net Prices	Business Service	Row Set Transformation Toolkit Service	Row Set Lookup Transform	Not applicable	Updates the net price for any covered products that were repriced.
Get Covered Product List Price	Not applicable	Not applicable	Not applicable	Not applicable	Looks up list price information for the covered product.
Calculate Covered Product Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Calculates the Start Price for a service product by applying the percentage in the price list item to the Net or Start Price of the covered product.
Net = Start:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Before adjustments and discounts, Net Price = Start Price.
Simple Volume Discount for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Applies the volume discount adjustment that matches the quantity on the line item.

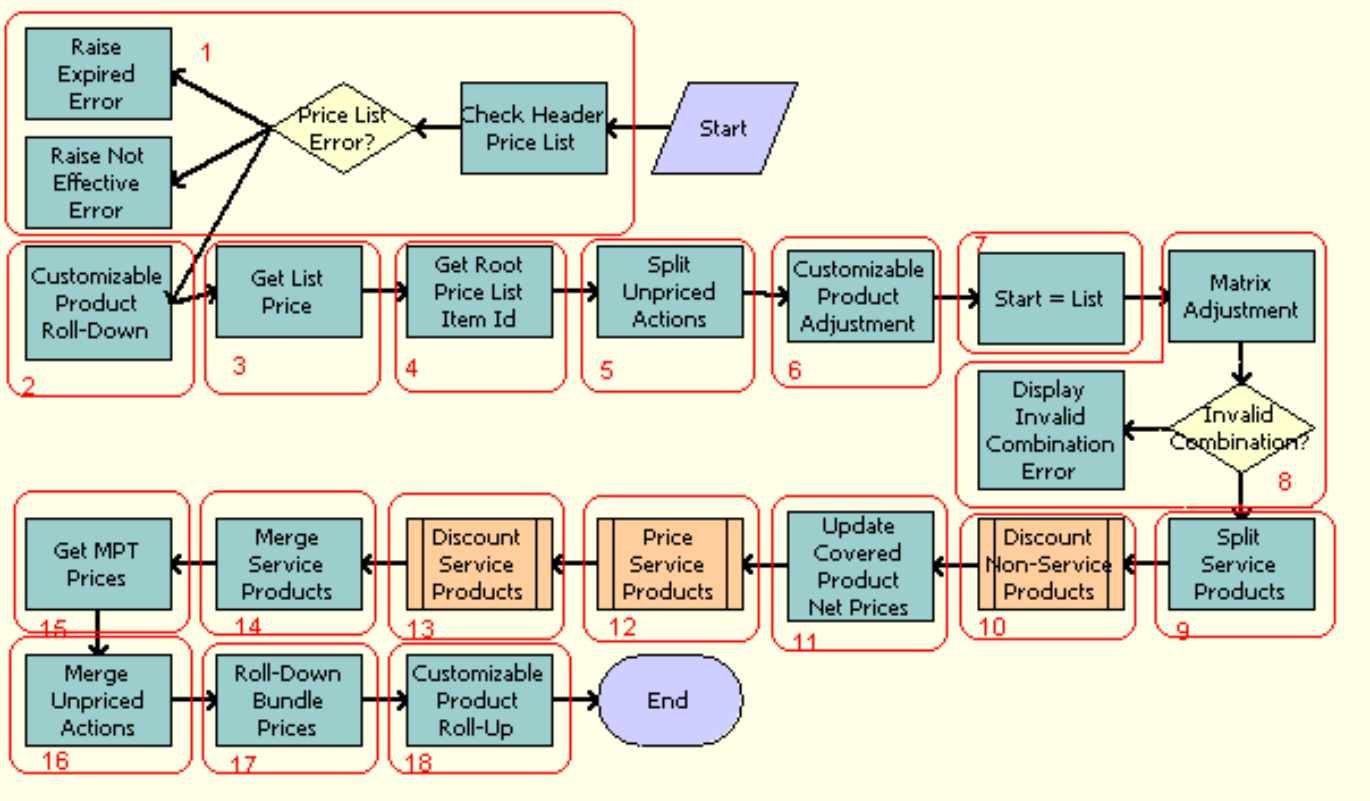
Name	Type	Business Service	Method	Sub Process	Description
Simple Volume Discount Upsell for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	This is a step to find out upsell discount information.
Tiered Volume Discount for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Applies tiered volume discounts to the Net Price.
Tiered Volume Discount Upsell for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Tiered Volume Discount Upsell for Keep Price <> Y:Service Price.
Check Min or Max Price for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Forces the Net Price to be within the Minimum and Maximum defined in the price list item.
Simple Volume Discount for Keep Price <> Y:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Applies the volume discount adjustment that matches the quantity on the line item.
Manual Adjustment:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Applies line item and header-level manual discounts.
Set Pricing Adjustment:Service Price	Not applicable	Not applicable	Not applicable	Not applicable	Before adjustments and discounts, Net Price = Start Price.
Merge Service Products	Business Service	Row Set TransformationToolkit Service	Merge Transform	Not applicable	Merges service and nonservice line items.
Merge Unpriced Actions	Business Service	Row Set	Hierarchical Transform	Not applicable	Merges the priced and unpriced row sets into one.

Name	Type	Business Service	Method	Sub Process	Description
		TransformationToolkit Service			
Get MPT Prices	Business Service	Row Set TransformationToolkit Service	Conditional Action Transform	Not applicable	Gets NRC and MRC price components.
Roll-Down Bundle Prices	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Spreads the discount from the bundle product root to subcomponents.
Customizable Product Roll-Up	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Calculates the total extended price of the customizable product including all subcomponent.

Dynamic Pricing Procedure Workflow

The Dynamic Pricing Procedure workflow, as shown in the following figure, provides the advanced pricing capabilities. It contains steps and subprocedures that populate the List Price, Start Price, and Net Price fields. It also calculates the monthly recurring and nonrecurring subtotals for root Customizable Products. These prices and subtotals are calculated based on information from price lists, customizable product adjustments, volume discounts, attribute adjustments, aggregate discounts, service price details, product promotions, and manual discounts.

Note: The Dynamic Pricing Procedure replaces the Pricing Procedure - Default.



Workflow Description

This workflow does the following:

1. Checks for price list effectivity. The application will throw any error if the price list is not yet effective or has expired.
2. Calculates the extended quantity of all component products of a CP instance. Also, it carries over some useful information from the CP root to CP components for later use.
3. Gets the list price for each row in the row set, and appends other values from the Price List Item Buscomp to the row.
4. Puts the CP Root Price List Item Id on each component of the CP for later use.
5. Moves those rows with an action code that are not priced to another row set that will not be priced throughout the procedure.
6. Gets CP Pricing Designer adjustments for components of a CP.
7. Initializes the start price value as the list price value.
8. Applies attribute adjustments to the start price and throws an error when an invalid combination for attribute pricing occurs.
9. Moves service products from the row set into another row set for use in the service pricing subprocedure.
10. Calls subprocedure Pricing Procedure - Calculate Net Price for nonservice products.
11. Updates the covered product net price field on the service product rows in the row set.
12. Calls subprocedure Pricing Procedure – Service to handle the service product-specific pricing for service products.
13. Calls subprocedure Pricing Procedure - Calculate Net Price for service products.
14. Merges the service and nonservice row sets into one row set.
15. Establishes the NRC and MRC prices for one-time and monthly recurring price types, which will later be used for Customizable Product Rollup.

16. Merges the priced and unpriced row sets into one row set.
17. Spreads the discount of the root bundle products to the bundle component products.
18. Rolls up the prices of component products to the CP root.

The following table lists the steps in the Dynamic Pricing Procedure workflow with the business service and method that is called by each step.

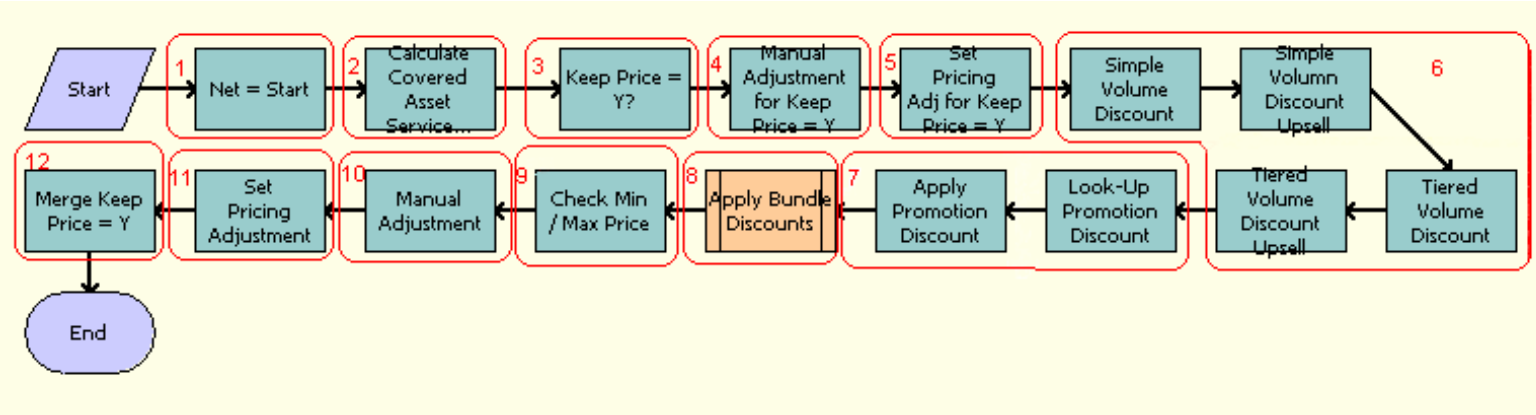
Name	Type	Business Service	Method	Sub Process	Description
Check Header Price List	Business Service	Row Set Transformation Toolkit Service	Query Transform	Not applicable	Determines if the header price list is expired or not yet effective.
Raise Expired Error	Business Service	Pricing Manager	Raise Price List Expired Error	Not applicable	Halts execution and provides an error message to the user if the header Price List is expired.
Raise Not Effective Error	Business Service	Pricing Manager	Raise Price List Not Effective Error	Not applicable	Halts execution and provides an error message to the user if the header Price List is not yet effective.
Customizable Product Roll-Down	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Calculates the quantity of a component product of a CP instance.
Get List Price	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Gets the list price for each row in the row set. Appends other values from the price list item Buscomp to the row.
Get Root Price List Item Id	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Puts the CP root price list item ID on each component of the CP for later use.
Split Unpriced Actions	Business Service	Row Set Transformation Toolkit Service	Split Transform	Not applicable	Moves those rows with an action code that are not priced to another row set that will not be priced throughout the procedure.
Customizable Product Adjustment	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Gets CP adjustments for components of a CP.

Name	Type	Business Service	Method	Sub Process	Description
Start = List	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Initializes the start price value as the list price value.
Matrix Adjustment	Business Service	Row Set Transformation Toolkit Service	Dynamic Lookup Transform	Not applicable	Applies attribute adjustments to the start price.
Invalid Combination?	Decision Point	Not applicable	Not applicable	Not applicable	Determines if there was an invalid combination for the attribute pricing matrix.
Display Invalid Combination Error	Business Service	Dynamic Matrix Retrieval Service	Throw Invalid Combination Error	Not applicable	Halts execution of procedure and throws an error when an invalid combination for attribute pricing occurs.
Split Service Products	Business Service	Row Set Transformation Toolkit Service	Split	Not applicable	Moves service products from the row set into another row set for use in the service pricing subprocedure.
Discount Non Service Products	Sub Process	Not applicable	Not applicable	Pricing Procedure – Calculate Net Price	Not applicable
Update Covered Product Net Prices	Business Service	Row Set Transformation Toolkit Service	Row Set Lookup Transform	Not applicable	Updates the covered product net price field on the service product rows in the row set.
Price Service Products	Sub Process	Not applicable	Not applicable	Pricing Procedure – Calculate Net Price	Prices the service product row set.
Discount Service Products	Sub Process	Not applicable	Not applicable	Pricing Procedure – Service	Service Pricing.

Name	Type	Business Service	Method	Sub Process	Description
Merge Service Products	Business Service	Row Set Transformation Toolkit Service	Merge Transform	Not applicable	Merges the service and nonservice row sets into one.
Get MPT Prices	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Establishes the NRC and MRC prices for One-Time and Monthly Recurring price types.
Merge Unpriced Actions	Business Service	Row Set Transformation Toolkit Service	Merge Transform	Not applicable	Merges the priced and unpriced row sets into one.
Roll-down bundle prices	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Spreads the discount of the root bundle products to the child products.
Customizable Product Roll-ups	Business Service	Row Set Transformation Toolkit Service	Hierarchical Transform	Not applicable	Rolls up the prices of component products to the CP root.

Pricing Procedure - Calculate Net Price Workflow

The Pricing Procedure - Calculate Net Price workflow, as shown in the following figure, is responsible for the net price calculation, which takes into account volume discount, aggregate discount sequence, product promotion, manual adjustment and minimum or maximum price bound checks.



Workflow Description

This workflow does the following:

1. Initializes the net price value as the start price value.
2. Queries in FS Product Price Item Details Buscomp for any covered asset service pricing adjustment.
3. Moves those rows with the Exclude Pricing flag set to another the Keep Price Row Set row set. This will skip all Siebel-Pricer-generated discounts throughout the procedure.
4. Applies a manual adjustment (discount amount, discount percentage, discount price and header discount percentage) for the row set with Exclude Pricing flag Set.
5. Sets the pricing adjustment that indicates the total amount of Siebel-Pricer-generated discount accounts for the row set with Exclude Pricing flag = Y.
6. Evaluates simple and tiered volume discount. This would also populate the next discount and upsell message for upsell purposes.
7. Evaluates product promotion discounts.
8. Calls the Pricing Procedure - Bundle Discount subprocedure to evaluate the aggregate discount sequence.
9. Checks whether the current net price is within the boundary of the minimum and maximum price.
10. Applies a manual adjustment (discount amount, discount percentage, discount price and header discount percentage).
11. Sets the pricing adjustment that indicates the total amount of Siebel-Pricer-generated discount accounts.
12. Merges the two row sets generated from the Keep Price = Y? back into one row set.

    The following table lists the steps of the Pricing Procedure - Calculate Net Price workflow with the business service and method that is called by each step.

Name	Type	Business Service	Method	Sub Process	Description
Net = Start	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Initializes the Net Price value as the Start Price value.
Calculate Covered Asset Service Price	Business Service	Row Set Transformation Toolkit	Simple Look-Up Transform	Not applicable	Queries in FS Product Price Item Details Buscomp for any covered asset service pricing adjustment.
Keep Price = Y?	Business Service	Row Set Transformation Toolkit	Split Transform	Not applicable	Moves those rows with Exclude Pricing Flag set to another row set, the Keep Price Row Set row set. For these, all Siebel-Pricer-generated discounts will be skipped throughout the procedure.
Manual Adjustment for Keep Price = Y	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Applies Manual Adjustment for Keep Price Row Set row set.

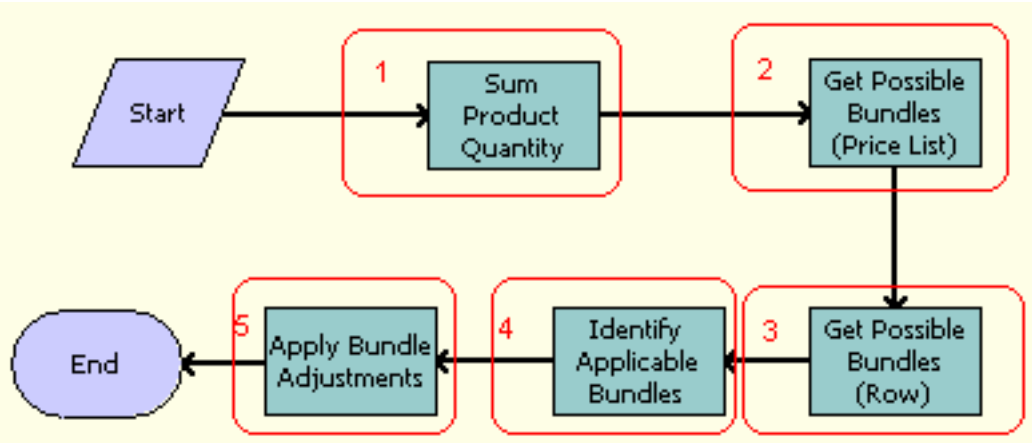
Name	Type	Business Service	Method	Sub Process	Description
Set Pricing Adj for Keep Price = Y	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Sets the Pricing Adjustment that indicates the total amount of Siebel-Pricer-generated discounts for the Keep Price Row Set row set.
Simple Volume Discount	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Evaluates Simple Volume Discount.
Simple Volume Discount Upsell	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Populates Simple Volume Discount Upsell information.
Tiered Volume Discount	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Evaluates Tiered Volume Discount.
Tiered Volume Discount Upsell	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Populates Tiered Volume Discount Upsell information.
Look-Up Promotion Discount	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Checks whether a promotion discount is defined.
Apply Promotion Discount	Business Service	Row Set Transformation Toolkit	Conditional Action Transform	Not applicable	Applies the Promotion Discount found in the previous step.
Apply Bundle Discounts	Sub Process	Not applicable	Not applicable	Pricing Procedure - Aggregate Discounts	Invokes a subprocedure for calculating Aggregate Sequence.
Check Min or Max Price	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Ensures that the net price is within the minimum and maximum defined in the price list item.

Name	Type	Business Service	Method	Sub Process	Description
Manual Adjustment	Business Service	Row Set Transformation Toolkit	Conditional Action Transform	Not applicable	Applies manual adjustment.
Set Pricing Adjustment	Business Service	Row Set Transformation Toolkit Service	Conditional Action Transform	Not applicable	Sets the pricing adjustment, which is the total amount of Siebel-Pricer- generated discounts.
Merge Keep Price = Y	Business Service	Row Set Transformation Toolkit	Merge Transform	Not applicable	Merges back the Keep Price Row Set row set.

Pricing Procedure - Aggregate Discounts Workflow

The Pricing Procedure - Aggregate Discounts workflow, as shown in the following figure, is the subprocedure that evaluates the Aggregate Discount Sequence adjustment. The workflow first aggregates quantities for each product across line items. It then checks whether there are possible aggregate discounts set up by examining the price list on both the header and line item level.

The workflow then compares the aggregate discount with the product quantity aggregation it has evaluated earlier to see whether it qualifies for any discounts.



Workflow Description

This workflow does the following:

1. Groups line items by Product ID and performs a quantity aggregation. The result of this operation is a property set with Product ID and corresponding aggregated quantities.

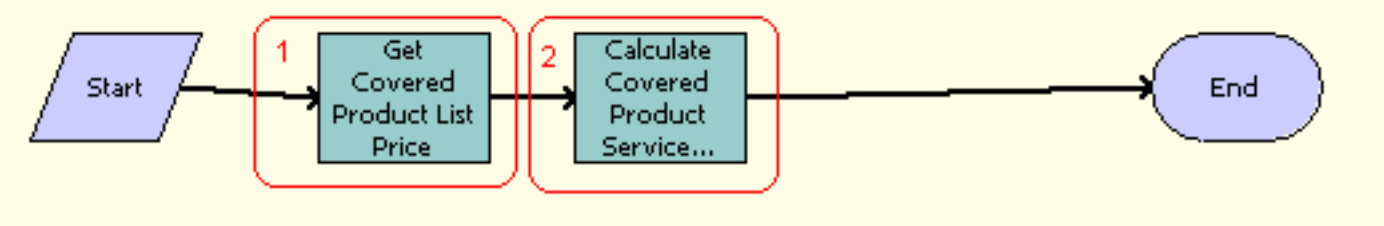
2. Uses Query Transformation to check if there is an Aggregate Discount Sequence associated with the price list from the header, and whether that Aggregate Discount Sequence is effective. If all conditions are met, then the list of IDs of all the aggregate discounts under that sequence would be stored in a property set.
3. This step is similar to the previous one except that it examines the Price List association of the line item. The IDs of all the possible aggregate discounts would be appended to the previously mentioned property set.
4. Using Rule Set Lookup Transformation and the information from previous steps, it identifies which of the possible aggregate discounts are satisfied along with the corresponding multiplier for each aggregate discount. This information is used as input to the next step.
5. Using Rule Set Lookup Transformation and the information from previous steps, it applies adjustments defined in the applicable aggregate discounts and put that on each of the line item. Line items having the same product would be ordered by their net price. Line items with a higher net price would have priority over items with a lower net price.

The following table lists the steps in the Pricing Procedure - Aggregate Discounts workflow with the business service and method that is called by each step.

Name	Type	Business Service	Method	Sub Process	Description
Sum Product Quantity	Business Service	Row Set Transformation Toolkit	Aggregate Transform	Not applicable	Sums up quantities of line items by Product ID.
Get Possible Bundles (Price List)	Business Service	Row Set Transformation Toolkit Service	Query Transform	Not applicable	Checks for the Aggregate Discount Sequence associated with the Price List from the Header.
Get Possible Bundles (Row)	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Checks for the Aggregate Discount Sequence associated with the Price Lists from the Line Items.
Identify Applicable Bundles	Business Service	Row Set Transformation Toolkit Service	Rule Set Look-Up Transform	Not applicable	Identifies applicable Aggregate Discounts and corresponding Multiplier.
Apply Bundle Adjustments	Business Service	Row Set Transformation Toolkit Service	Rule Set Look-Up Transform	Not applicable	Applies the adjustments defined in aggregate discounts.

Pricing Procedure - Service Workflow

The Pricing Procedure - Service workflow, as shown in the following figure, adjusts the start prices for service products by taking account their covered products and primary covered assets.



Workflow Description

This workflow does the following:

1. Looks up list price information for covered products.
2. Calculates the start price for a service product by applying the percentage in the Price List Item Buscomp to the net price or start price of the covered product.

The following table lists the steps in the Pricing Procedure - Service workflow with the business service and method that is called by each step.

Name	Type	Business Service	Method	Sub Process	Description
Get Covered Product List Price	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Looks up list price information for the covered product.
Calculate Covered Product Service Price	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Calculates the Start Price for a service product by applying the percentage in the Price List Item to the Net Price or Start Price of the covered product.

Switching from Basic to Dynamic Pricing Procedure

It is possible to switch from the Basic Pricing Procedure to the Dynamic Pricing Procedure, depending on your requirements.

- The Basic Pricing Procedure is enabled out of the box. It does the following:
  - Duplicates the 7.7 basic pricer license functionality where only the list price, volume discounts, and service pricing are supported.
  - Uses the variable maps that do not contain the XA child variable maps.
- The Dynamic Pricing Procedure is not enabled for out of the box execution. It does the following:
  - Uses the variable maps that contain the XA child variable maps.

Provides complete pricing capabilities such as attribute pricing, promotions, aggregate discounts and so on.

You can either manually enable the Dynamic Pricing Procedure for standard signals as required, or you can automatically enable the Dynamic Pricing Procedure for all signals, as described in the following procedure.

To switch from the Basic Pricing Procedure to the Dynamic Pricing Procedure

1. Make sure that you have the Dynamic Pricer license.

   > **Note:** Before proceeding any further, it is recommended that you familiarize yourself with the Dynamic Pricing Procedure, as described in *Dynamic Pricing Procedure Workflow*.

2. Do one of the following:

   a. Manually enable the Dynamic Pricing Procedure for appropriate standard signals:

      - For information about how to manually enable dynamic pricing for a signal, see *Manually Enabling the Dynamic Pricing Procedure*.
      - For a list of the standard signals that can be manually modified for dynamic pricing, including the parameters in each signal that must be modified, see *Manual Signal Modifications for Siebel Business Applications and Industry Applications* .

   b. Automatically enable the Dynamic Pricing Procedure for all pricing signals:

      - Navigate to the Administration - Pricing screen, then Price Lists view in your Siebel Business Application.
      - Click Menu, and then select Enable Dynamic Pricing.

        A message dialog displays confirming that Dynamic Pricing is enabled.

        Doing this automatically makes the appropriate changes to all standard signal definitions.

        You can extend the list of signals to which Dynamic Pricing applies as described in *Extending the List of Signals to which Dynamic Pricing Applies*.

Extending the List of Signals to which Dynamic Pricing Applies

Extending the list of signals to which Dynamic Pricing applies involves adding a new signal that has been configured for dynamic pricing, so that it automatically gets enabled for dynamic pricing when the Enable Dynamic Pricing option is selected from the Applet Menu.

Complete the steps in the following procedure to extend the list of signals to which Dynamic Pricing applies.

To extend the list of signals to which dynamic pricing applies

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. Select Business Service in the Object Explorer and search for the Signal Modify Service business service.
3. Under the Business Service User Properties, add a new user property to the Business Service.

   This new user property must correspond to the Signal that you want to enable dynamic pricing for. For example, to enable dynamic pricing for a Signal named TestSignal:

   a. Under the Business Service User Properties, search for Signal *.
   b. Add a new record and name it Signal *, where * corresponds to the next consecutive number.

For example, if the highest Signal * User Property is Signal 24, then create a new user property called Signal 25.

c. Specify the Signal name as the User Property Value, consider this example in the following table, TestSignal:

User Property Name	User Property Value
Signal 25	TestSignal

4. Under the Business Service User Properties, add another new record and specify the signal parameters to change to implement dynamic pricing for that signal.

   For example, the following table lists the User Properties that you can create under the Signal Modify Service Business Service in order to enable dynamic pricing for a signal called TestSignal.

User Property Name	User Property Value
TestSignal 1	SubPSPPricingWFName:Dynamic Pricing Procedure
TestSignal 2	Variable Map - Row Set:Default Pricing Variable Map - Row Set

   The User Property Value corresponds to the Signal Parameter that you want to change, and is a combination of the Signal Parameter Name (for example, SubPSPPricingWFName) and the Signal Parameter Value (for example, Dynamic Pricing Procedure) with a colon (:) separating them. For example:

   SubPSPPricingWFName:Dynamic Pricing Procedure

   Continue to add more records, as necessary, specifying the signal parameters to change to implement dynamic pricing for the signal. For each new record, be sure to increase the numeric value after the Signal name (TestSignal 3, TestSignal 4, and so on).
5. Deliver the Business Service to the Siebel runtime repository, and apply it to the server.
6. Restart the server.

Modifying Signals from Basic Pricing Procedure to Dynamic Pricing Procedure

The Siebel Business Application executes the Basic Pricing Procedure. In order to leverage the Dynamic Pricing Procedure, you need to obtain a Dynamic Pricer license and either:

- Automatically enable the Dynamic Pricing Procedure for all signals, as described in Step 2b of *Switching from Basic to Dynamic Pricing Procedure*.
- Or, manually modify several signal definitions as detailed in the following topics:

> **Note:** If you automatically enable the Dynamic Pricing Procedure for all signals as described in Step 2b of *Switching from Basic to Dynamic Pricing Procedure*, then you do not need to make any changes to the signal definitions, and can ignore the following tasks:

- *Manually Enabling the Dynamic Pricing Procedure*
- *Manual Signal Modifications for Siebel Business Applications and Industry Applications*
- *Manual Signal Modifications for Communications, Media and Energy Applications*

Manually Enabling the Dynamic Pricing Procedure

Manually enabling the Dynamic Pricing Procedure involves modifying several of the standard signals. Complete the following procedure to manually enable dynamic pricing for each standard signal.

> **Note:** Complete this procedure only if you have not automatically enabled the Dynamic Pricing Procedure for all price lists as described in Step 2b of *Switching from Basic to Dynamic Pricing Procedure*.

To manually enable the Dynamic Pricing Procedure

1. In the Siebel client, navigate to the Administration – Order Management screen, then the Signals view.
2. Check the Locked flag in the Signals list applet, then drill down on the Work Space version of the signal.
3. Modify the parameters listed for the specified signal actions. If more than one action is listed, then modify the parameters for all actions listed.

   Manual Signal Modifications for Siebel Business Applications and Industry Applications lists the standard signals that must be modified, including the parameters in each signal that must be modified, to enable dynamic pricing for your Siebel Business Application.
4. After modifying the signal definition, you need to release the updated signal as follows:

   a. Return to the Signals list applet
   b. Highlight the modified signal record highlighted
   c. Click the Release New Version button.
5. You will need to log out and log back into the Siebel client to test the updated signal.

For additional details about using signals, see *Siebel Order Management Infrastructure Guide* .

Manual Signal Modifications for Siebel Business Applications and Industry Applications

> **Note:** If you automatically enable the Dynamic Pricing Procedure for all signals as described in Step 2b of *Switching from Basic to Dynamic Pricing Procedure*, then you do not need to implement any of the modifications outlined in this topic.

This topic lists the manual signal modifications required to implement dynamic pricing for Siebel Business Applications and Industry Applications. For each signal definition, you can modify the parameters of the actions listed. If multiple actions are listed for the signal, then make sure that you modify the parameters for each of the actions listed.

Note: In some cases, the signal definition includes additional actions and parameters that are not listed. The following actions and parameters are the only ones that must be changed to implement dynamic pricing.

CalculatePriceAndCheckEligibility

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePriceAndCheckEligibility signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	-	-

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Eligibility Variable Map - Row Set

QuotesAndOrdersValidate

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the QuotesAndOrdersValidate signal.

Note: Make sure you change the parameters for each of the four actions listed.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Order	Header
	Workflow	PSP Driver Workflow Process	Order	Line Item
	Workflow	PSP Driver Workflow Process	Quote	Header
	Workflow	PSP Driver Workflow Process	Quote	Line Item

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Eligibility Variable Map - Row Set

Verifyitem

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the Verifyitem signal.

Note: Make sure you change the parameters for each of the three actions listed.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Agreement	Header
	Workflow	PSP Driver Workflow Process	Order	Header
	Workflow	PSP Driver Workflow Process	Order	Header

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Eligibility Variable Map - Row Set

ProductRecommendation

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the ProductRecommendation signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	Product Recommendation Driver Workflow	Not applicable	Line Item

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure

CalculatePriceExternal

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the CalculatePriceExternal signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Business Service	RunPricerWorkflow	Not applicable	Not applicable

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure

SpreadDiscount

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the SpreadDiscount signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	Spread Discount Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure

SpreadDiscount - All

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the SpreadDiscount - All signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	Spread Discount Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPPricingWFName	Dynamic Pricing Procedure

Approveitem

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the Approveitem signal.

Note: Make sure you change the parameters for each of the three actions listed.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Agreement	Header
	Workflow	PSP Driver Workflow Process	Order	Header
	Workflow	PSP Driver Workflow Process	Quote	Header

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePrice

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePrice signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePriceAll

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePriceAll signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Line Item

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePrice_eSales

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePrice_eSales signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePriceAll_eSales

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePriceAll_eSales signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Line Item

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePrice_Configurator

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePrice_Configurator signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePrice_Advisor

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePrice_Advisor signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

MergeIntoOnePackage

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the MergeIntoOnePackage signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Line Item

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

OrderTemplate

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the OrderTemplate signal.

Note: Make sure you change the parameters for each of the two actions listed.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Order	Header
	Workflow	PSP Driver Workflow Process	Quote	Header

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

OrderTemplateCopy

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the OrderTemplateCopy signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Order	Header

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

OrderTemplateSelectItems

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the OrderTemplateSelectItems signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Order	Header

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

QuoteTemplateCopy

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the QuoteTemplateCopy signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Quote	Header

Name	Value
SubPSPPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

QuoteTemplateSelectItems

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the QuoteTemplateSelectItems signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Quote	Header

Name	Value
SubPSPPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

SetFieldValue

The following tables describe the signal actions and list the parameters that you must modify to implement dynamic pricing for the SetFieldValue signal.

Note: Make sure you change the parameters for each of the four actions listed.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Header
	Workflow	PSP Driver Workflow Process	Agreement	Line Item
	Workflow	PSP Driver Workflow Process	Order	Line Item
	Workflow	PSP Driver Workflow Process	Quote	Line Item

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Default Pricing Variable Map - Row Set

Note: The following fields are available for the Agreement, Order, and Quote actions: Product, Quantity Requested, Unit Price, Discount Amount, Discount Percent, Prod Prom Name, Currency Code, Item Price List, Covered Product Net Price. Fields available for the Header instance type are: Discount, Price List, and Currency Code.

GetConfigItemPrice

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the GetConfigItemPrice signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	Get Config Item Price - Driver Workflow	Not applicable	Not applicable

Name	Value
SubPSPWFName	Get Config Item Price - Dynamic

GetProdPrice

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the GetProdPrice signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Product Pricing Variable Map - Row Set

GetUsedProdPrice

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the GetUsedProdPrice signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Dynamic Pricing Procedure
Variable Map - Row Set	Product Pricing Variable Map - Row Set

Manual Signal Modifications for Communications, Media and Energy Applications

> **Note:** If you automatically enable the Dynamic Pricing Procedure for all price lists as described in Step 2b of *Switching from Basic to Dynamic Pricing Procedure*, then you do not need to implement any of the modifications outlined in this topic.

This topic lists the manual signal modifications required to implement dynamic pricing for Siebel Communications, Media and Energy (CME) applications.

Note: If you are using tiered pricing, then the following signals should refer to Pricing Procedure - CME.

CalculatePriceCME

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePriceCME signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Not applicable

Name	Value
SubPSPWFName	Pricing Procedure - CME
Variable Map - Row Set	Default Pricing Variable Map - Row Set

CalculatePriceAllCME

The following tables describe the signal action and list the parameters that you must modify to implement dynamic pricing for the CalculatePriceAllCME signal.

Definition	Service Type	Service Name	Mode	Instance Type
Action	Workflow	PSP Driver Workflow Process	Not applicable	Line Item

Name	Value
SubPSPWFName	Pricing Procedure - CME
Variable Map - Row Set	Default Pricing Variable Map - Row Set

SaveWaterfall-Quote

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the SaveWaterfall-Quote signal.

Definition	Service Type	Service Name	Service Method	Mode	Instance Type
Action	Business Service	PSP Waterfall Service	SyncToDB	Not applicable	Not applicable

Definition	Service Type	Service Name	Service Method	Mode	Instance Type

Name	Value
Waterfall Signal	CalculatePriceAllCME

SaveWaterfall-Order

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the SaveWaterfall-Order signal.

Definition	Service Type	Service Name	Service Method	Mode	Instance Type
Action	Business Service	PSP Waterfall Service	SyncToDB	Not applicable	Not applicable

Name	Value
Waterfall Signal	CalculatePriceAllCME

SaveWaterfall-ServiceAgreement

The following tables describe the signal action and list the parameter that you must modify to implement dynamic pricing for the SaveWaterfall-ServiceAgreement signal.

Definition	Service Type	Service Name	Service Method	Mode	Instance Type
Action	Business Service	PSP Waterfall Service	SyncToDB	Not applicable	Not applicable

Name	Value
Waterfall Signal	CalculatePriceAllCME

Miscellaneous Pricing Workflows

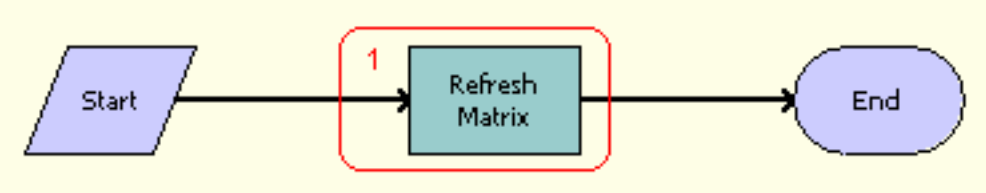
The workflows used for pricing are covered in *Pricing Procedures and Workflow References*.

The following workflows are used by Siebel Pricer for other purposes:

- *PSP Dynamic Matrix - Refresh Matrix Cache Workflow*
- *PSP Waterfall Synch to DB Workflow*
- *Spread Discount Driver Workflow*

PSP Dynamic Matrix - Refresh Matrix Cache Workflow

The PSP Dynamic Matrix - Refresh Matrix Cache workflow, as shown in the following figure, clears the cache for the Attribute Adjustment Rules that are internally maintained by the PSP Engine.



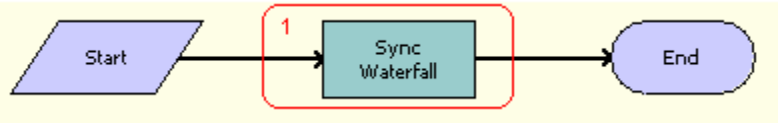
This workflow refreshes the Attribute Adjustment cache for the currently highlighted record.

The following table describes the step in the PSP Dynamic Matrix - Refresh Matrix Cache workflow with the business service and method that it calls.

Name	Type	Business Service	Method	Sub Process	Description
Refresh Matrix	Business Service	Dynamic Matrix Retrieval Service	Refresh Matrix	Not applicable	Refreshes the Attribute Adjustment cache for the current highlighted record.

PSP Waterfall Synch to DB Workflow

The PSP Waterfall Synch to DB workflow, as shown in the following figure, writes waterfall records to the database according to input from the data map object.



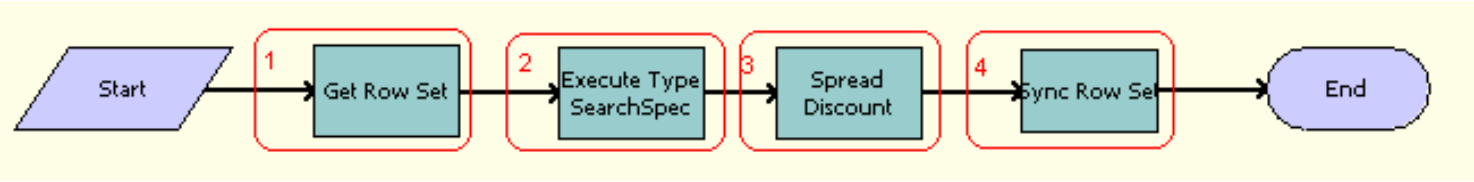
The following table describes the step in the PSP Waterfall Synch to DB workflow with the business service and method that is called.

Name	Type	Business Service	Method	Sub Process	Description
Sync Waterfall	Business Service	PSP Waterfall Service	WriteWaterfall	Not applicable	Writes waterfall records to database.

Spread Discount Driver Workflow

The Spread Discount Driver workflow, as shown in the following figure, spreads a discount amount over the specified line items.

Note: The specialized implementation in the Quote Spread Discount Pop-up applet subsequently triggers Pricing Procedure – Default, which evaluates the net prices and subtotal based on the new discount amounts.



Workflow Description

This workflow does the following:

1. Generates a property set of the Line Item buscomp using Context Service variable maps. For more information, see *Siebel Order Management Guide* .
2. Filters out items that will not get a spread discount. By default, only nonrecurring change items are candidates for a spread discount.
3. Spreads a discount amount depending on the target total or target discount amount specified by the user.
4. Synchronizes the updated information back to the database.

   The following table lists the steps in the Spread Discount Driver workflow with the business service and method that is called for each step.

Name	Type	Business Service	Method	Sub Process	Description
Get Row Set	Business Service	Context Service	GetRowSetData	Not applicable	Generates a property set from the Line Item business component.
Execute Type SearchSpec	Business Service	Row Set Transformation Toolkit	Split Transform	Not applicable	Filters out items that will not get a spread discount.
Spread Discount	Business Service	ISS Spread Discount Service	SpreadDiscount	Not applicable	Spreads a discount amount.
Sync Row Set	Business Service	Context Service	SyncRowSetData	Not applicable	Synchronizes the updated information back to database.

16 Siebel Pricer Deployment and Integration

Siebel Pricer Deployment and Integration

This chapter contains information about integrating Siebel Pricer with Siebel business components and external applications. It covers the following topics:

- *About Pricing Integration*
- *Integration Methods for Siebel Pricer*
- *Internal Integration of Siebel Pricer*
- *External Integration of Siebel Pricer Using Pricer as a Service*
- *Siebel Pricing Manager API Reference*
- *Workflows for External Pricing API*
- *Cache Configuration Settings*

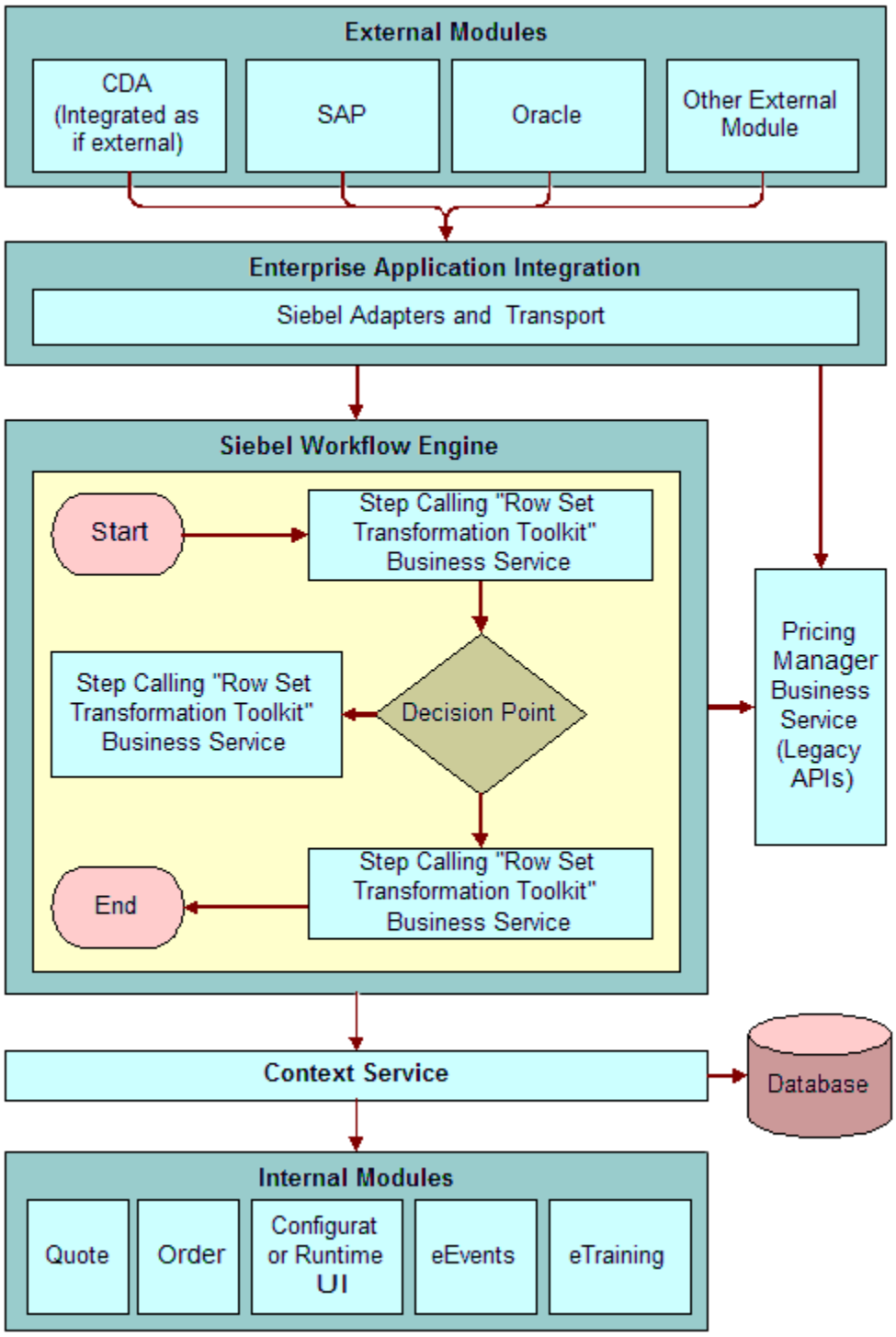
About Pricing Integration

You can integrate Siebel Pricer internally or externally, that is, from other Siebel application modules, or from other manufacturers' applications.

Siebel Pricer functionality is used internally for quotes, orders, and agreements, and can be used with popular external applications and services, for example, Enterprise Resource Planning (ERP) applications. Pricing integration executes as follows:

- **External Modules.** This is the point where the integration starts. For example, Oracle eBusiness Suite, SAP or other systems provide product pricing information to Siebel.
- **Enterprise Application Integration.** Siebel adapters and transport mechanisms are used to obtain the pricing information from external modules.
- **Workflow Engine or the Pricing Manager Business Service.** Siebel workflow processes are triggered using signals that use events within the Siebel application.
- **Context Service.** The first step in the pricing workflow is to retrieve the context information from the quote, the order, and so on, to obtain the header information from the database, and then to provide it to the pricing workflow.
- **Siebel Workflow Process.** This consists of steps that invoke methods from Row Set Transformation Toolkit business service.

The following figure shows how the Pricing Manager business service can be integrated to provide Siebel Pricer functionality to both internal and external application modules.


External Modules
CDA (Integrated as if external)
SAP
Oracle
Other External Module
Enterprise Application Integration
Siebel Adapters and Transport
Siebel Workflow Engine
Start
Step Calling "Row Set Transformation Toolkit" Business Service
Decision Point
Step Calling "Row Set Transformation Toolkit" Business Service
Step Calling "Row Set Transformation Toolkit" Business Service
End
Pricing Manager Business Service (Legacy APIs)
Context Service
Database
Internal Modules
Quote
Order
Configurator Runtime UI
eEvents
eTraining

Integration Methods for Siebel Pricer

Siebel Pricer supports the following types of integration:

- **Internal integration.** This allows you to configure the Siebel application so that Siebel Pricer works with other Siebel business objects. Without configuration, Siebel Pricer works with quotes, orders, and agreements. For more information, see *Internal Integration of Siebel Pricer*.
- **External integration.** This allows you to integrate Siebel Pricer with other external applications, such as back office applications, in the following way:
    - **External integration using Pricer as a service.** You can use the Siebel Pricer engine without using the Siebel interface. Use this method when your business model does not require the external quotes and orders to be replicated in the Siebel application. For more information, see *External Integration of Siebel Pricer Using Pricer as a Service*.
- **Integration using the Pricer API.** The API is retained to support users who used the Pricer API in earlier versions of the Siebel application, but for whom it is no longer the best practice in version 7.8 or later. For more information, see *Siebel Pricing Manager API Reference*.

Internal Integration of Siebel Pricer

Without configuration, pricing functionality is used by quotes, orders, and agreements.

You can configure the application so that any business component can use pricing functionality. This topic gives an example of configuring the Opportunity business component to integrate it with Pricer. You can use similar methods for other business components.

This example allows the Opportunity and Opportunity Product business components to invoke the pricing procedure. For information about the pricing procedure, see *Siebel Pricer Technical Reference*.

> **CAUTION:** Perform these changes on the local environment only. The change has enterprise-wide effect and may affect others using the same URL. Do not perform these changes unless you are familiar with Web Tools and Siebel order management infrastructure.

To allow the Opportunity and Opportunity Product business components to invoke the pricing procedure, do the following:

- *Changing the Base Class of the Business Component to User CSSBCOrderMgmt Base*
- *Setting Up the Context Service Variable Map Integration Object and Mode*
- *Triggering CalculatePriceForOpportunity*
- *Designing the CalculatePriceForOpportunity Signal*
- *Creating an Integration Object*
- *Creating the Variable Map*
- *Creating the Pricing Procedure*

Changing the Base Class of the Business Component to User CSSBCOrderMgmtBase

Use the following procedure to change the base class of the business component to user CSSBCOrderMgmtBase.

To change the base class of the business component to User CSSBCOrderMgmtBase

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. Select Business Component in the Object Explorer.
3. Query for Opportunity Product and Opportunity.
4. Lock these objects.
5. For these objects, set Class to CSSBCOrderMgmtBase.

   Setting the base class to CSSBCOrderMgmtBase enables a business component to raise a signal.

Note: You can do this only if the original business component is not using a specialized buscomp class. Otherwise, you risk losing the functionality offered by the specialized buscomp class.

Setting Up the Context Service Variable Map Integration Object and Mode

Use the following procedure to set up the Context Service Variable Map Integration Object and Mode

To set up the Context Service Variable Map Integration Object and Mode

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. In Object Explorer and under Business Component, click Business Component User Prop.
3. For both the Opportunity Product and Opportunity Buscomp, add the new user properties shown in the following table.

Name	Value
Instance Uniform Name EAI Object:Opportunity	ISS Opportunity
Mode	Opportunity

   a. The first user property sets the Integration Object to be used for the variable map.
   b. The second user property set the mode of the business component. This corresponds to the mode in the Signal and Variable Map definition. In this case, items with mode All and Opportunity would be picked up.

Triggering CalculatePriceForOpportunity

Use the following procedure to expose a button on the UI to trigger CalculatePriceForOpportunity.

To expose a button on the UI to trigger CalculatePriceForOpportunity

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. Click on the Applet object in Object Explorer, and query for Opportunity Product Applet.
3. Expland the Applet list and click Control.
4. In the Control applet, create a new control using the following information:
   - **Name.** Reprice
   - **Caption.** Reprice
   - **HTML Type.** Mini Button
   - **Method Invoked.** CalculatePriceForOpportunity
5. Expand the Applet list and click Applet Web Template.
   a. In Applet Web Templates, select Name = Edit List, and expand the Web Template field.
   b. In Web Templates applet, expand the Name field.
6. In the Object Explorer editor, map the Reprice control to the required control.

   You can preview your changes by clicking on the eye icon.
7. Save and deliver your workspace.

Designing the CalculatePriceForOpportunity Signal

Use the following procedure to design the CalculatePriceForOpportunity signal.

To design the CalculatePriceForOpportunity signal

1. Create a new signal with a name CalculatePriceForOpportunity.
2. Drill down on the workspace.
3. Create an Action as shown in the following table.

Sequence	Service Name	Service Type	Service Method
1	PSP Driver Workflow Process	Workflow	RunProcess

4. Create the parameters as shown in the following table.

Name	Value
CPScope	Whole
RowScope	Selected
SubPSPWFName	Pricing Procedure - Opportunity
Variable Map - Context	Opportunity Pricing Variable Map - Context
Variable Map - Row Set	Opportunity Pricing Variable Map - Row Set

5. Click the Release New Version button.

   Create the signal to be triggered by the button created in Step 4.

Creating an Integration Object

Use the following procedure to create an Integration Object.

To create an integration object

1. In Web Tools, open a workspace and then navigate to Object Explorer.

   For more information on using the workspace dashboard, see *Using Siebel Tools* .
2. Select Integration Object in Object Explorer and create a new record, ISS Opportunity Integration Object, as shown in the following table.

Name	Project	Business Object	External Name	XML Tag
ISS Opportunity	ISS Order Management	Opportunity	ISS Object	ListOfISSObject

3. Expand Integration Object in the Object Explorer and click Integration Component.
4. Create two new records under ISS Opportunity Integration Object, as shown in the following table.

Name	External Name Context	Parent Integration Component	External Name	External Sequence	Card-inality	XML Container Elemen	XML Tag
Header	Opportunity	-	Opportunity	1	One	ListOfHeader	Header

Name	External Name Context	Parent Integration Component	External Name	External Sequence	Card-inality	XML Container Elemen	XML Tag
Line Item	Opportunity Product	Header	Opportunity Product	2	Zero or More	ListOfLineItem	LineItem

5. Enter Integration Component Fields of Opportunity, as shown in the following table.

Name	Data Type	Length	External Name	External Data Type	External Length	XML Style	XML Tag
Name	DTYPE_TEXT	50	Name	DTYPE_TEXT	50	Element	Name

6. Enter Integration Component Fields of Opportunity Product, as shown in the following table.

Name	Data Type	Length	Precisic	External Name	External Data Type	External Length	XML Style	XML Tag
Comment	DTYPE_TEXT	255	N/A	Comment	DTYPE_TEXT	255	Element	Comment
Cost	DTYPE_ NUMBER	N/A	N/A	Cost	DTYPE_ NUMBER	N/A	Element	Cost
Product Id	DTYPE_ID	N/A	N/A	Product Id	DTYPE_ID	N/A	Element	ProductId
Product Price	DTYPE_ NUMBER	N/A	N/A	Product Price	DTYPE_ NUMBER	N/A	Element	ProductPrice
Product Quantity	DTYPE_ NUMBER	N/A	16	Product Quantity	DTYPE_ NUMBER	N/A	Element	Product Quantity
Quotable	DTYPE_TEXT	N/A	N/A	Quotable	DTYPE_BOOL	N/A	Element	Quotable
Revenue	DTYPE_ NUMBER	N/A	N/A	Revenue	DTYPE_ NUMBER	N/A	Element	Revenue

7. Save your changes and submit the workspace for delivery.

Creating the Variable Map

Use the following procedure to create the variable map.

To create the variable map

1. Create the Opportunity Pricing Variable Map - Context Variable Map.
2. Drill down on Workspace.
3. Create the Variable Definition as shown in the following table.

Variable Name	In/Out	Type	On Null
Name	In/Out	Text	Ignore

4. Create the Variable Source for the Variable Definition as shown in the following table.

Mode	Path	Source Type
Opportunity	$Current/Header/Name	Instance

5. Click the Mode tab and create the recording as shown in the following table.

Mode	Object Name
Any	Header

6. Create the Opportunity Pricing Variable Map - Row Set Variable Map.
7. Drill down on Workspace.
8. Create the Variable Definitions as shown in the following table.

Variable Name	In/Out	Type	On Null
Comment	In/Out	Text	Ignore
Cost	In/Out	Number	Ignore
Product Id	In/Out	Id	Ignore
Product Price	In/Out	Number	Ignore

Variable Name	In/Out	Type	On Null
Product Quantity	In/Out	Number	Ignore
Quotable	In/Out	Boolean	Ignore
Revenue	In/Out	Number	Ignore

9. Create the Variable Source for each of these Variable Definitions as shown in the following table.

Mode	Path	Source Type
Opportunity	$Current/Line Item/Comment	Instance
Opportunity	$Current/Line Item/Cost	Instance
Opportunity	$Current/Line Item/Product Id	Instance
Opportunity	$Current/Line Item/Product Price	Instance
Opportunity	$Current/Line Item/Product Quantity	Instance
Opportunity	$Current/Line Item/Quotable	Instance
Opportunity	$Current/Line Item/Revenue	Instance

10. Click the Mode tab and create the recording as shown in the following table.

Mode	Object Name
Any	Line Item

Create the variable maps referenced by the signal.

Creating the Pricing Procedure

Use the following procedure to create the pricing procedure.

To create the pricing procedure

- Create a pricing procedure that drives the pricing logic.

  One example is Pricing Procedure - Opportunity.xml in *Sample Pricing Procedure for Internal Integration*. Make additions or modifications to this example as needed.

  Pricing procedures are a type of PSP procedure. For more information about PSP procedures, see *Siebel Order Management Infrastructure Guide* .

Sample Pricing Procedure for Internal Integration

This topic contains Pricing Procedure - Opportunity.xml, the XML version of a sample pricing procedure meant to drive the pricing logic for internally integrating pricing with the Opportunity and Opportunity Product business components.

```
<?xml version="1.0" encoding="UTF-8" ?>
<?Siebel-Property-Set EscapeNames="false"?>
-<SiebelMessage MessageId="" IntObjectName="Repository Workflow Definition Export
V.7.7.1" MessageType="Integration Object" IntObjectFormat="Siebel Hierarchical">
-<ListOfRepositoryWorkflowProcess>
-<RepositoryWorkflowProcess>
 <BusinessObject />
 <CacheLocale />
 <Comments />
 <Description>Support Pricer APIs GetProductListPrice and GetConfigItemPrice</
 Description>
 <EffectiveEndDate />
 <EffectiveStartDate />
 <ErrorProcessName />
 <Group />
 <Inactive>N</Inactive>
 <Module />
 <Name>Pricing Procedure - Opportunity: 3</Name>
 <PersistentFrequency>N</PersistentFrequency>
 <PersistentLevel />
 <ProcessName>Pricing Procedure - Opportunity</Proces sName>
 <ProjectName>Pricer</ProjectName>
 <ReplicationLevel>None</ReplicationLevel>
 <Runnable>N</Runnable>
 <Status>COMPLETED</Status>
<Version>3</Version>
<WorkflowMode>SRVFLOW</WorkflowMode>
- <ListOfRepositoryWfProcessProp>
- <RepositoryWfProcessProp>
 <AccessMode>RW</AccessMode>
<BusinessComponent />
 <Comments>Variables that are shared across all rows in Row Set.</Comments>
 <CorrelatorFlag>N</CorrelatorFlag>
 <DataType>PROPSET_CHILD</DataType>
 <DefaultDate />
 <DefaultNumber />
 <DefaultString />
 <DisplayName />
 <InOut>INOUT</InOut>
 <Inactive />
 <IntegrationObject />
 <Name2>Context</Name2>
 <VirtualField />
 </RepositoryWfProcessProp>
```

```
- <RepositoryWfProcessProp>
 <AccessMode>RW</AccessMode>
 <BusinessComponent />
 <Comments>The set of line items to reprice.</Comments>
 <CorrelatorFlag>N</CorrelatorFlag>
 <DataType>PROPSET_CHILD</DataType>
 <DefaultDate />
 <DefaultNumber />
 <DefaultString />
 <DisplayName />
 <InOut>INOUT</InOut>
 <Inactive />
 <IntegrationObject />
 <Name2>Row Set</Name2>
 <VirtualField />
 </RepositoryWfProcessProp>
 </ListOfRepositoryWfProcessProp>
- <ListOfRepositoryWfStep>
- <RepositoryWfStep>
 <AllowRetryFlag>N</AllowRetryFlag>
 <BusinessComponent />
 <Operation />
 <BusinessServiceMethod />
 <BusinessServiceName />
 <Comments />
 <Description />
 <ErrorCode />
 <ErrorMessage />
 <EvaluateAll>Y</EvaluateAll>
 <Inactive>N</Inactive>
 <Layout>@0*0*3*0*0*0*1*r13*288 12 352 529*FillBgClr8 *167641088*ArrowEnd1*1</
 Layout>
 <MaximumIterations />
 <Name3>End</Name3>
 <ProcessingMode />
 <SubprocessName />
 <Type>END</Type>
 <UserInteractView />
 <ListOfRepositoryWfStepBranch />
 <ListOfRepositoryWfStepIOArgument />
 <ListOfRepositoryWfStepRecipient />
 </RepositoryWfStep>
- <RepositoryWfStep>
 <AllowRetryFlag>N</AllowRetryFlag>
 <BusinessComponent />
 <Operation />
 <BusinessServiceMethod>Simple Look-Up Transform</Bus inessServiceMethod>
 <BusinessServiceName>Row Set Transformation Toolkit< /BusinessServiceName>
 <Comments />
 <Description>Look up all values on the price list it em.</Description>
 <ErrorCode />
 <ErrorMessage />
 <EvaluateAll>Y</EvaluateAll>
 <Inactive>N</Inactive>
 <Layout>@0*0*3*0*0*0*1*r13*180 12 244 529*FillBgClr8 *134217218*ArrowEnd1*1</
 Layout>
 <MaximumIterations />
 <Name3>Get List Price</Name3>
 <ProcessingMode />
 <SubprocessName />
 <Type>TASK</Type>
 <UserInteractView />
- <ListOfRepositoryWfStepBranch>
- <RepositoryWfStepBranch>
 <Comments />
 <Count />
```

```
 <EvaluateSequence />
 <Event />
 <EventCancelFlag>N</EventCancelFlag>
 <EventObject />
 <EventObjectType />
 <EventVisibility>ENTERPRISE</EventVisibility>
 <Expression />
 <Inactive>N</Inactive>
 <Name2>NewConn:1-3J9N9</Name2>
 <Subevent />
 <TimeLimit />
 <Type>CONNECTOR</Type>
 <UserEventName />
 <UserEventStorage />
 <UserEventTimeout>0</UserEventTimeout>
 <ListOfRepositoryWfBranchCriteria />
 </RepositoryWfStepBranch>
</ListOfRepositoryWfStepBranch>
- <ListOfRepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Component</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Price List Item</ValueSear chSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Component</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Price List Item</ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Component</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Price List Item</ValueSear chSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
```

```
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Component</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Price List Item</ValueSear chSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Object</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Admin Price List</ValueSea rchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Object</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Admin Price List</ValueSea rchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Object</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Admin Price List</ValueSea rchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Business Object</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>Admin Price List</ValueSea rchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
```

```
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Context</Name2>
 <OutputArg />
 <PropertyName>Context</PropertyName>
 <Sequence>0</Sequence>
 <Type>PROPERTY</Type>
 <ValueSearchSpecification />
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 5</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Product Price} = {Mat ch.List Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 5</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Product Price} = {Mat ch.List Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 5</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Product Price} = {Mat ch.List Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 5</Name2>
```

```
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Product Price} = {Mat ch.List Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 6</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Revenue} = {Match.MSR P Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 6</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Revenue} = {Match.MSR P Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 6</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Revenue} = {Match.MSR P Price}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 6</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Revenue} = {Match.MSR P Price}</
 ValueSearchSpecification>
```

```
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 8</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Cost} = {Match.Cost}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 8</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Cost} = {Match.Cost}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 8</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Cost} = {Match.Cost}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>On First Match 8</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>{Row.Cost} = {Match.Cost}</
 ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
```

```
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Row Set</Name2>
 <OutputArg />
 <PropertyName>Row Set</PropertyName>
 <Sequence>0</Sequence>
 <Type>PROPERTY</Type>
 <ValueSearchSpecification />
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Search Specification</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>[Price List Id] = {Context .Name} AND [Product Id] =
 {Row.Product Id}</ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Search Specification</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>[Price List Id] = {Context .Name} AND [Product Id] =
 {Row.Product Id}</ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Search Specification</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
 <ValueSearchSpecification>[Price List Id] = {Context .Name} AND [Product Id] =
 {Row.Product Id}</ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>I</InputFlag>
 <ItemTypeName />
 <Name2>Search Specification</Name2>
 <OutputArg />
 <PropertyName />
 <Sequence>0</Sequence>
 <Type>LITERAL</Type>
```

```
 <ValueSearchSpecification>[Price List Id] = {Context .Name} AND [Product Id] =
 {Row.Product Id}</ValueSearchSpecification>
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>O</InputFlag>
 <ItemTypeName />
 <Name2>O:Get List Price1</Name2>
 <OutputArg>Row Set</OutputArg>
 <PropertyName>Row Set</PropertyName>
 <Sequence>1</Sequence>
 <Type>ARGUMENT</Type>
 <ValueSearchSpecification />
</RepositoryWfStepIOArgument>
- <RepositoryWfStepIOArgument>
 <BusinessComponent />
 <BusinessComponentField />
 <Comments />
 <Inactive>N</Inactive>
 <InputFlag>O</InputFlag>
 <ItemTypeName />
 <Name2>O:Get List Price20</Name2>
 <OutputArg>Context</OutputArg>
 <PropertyName>Context</PropertyName>
 <Sequence>2</Sequence>
 <Type>ARGUMENT</Type>
 <ValueSearchSpecification />
 </RepositoryWfStepIOArgument>
 </ListOfRepositoryWfStepIOArgument>
 <ListOfRepositoryWfStepRecipient />
</RepositoryWfStep>
- <RepositoryWfStep>
 <AllowRetryFlag>N</AllowRetryFlag>
 <BusinessComponent />
 <Operation />
 <BusinessServiceMethod />
 <BusinessServiceName />
 <Comments />
 <Description />
 <ErrorCode />
 <ErrorMessage />
 <EvaluateAll>Y</EvaluateAll>
 <Inactive>N</Inactive>
 <Layout>@0*0*3*0*0*0*1*r12*96 12 160 529*FillBgClr8* 167641088*ArrowEnd1*1</
 Layout>
 <MaximumIterations />
 <Name3>Start</Name3>
 <ProcessingMode />
 <SubprocessName />
 <Type>START</Type>
 <UserInteractView />
- <ListOfRepositoryWfStepBranch>
- <RepositoryWfStepBranch>
 <Comments />
 <Count />
 <EvaluateSequence />
 <Event />
 <EventCancelFlag>N</EventCancelFlag>
 <EventObject />
 <EventObjectType />
 <EventVisibility>ENTERPRISE</EventVisibility>
 <Expression />
 <Inactive>N</Inactive>
```

```
 <Name2>NewConn:1-3J9N7</Name2>
 <Subevent />
 <TimeLimit />
 <Type>DEFAULT</Type>
 <UserEventName />
 <UserEventStorage />
 <UserEventTimeout>0</UserEventTimeout>
<ListOfRepositoryWfBranchCriteria />
 </RepositoryWfStepBranch>
 </ListOfRepositoryWfStepBranch>
 <ListOfRepositoryWfStepIOArgument />
 <ListOfRepositoryWfStepRecipient />
 </RepositoryWfStep>
 </ListOfRepositoryWfStep>
- <ListOfRepositoryWfBranchConnector>
- <RepositoryWfBranchConnector>
 <Comments />
 <FromStepName>Start</FromStepName>
 <Inactive>N</Inactive>
 <Layout>@0*0*9*0*0*0*9*LineWidth1*24*FrHS1*22*P16*18 0 324*NPts1*22*P06*152
 324*ToID8*04-EAATQ6*FromID8*04-EAAU84*ToHS1*08*Arrow End1*1</Layout>
 <Name2>NewConn:1-3J9N7</Name2>
 <ForBranchName>NewConn:1-3J9N7</ForBranchName>
 <ParentType>DEFAULT</ParentType>
 <ToStepName>Get List Price</ToStepName>
</RepositoryWfBranchConnector>
- <RepositoryWfBranchConnector>
 <Comments />
 <FromStepName>Get List Price</FromStepName>
 <Inactive>N</Inactive>
 <Layout>@0*0*9*0*0*0*9*LineWidth1*24*FrHS1*22*P16*2 8 8 324*NPts1*22*P06*244
 324*ToID8*04-EAATP6*FromID8*04-EAATQ4*ToHS1*08*Arro w End1*1</Layout>
 <Name2>NewConn:1-3J9N9</Name2>
 <ForBranchName>NewConn:1-3J9N9</ForBranchName>
 <ParentType>CONNECTOR</ParentType>
 <ToStepName>End</ToStepName>
 </RepositoryWfBranchConnector>
 </ListOfRepositoryWfBranchConnector>
</RepositoryWorkflowProcess>
</ListOfRepositoryWorkflowProcess>
</SiebelMessage>
```

External Integration of Siebel Pricer Using Pricer as a Service

You use Pricer as an external service if you do not want to use the Siebel interface, but only the Siebel pricing engine. The Siebel pricing engine must receive data in a particular format, which is called the property set. Integration involves getting data in that format and passing it to the pricing procedure.

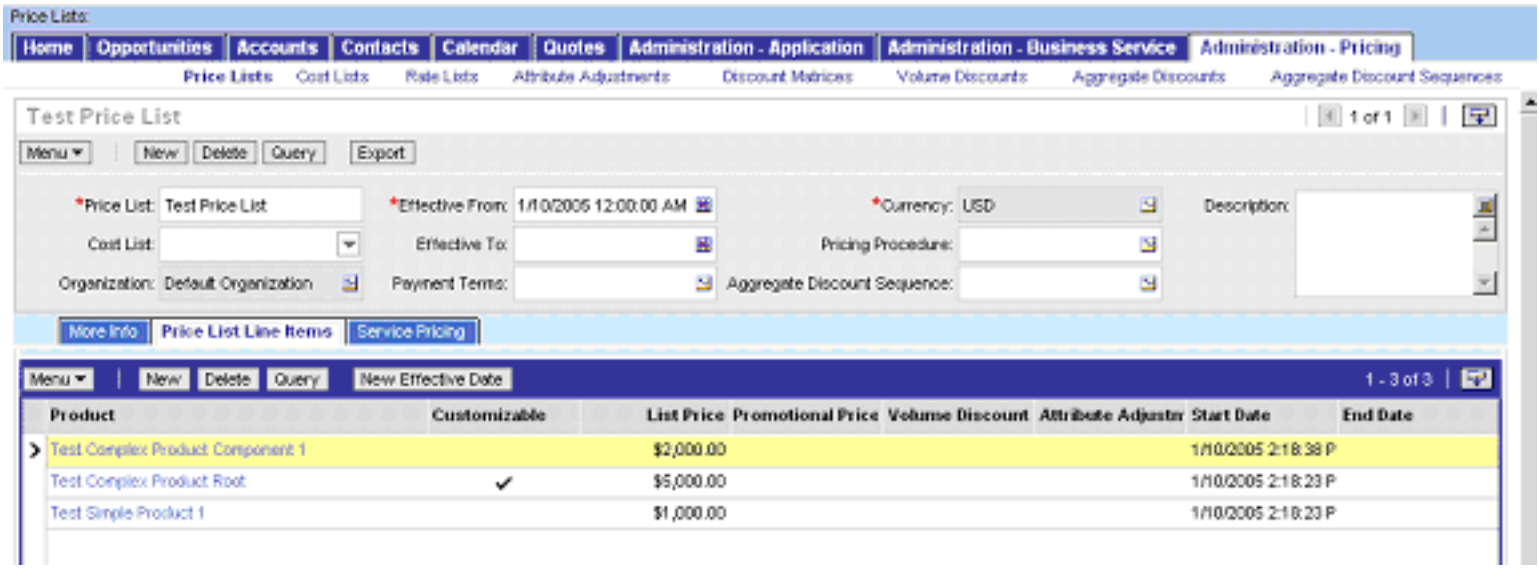
A typical flow for this type of external integration is as follows:

- The property set is the standard data format for integration. It is created by a Business Service Script that is based on the data from the external application. However, the property set format must be understood by the pricing procedure. For the property set format expected by the pricing procedure, see *Sample Pricing Procedure for Internal Integration* and *Sample XML Output Property Set for External Integration*.
- The desired pricing procedure is called with the property set generated in the previous step. Pricing information is updated on the property set.

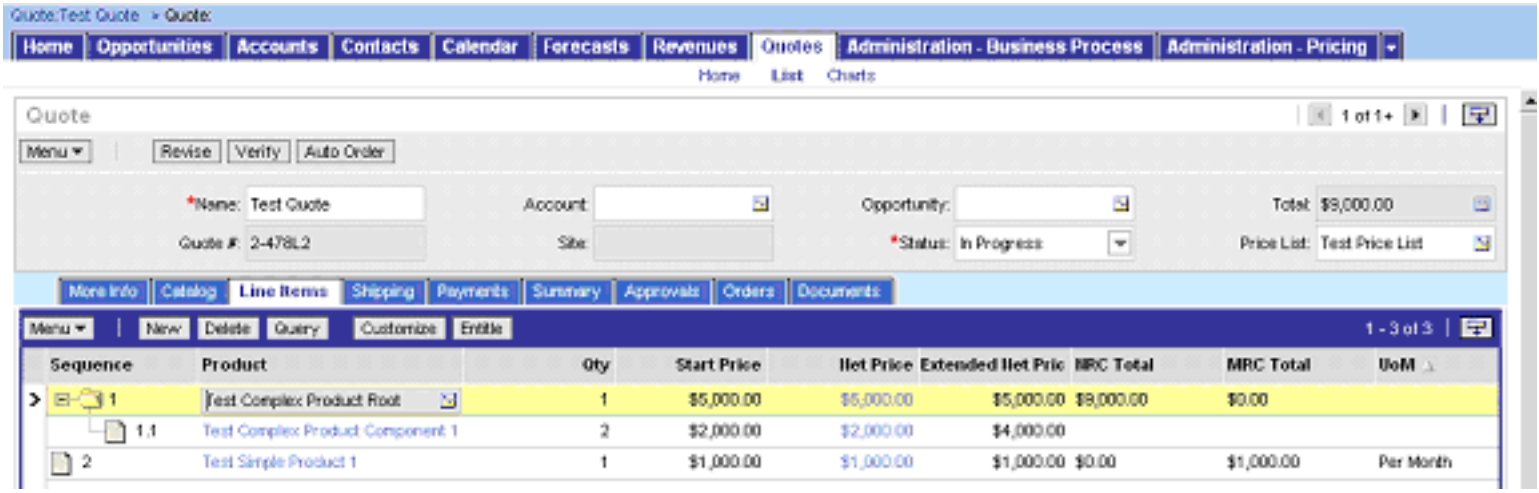
- The Business Service Script extracts updated information from the property set and synchronizes back to the external data source.

The following example illustrates this process.

A simple product and a complex product with a component product are created. A price list is then created and the products are associated with it, as shown in the following figure.



Products are added to the quote, as shown in the following figure. As expected, prices are stamped on each of the products, and all the steps under Internal Integration are performed to produce the prices.



Now, instead of using the Siebel application interface to create the quote, the following script is used to create the quote:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var fileName = "C:\\log.txt";
 var xmlFile = "C:\\pricer_input.xml";
 var procedureName = "Dynamic Pricing Procedure";

 var psXMLSvcInputs = TheApplication ().NewPropertySet ();
 var psPricingProcedureInputs = TheApplication ().NewPropertySet ();
 var psPricingProcedureOutputs = TheApplication ().NewPropertySet ();
```

```
// In this example, Read the Property Set from an XML file.
// The Property Set contains a simple product and a complex product to be priced.
// This can be generated from other sources as well, such as from
// external systems, as long as the Property Set conformed to the format
var xmlSvc = TheApplication().GetService ("EAI XML Read from File");
psXMLSvcInputs.SetProperty ("FileName", xmlFile);
xmlSvc.InvokeMethod ("ReadPropSet", psXMLSvcInputs, psPricingProcedureInputs);

// After the property set is created, the "Dynamic Pricing Procedure" WF
// is invoked. The output would contain the simple product and the complex
// product with updated pricing information.
var wfSvc = TheApplication().GetService ("Workflow Process Manager");
psPricingProcedureInputs.SetProperty ("ProcessName", procedureName);

wfSvc.InvokeMethod ("RunProcess", psPricingProcedureInputs,
psPricingProcedureOutputs);
return (CancelOperation);
}
```

The first function in this script contains three essential processes for external integration:

- A property set is created by a Business Service Script based on the data from the external application. In this example, the property set is loaded from an XML file. The property set format must be understood by the pricing procedure. For the required property set format, see the step that follows this one.

  ```
  // In this example, Read the Property Set from an XML file.
  // The Property Set contains a simple product and a complex product to be priced.
  // This can be generated from other sources as well, such as from
  // external systems, as long as the Property Set conformed to the format
  var xmlSvc = TheApplication().GetService ("EAI XML Read from File");
  psXMLSvcInputs.SetProperty ("FileName", xmlFile);
  xmlSvc.InvokeMethod ("ReadPropSet", psXMLSvcInputs, psPricingProcedureInputs);
  ```

- The desired Pricing Workflow is called with the property set generated in the previous step. Pricing information is updated on the property set.

  ```
  var procedureName = "Dynamic Pricing Procedure";
  // ...
  var wfSvc = TheApplication().GetService ("Workflow Process Manager");
  psPricingProcedureInputs.SetProperty ("ProcessName", procedureName);
  wfSvc.InvokeMethod ("RunProcess", psPricingProcedureInputs,
  psPricingProcedureOutputs);
  ```

- The Business Service Script extracts updated information from the property set and synchronizes back to the external data source. See the following examples for more information:
- *Sample XML Input Property Set for External Integration*
- *Sample XML Output Property Set for External Integration*

Sample XML Input Property Set for External Integration

The following code sample shows some sample XML Input Property Set for external integration:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set
EscapeNames="true"?><PropertySet
><Context
 Local_spcLanguage="ENU"
 Local_spcTime_spcZone="(GMT-08:00) Pacific Time (US &amp;amp; Canada); Tijuana"
 CPScope="Whole"
 CPCollapseAll=""
 RowScope="Current"
```

```
 Mode="Quote"
>Default Pricing Variable Map - Context<Header
 Currency_spcCode="USD"
 Discount_spcPercent=""
 Exchange_spcDate="02/09/2005"
 Price_spcList_spcId="2-478KT"
>2-478L2</Header
></Context
><Row_spcSet
 Local_spcLanguage="ENU"
 Local_spcTime_spcZone="(GMT-08:00) Pacific Time (US &amp;amp; Canada); Tijuana"
 CPScope="Whole"
 CPCollapseAll=""
 RowScope="Current"
 Mode="Quote"
>Default Pricing Variable Map - Row Set<Line_spcItem
 NRC_spcCxTotal="0"
 Class_spcId=""
 Unit_spcof_spcMeasure=""
 Service_spcPrice_spcMethod=""
 Id="2-478LF"
 Current_spcDiscount_spcId=""
 Exclude_spcPricing_spcFlag="N"
 Extended_spcQuantity_spcRequested="1"
 Sales_spcService_spcFlag="N"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Product_spcId="2-478K6"
 Min_spcPrice=""
 Next_spcDiscount_spcId=""
 Discount_spcPercent=""
 Prod_spcProm_spcId=""
 Rollup_spcAmount="0"
 Covered_spcProduct_spcId=""
 Covered_spcAsset_spcProduct_spcId=""
 Parent_spcItem_spcId=""
 Product_spcPort_spcId=""
 Net_spcPrice=""
 List_spcPrice=""
 Promotion_spcItem_spcFlag="N"
 Upsell=""
 Root_spcBundle_spcFlag="N"
 Price_spcType="One-Time"
 Covered_spcLine_spcItem_spcId=""
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcPrice=""
 Max_spcPrice=""
 Discount_spcAmount=""
 Exchange_spcDate="02/09/2005"
 Quantity="1"
 Action_spcCode="Add"
 Usage_spcAsset_spcProduct_spcId=""
 Price_spcList_spcId=""
 Start_spcPrice=""
 Service_spcPrice_spcPercent=""
 MRC_spcCxTotal="0"
>2-478LF<Attributes
>Default Pricing Variable Map - XA</Attributes
></Line_spcItem
><Line_spcItem
 NRC_spcCxTotal=""
 Class_spcId=""
 Unit_spcof_spcMeasure=""
 Service_spcPrice_spcMethod=""
 Id="2-478NH"
 Current_spcDiscount_spcId=""
```

```
 Exclude_spcPricing_spcFlag="N"
 Extended_spcQuantity_spcRequested="2"
 Sales_spcService_spcFlag="N"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Product_spcId="2-478JP"
 Min_spcPrice=""
 Next_spcDiscount_spcId=""
 Discount_spcPercent=""
 Prod_spcProm_spcId=""
 Rollup_spcAmount="0"
 Covered_spcProduct_spcId=""
 Covered_spcAsset_spcProduct_spcId=""
 Parent_spcItem_spcId="2-478LF"
 Product_spcPort_spcId="2-478KD"
 Net_spcPrice=""
 List_spcPrice=""
 Promotion_spcItem_spcFlag="N"
 Upsell=""
 Root_spcBundle_spcFlag="N"
 Price_spcType="One-Time"
 Covered_spcLine_spcItem_spcId=""
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcPrice=""
 Max_spcPrice=""
 Discount_spcAmount=""
 Exchange_spcDate="02/09/2005"
 Quantity="2"
 Action_spcCode="Add"
 Usage_spcAsset_spcProduct_spcId=""
 Price_spcList_spcId=""
 Start_spcPrice=""
 Service_spcPrice_spcPercent=""
 MRC_spcCxTotal=""
>2-478NH<Attributes
>Default Pricing Variable Map - XA</Attributes
></Line_spcItem
><Line_spcItem
 NRC_spcCxTotal="0"
 Class_spcId=""
 Unit_spcof_spcMeasure="Per Month"
 Service_spcPrice_spcMethod=""
 Id="2-478OD"
 Current_spcDiscount_spcId=""
 Exclude_spcPricing_spcFlag="N"
 Extended_spcQuantity_spcRequested="1"
 Sales_spcService_spcFlag="N"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Product_spcId="2-478J6"
 Min_spcPrice=""
 Next_spcDiscount_spcId=""
 Discount_spcPercent=""
 Prod_spcProm_spcId=""
 Rollup_spcAmount="0"
 Covered_spcProduct_spcId=""
 Covered_spcAsset_spcProduct_spcId=""
 Parent_spcItem_spcId=""
 Product_spcPort_spcId=""
 Net_spcPrice=""
 List_spcPrice=""
 Promotion_spcItem_spcFlag="N"
 Upsell=""
 Root_spcBundle_spcFlag="N"
 Price_spcType="Recurring"
 Covered_spcLine_spcItem_spcId=""
```

```
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcPrice=""
 Max_spcPrice=""
 Discount_spcAmount=""
 Exchange_spcDate="02/09/2005"
 Quantity="1"
 Action_spcCode="Add"
 Usage_spcAsset_spcProduct_spcId=""
 Price_spcList_spcId=""
 Start_spcPrice=""
 Service_spcPrice_spcPercent=""
 MRC_spcCxTotal="0"
>2-478OD<Attributes
>Default Pricing Variable Map - XA</Attributes
></Line_spcItem
></Row_spcSet
></PropertySet
>
```

Sample XML Output Property Set for External Integration

The following code sample shows some sample XML Output Property Set for external integration:

```
<?xml version="1.0" encoding="UTF-8"?><?Siebel-Property-Set
EscapeNames="true"?><PropertySet
><Context
 Local_spcLanguage="ENU"
 Local_spcTime_spcZone="(GMT-08:00) Pacific Time (US &amp; Canada); Tijuana"
 CPScope="Whole"
 CPCollapseAll=""
 RowScope="Current"
 Mode="Quote"
>Default Pricing Variable Map - Context<Header
 Price_spcList_spcName="Test Price List"
 Currency_spcCode="USD"
 Discount_spcPercent=""
 Is_spcPrice_spcList_spcExpired_spcFlag="N"
 Exchange_spcDate="02/09/2005"
 Is_spcPrice_spcList_spcNot_spcEffective_spcFlag="N"
 Price_spcList_spcId="2-478KT"
>2-478L2</Header
></Context
><Row_spcSet
 Local_spcLanguage="ENU"
 Local_spcTime_spcZone="(GMT-08:00) Pacific Time (US &amp; Canada); Tijuana"
 CPScope="Whole"
 CPCollapseAll=""
 RowScope="Current"
 Mode="Quote"
>Default Pricing Variable Map - Row Set<Line_spcItem
 Extended_spcQuantity_spcRequested="1"
 Temp_spcList_spcPrice="5000"
 Discount_spcPercent=""
 Rollup_spcAmount="4000"
 Upsell=""
 Covered_spcLine_spcItem_spcId=""
 Temp_spcMRC_spcRollup_spcItem_spcPrice="0"
 Quantity="1"
 Price_spcList_spcId="2-478KT"
 MRC_spcCxTotal="0"
 Temp_spcNRC_spcCxTotal="9000"
 Skip_spcPricing_spcFlag="N"
```

```
 Effective_spcPromotion_spcId=""
 Product_spcId="2-478K6"
 Prod_spcProm_spcId=""
 Temp_spcNRC_spcLine_spcTotal="9000"
 Covered_spcAsset_spcProduct_spcId=""
 Temp_spcMRC_spcLine_spcTotal="0"
 Max_spcPrice=""
 Root_spcPrice_spcList_spcItem_spcId="2-478KW"
 MSRP_spcPrice=""
 Temp_spcNRC_spcNet_spcPrice="5000"
 Bundle_spcSequence_spcId=""
 Net_spcPrice="5000"
 List_spcPrice="5000"
 Root_spcBundle_spcFlag="N"
 Price_spcType="One-Time"
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcof_spcMeasure=""
 Temp_spcMRC_spcNet_spcPrice="0"
 Id="2-478LF"
 Purchase_spcPrice=""
 Parent_spcItem_spcId=""
 Rollup_spcItem_spcPrice="9000"
 Exchange_spcDate="02/09/2005"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Root_spcExchange_spcDate="02/09/2005"
 Cost_spcPrice=""
 Covered_spcProduct_spcId=""
 Product_spcPort_spcId=""
 Root_spcPromotion_spcId=""
 Unit_spcPrice=""
 Discount_spcAmount=""
 Dynamic_spcMatrix=""
 Root_spcPrice_spcList_spcId="2-478KT"
 Root_spcCurrency_spcCode="USD"
 Exclude_spcPricing_spcFlag="N"
 List_spcPrice_spcWaterfall_spcText="List price from &apos;Test Price List&apos;
price list"
 Price_spcList_spcItem_spcId="2-478KW"
 Min_spcPrice=""
 Temp_spcMRC_spcCxTotal="0"
 Service_spcPrice_spcMethod=""
 Next_spcDiscount_spcId=""
 Promotion_spcItem_spcFlag="N"
 Header_spcDiscount_spcAmount="0"
 Action_spcCode="Add"
 NRC_spcCxTotal="9000"
 Class_spcId=""
 List_spcPrice_spcWaterfall="5000"
 Current_spcDiscount_spcId=""
 Sales_spcService_spcFlag="N"
 Rollup_spcAmount_spcMRC="0"
 Temp_spcParent_spcQuantity="1"
 Volume_spcDiscount_spcId=""
 Root_spcProduct_spcId="2-478K6"
 Pricing_spcAdjustment_spcAmount="0"
 Usage_spcAsset_spcProduct_spcId=""
 Start_spcPrice="5000"
 Service_spcPrice_spcPercent=""
>2-478LF<Attributes
>Default Pricing Variable Map - XA</Attributes
><Net_spcPrice_spcWaterfall
><Waterfall
 Adjustment_spcAmount="5000"
 Price="5000"
 Currency_spcCode="USD"
```

```
 Text="List price from &apos;Test Price List&apos; price list"
 Adjustment_spcType="Price Override"
></Waterfall
></Net_spcPrice_spcWaterfall
></Line_spcItem
><Line_spcItem
 Extended_spcQuantity_spcRequested="2"
 Temp_spcList_spcPrice="2000"
 Discount_spcPercent=""
 Rollup_spcAmount="0"
 Upsell=""
 Covered_spcLine_spcItem_spcId=""
 Temp_spcMRC_spcRollup_spcItem_spcPrice="0"
 Quantity="2"
 Price_spcList_spcId="2-478KT"
 MRC_spcCxTotal=""
 Temp_spcNRC_spcCxTotal="4000"
 Skip_spcPricing_spcFlag="N"
 Effective_spcPromotion_spcId=""
 Product_spcId="2-478JP"
 Prod_spcProm_spcId=""
 Temp_spcNRC_spcLine_spcTotal="4000"
 Covered_spcAsset_spcProduct_spcId=""
 Temp_spcMRC_spcLine_spcTotal="0"
 Max_spcPrice=""
 Root_spcPrice_spcList_spcItem_spcId="2-478KW"
 MSRP_spcPrice=""
 Temp_spcNRC_spcNet_spcPrice="2000"
 Bundle_spcSequence_spcId=""
 Net_spcPrice="2000"
 List_spcPrice="2000"
 Root_spcBundle_spcFlag="N"
 Price_spcType="One-Time"
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcof_spcMeasure=""
 Temp_spcMRC_spcNet_spcPrice="0"
 Id="2-478NH"
 Purchase_spcPrice=""
 Parent_spcItem_spcId="2-478LF"
 Rollup_spcItem_spcPrice="2000"
 Exchange_spcDate="02/09/2005"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Root_spcExchange_spcDate="02/09/2005"
 Cost_spcPrice=""
 Covered_spcProduct_spcId=""
 Product_spcPort_spcId="2-478KD"
 Root_spcPromotion_spcId=""
 Unit_spcPrice=""
 Discount_spcAmount=""
 Dynamic_spcMatrix=""
 Root_spcPrice_spcList_spcId="2-478KT"
 Root_spcCurrency_spcCode="USD"
 Exclude_spcPricing_spcFlag="N"
 List_spcPrice_spcWaterfall_spcText="List price from &apos;Test Price List&apos;
price list"
 Price_spcList_spcItem_spcId="2-478KY"
 Min_spcPrice=""
 Temp_spcMRC_spcCxTotal="0"
 Service_spcPrice_spcMethod=""
 Next_spcDiscount_spcId=""
 Promotion_spcItem_spcFlag="N"
 Header_spcDiscount_spcAmount="0"
 Action_spcCode="Add"
 NRC_spcCxTotal=""
 Class_spcId=""
```

```
 List_spcPrice_spcWaterfall="2000"
 Current_spcDiscount_spcId=""
 Sales_spcService_spcFlag="N"
 Rollup_spcAmount_spcMRC="0"
 Temp_spcParent_spcQuantity="1"
 Volume_spcDiscount_spcId=""
 Root_spcProduct_spcId="2-478K6"
 Pricing_spcAdjustment_spcAmount="0"
 Usage_spcAsset_spcProduct_spcId=""
 Start_spcPrice="2000"
 Service_spcPrice_spcPercent=""
>2-478NH<Attributes
>Default Pricing Variable Map - XA</Attributes
><Net_spcPrice_spcWaterfall
><Waterfall
 Adjustment_spcAmount="2000"
 Price="2000"
 Currency_spcCode="USD"
 Text="List price from &apos;Test Price List&apos; price list"
 Adjustment_spcType="Price Override"
></Waterfall
></Net_spcPrice_spcWaterfall
></Line_spcItem
><Line_spcItem
 Extended_spcQuantity_spcRequested="1"
 Temp_spcList_spcPrice="1000"
 Discount_spcPercent=""
 Rollup_spcAmount="0"
 Upsell=""
 Covered_spcLine_spcItem_spcId=""
 Temp_spcMRC_spcRollup_spcItem_spcPrice="1000"
 Quantity="1"
 Price_spcList_spcId="2-478KT"
 MRC_spcCxTotal="1000"
 Temp_spcNRC_spcCxTotal="0"
 Skip_spcPricing_spcFlag="N"
 Effective_spcPromotion_spcId=""
 Product_spcId="2-478J6"
 Prod_spcProm_spcId=""
 Temp_spcNRC_spcLine_spcTotal="0"
 Covered_spcAsset_spcProduct_spcId=""
 Temp_spcMRC_spcLine_spcTotal="1000"
 Max_spcPrice=""
 Root_spcPrice_spcList_spcItem_spcId="2-478KX"
 MSRP_spcPrice=""
 Temp_spcNRC_spcNet_spcPrice="0"
 Bundle_spcSequence_spcId=""
 Net_spcPrice="1000"
 List_spcPrice="1000"
 Root_spcBundle_spcFlag="N"
 Price_spcType="Recurring"
 Product_spcLine_spcId="No Match Row Id"
 Unit_spcof_spcMeasure="Per Month"
 Temp_spcMRC_spcNet_spcPrice="1000"
 Id="2-478OD"
 Purchase_spcPrice=""
 Parent_spcItem_spcId=""
 Rollup_spcItem_spcPrice="0"
 Exchange_spcDate="02/09/2005"
 Covered_spcProduct_spcNet_spcPrice=""
 Currency_spcCode="USD"
 Root_spcExchange_spcDate="02/09/2005"
 Cost_spcPrice=""
 Covered_spcProduct_spcId=""
 Product_spcPort_spcId=""
 Root_spcPromotion_spcId=""
```

```
 Unit_spcPrice=""
 Discount_spcAmount=""
 Dynamic_spcMatrix=""
 Root_spcPrice_spcList_spcId="2-478KT"
 Root_spcCurrency_spcCode="USD"
 Exclude_spcPricing_spcFlag="N"
 List_spcPrice_spcWaterfall_spcText="List price from &apos;Test Price List&apos;
price list"
 Price_spcList_spcItem_spcId="2-478KX"
 Min_spcPrice=""
 Temp_spcMRC_spcCxTotal="1000"
 Service_spcPrice_spcMethod=""
 Next_spcDiscount_spcId=""
 Promotion_spcItem_spcFlag="N"
 Header_spcDiscount_spcAmount="0"
 Action_spcCode="Add"
 NRC_spcCxTotal="0"
 Class_spcId=""
 List_spcPrice_spcWaterfall="1000"
 Current_spcDiscount_spcId=""
 Sales_spcService_spcFlag="N"
 Rollup_spcAmount_spcMRC="0"
 Temp_spcParent_spcQuantity="1"
 Volume_spcDiscount_spcId=""
 Root_spcProduct_spcId="2-478J6"
 Pricing_spcAdjustment_spcAmount="0"
 Usage_spcAsset_spcProduct_spcId=""
 Start_spcPrice="1000"
 Service_spcPrice_spcPercent=""
>2-478OD<Attributes
>Default Pricing Variable Map - XA</Attributes
><Net_spcPrice_spcWaterfall
><Waterfall
 Adjustment_spcAmount="1000"
 Price="1000"
 Currency_spcCode="USD"
 Text="List price from &apos;Test Price List&apos; price list"
 Adjustment_spcType="Price Override"
></Waterfall
></Net_spcPrice_spcWaterfall
></Line_spcItem
></Row_spcSet
></PropertySet
```

Siebel Pricing Manager API Reference

Earlier versions of the Pricing Manager API reference included a user property reference and a business service reference. These user properties are obsolete because of the new order management infrastructure. If your integration needed these user properties, then you must rework the integration using pricing procedures, signals, variable maps and other features of the new order management infrastructure. For more information, see *Siebel Order Management Infrastructure Guide* .

The methods are retained for backward compatibility, even though the API is no longer the best practice for external integration. For preferred integration methods, see *Integration Methods for Siebel Pricer*.

This topic describes the following:

- Pricing Manager methods internal to the application in the first table in this topic.
- Pricing Manager methods that are published in Web Tools in the second table in this topic.

- Parameters for the CalculatePriceExternal Method in the third table in this topic.

The following table describes the Pricing Manager methods that are internal to Siebel application.

Method Name	External or Internal	Purpose	Input Arguments	Output Arguments
CalculatePrice	Internal	Used for calling Siebel Pricer from other business components.	None	None
CalculatePriceAll	Internal	CalculatePrice for all quote Item discounts and calculate aggregate price for the whole quote.	None	None

The following table describes the Pricing Manager methods that are published in Web Tools.

Method Name	External or Internal	Purpose	Input Arguments	Output Arguments
CalculateFinalDiscountExte	External	Calculates aggregate pricing for external application user. A multiple item record must be passed as the argument. Assumes the CalculatePrice function was called for each individual line.	Price list ID, current total (current total item price), Pricer VBC BO Name (current business component for pricing operations), list total (total list price of all items).	Pricing Comments, Final Discount Price, Pricing Comments and Discount Amount on the children Property set.
CalculatePrice	Internal	Used for calling Oracle's Siebel Pricer from other business components.	Not applicable	Not applicable
CalculatePrice- External	External	Single Item repricing for external users.	See the following table.	Not applicable
GetFinalDiscount	Obsolete - will be removed	Obsolete.	Not applicable	Not applicable
GetProductListPrice	External or Internal	Retrieve List Prices and other information from Price List Item for a product and price list combination.	Price List Id, Product Id.	Currency Code, Min Sale Price, Max Sale Price, Price Book Id, Current Price, List Price, List Price Field x (Auxiliary outputs start with List Price Field, then followed by List Price Field 1, and so on.)
Reload Cache	External or Internal	Obsolete	None.	None.

Method Name	External or Internal	Purpose	Input Arguments	Output Arguments
SetPriceList	External or Internal	To set the price list at the profile attribute level. This is currently used in Sales to override the Price list ID set by the session.	No input because this set is based on the global profile attributes in eSales.	No output arguments as described for input arguments.
Start	Internal	Hidden - not for use by external users. Obsolete	None.	None.

The following table describes the Parameters for the CalculatePriceExternal Method.

Name	Optional	Type	Explanation
Currency Code	N	Output	The currency code for the price
Current Discount	N	Output	The current volume discount.
Current Discount ID	N	Output	The current volume discount ID.
Current Price	N	Output	The final price.
List Price	N	Output	The start price.
Next Discount	N	Output	The next available item for volume discount.
Next Discount ID	N	Output	None
Original List Price	Y	Input	None
Price List ID	N	Input	Not applicable
Pricing Comments	N	Output	Not applicable
Product ID	N	Input	Not applicable
Quantity	N	Input	Not applicable
Service Price %	N	Output	Indicates the percentage used to determine the price of the Service Product, where the Service Price is set to a percentage of the product price.
Upsell Message	N	Output	The volume discount upsell message.

Name	Optional	Type	Explanation
Volume Discount ID	N	Output	The volume discount ID (a copy of the ID in Quote).

Workflows for External Pricing API

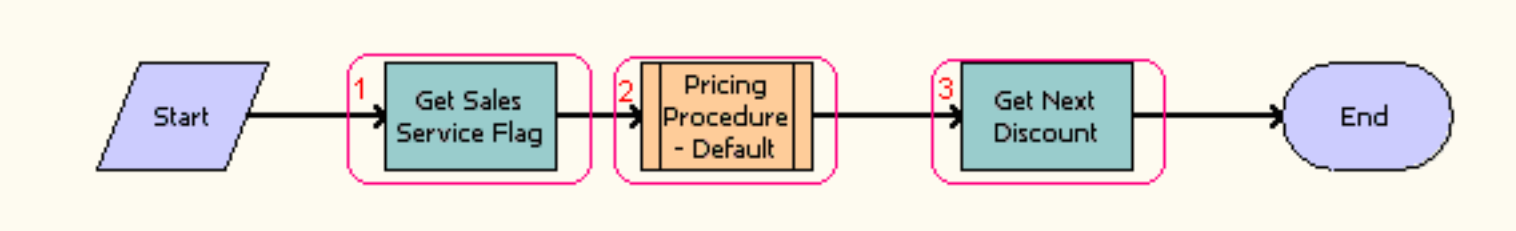
Earlier releases expose APIs on the Pricing Manager Business Service for external integration. To support backward compatibility, Oracle continues to provide support for these APIs.

Because of changes to the pricing engine for version 7.8 or later, the internals of these APIs are replaced by the Row Set Transformation Toolkit Business Service, which is driven by workflows. External API support uses the following workflows:

- *CalculatePriceExternal Workflow* (for Business Service method CalculatePriceExternal)
- *Get Config Item Price Workflow* (for Business Service method GetConfigItemPrice)
- *Get Product List Price Workflow* (for Business Service method GetProductListPrice)

CalculatePriceExternal Workflow

The CalculatePriceExternal Business Service method is an external interface that takes in product line items in a property set and provides pricing information for each of the product line items. The following figure illustrates the CalculatePriceExternal workflow process.



Workflow Description

This workflow does the following:

1. **Get Sales Service Flag.** Determines if the product is a service product. Service products have different pricing requirements from regular sales products. For more information, see *Dynamic Pricing Procedure Workflow*.
2. **Pricing Procedure - Default.** Calls the Dynamic Pricing Procedure workflow as a subprocess that handles all pricing operations.
3. **Get Next Discount.** Populates Volume Discount upsell messages.

The following table lists the steps in the CalculatePriceExternal workflow with the business service and method that is called by each.

Name	Type	Business Service	Method	Sub Process	Description
Get Sales Service Flag	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Determines if the product is a service product.
Dynamic Pricing Procedure	Sub Process	Not applicable	Not applicable	Dynamic Pricing Procedure	Calls the Dynamic Pricing Procedure workflow as a subprocess that handles all the pricing operations.
Get Next Discount	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Populates Volume Discount upsell messages.

Get Config Item Price Workflow

The GetConfigItemPrice Business Service Method is an API that fetches price adjustments from Customizable Product Pricing Designer for a price list, product, or product port combination. The following figure illustrates the GetConfigItemPrice workflow process.



Workflow Description

This workflow does the following:

1. **Get List Price.** Gets the List Price for the input product and price list, which provides the basis for subsequent Pricing Designer adjustments.
2. **Get Root Price List Item Id and Customizable Product Adjustment.** These steps retrieve the Customizable Product Pricing Designer adjustment.

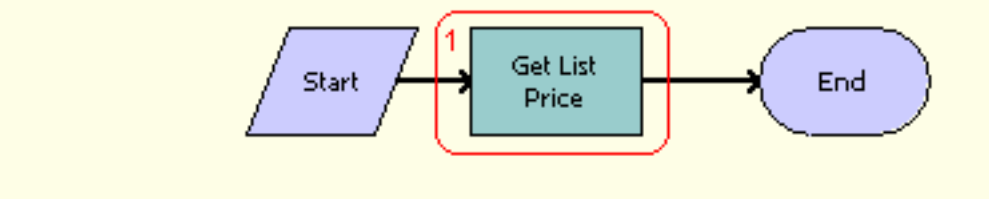
The following table lists the steps in the GetConfigItemPrice workflow with the business service and method that is called by each.

Name	Type	Business Service	Method	Sub Process	Description
Get List Price	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Gets the List Price for the input product and price list.
Get Root Price List Item Id	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Retrieves the Root Price List Item Id for each item for use in the next step.

Name	Type	Business Service	Method	Sub Process	Description
Customizable Product Adjustment	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Queries in Price Config Item Buscomp for Pricing Designer adjustments.

Get Product List Price Workflow

The GetProductListPrice Business Service method is an API that gets the price and some other attributes from the Price List Item Business Component for a price list and product combination. The following figure illustrates the GetProductListPrice workflow process.



This workflow gets the list price for the input product and price list. Also, it retrieves other information such as Min Price, Max Price, Volume Discount Id, Dynamic Matrix Id, and so on.

The following table lists the steps in the GetProductListPrice workflow with the business service and method that is called.

Name	Type	Business Service	Method	Sub Process	Description
Get List Price	Business Service	Row Set Transformation Toolkit Service	Simple Look-Up Transform	Not applicable	Gets List Price for the input product and price list.

Cache Configuration Settings

Parameters specifying cache size typically determine the number of records cached locally. When the specified cache limit is reached, the next new record must be loaded. As a rule, you should not change the cache sizes specified for your application unless advised to do so by your server administrator.

Configuration settings for pricing can be set at the server level and at the remote client level. The standard configuration file includes a section called pricing cache which includes the following parameter settings and corresponding default values:

```
[PricingCache]
PriceListCacheLifeTime = -1
PriceListItemCacheLifeTime = -1
PricerMappingCacheSize = 100
PricerPriceItemCacheSize = 100
PricerPriceListCacheSize = 20
```

```
PricerPriceModelCacheSize = 50
```

The following table describes the pricing cache parameters in more detail.

Parameter Name	Description
PriceListCacheLifeTime	This is the lifetime of the price list header information where the integer value represents seconds, the time unit of measure. The price list header information must be reloaded after this number of seconds has passed. If all other factors remain the same, then there is a trade-off between refresh-time and loading speed. If you decrease a LifeTime value, then the result is more frequent refresh of data and a longer loading time for data. This is represented by the parameter PARAM_SUBSYS_PRICE_LIFE-TIME. The default value for PriceListItemCacheLifeTime is -1.
PriceListItemCacheLifeTime	This is the lifetime of the price list line item and volume discount information where the integer value represents seconds, the time unit of measure. The price list line item information must be reloaded after this number of seconds has passed. This is represented by the parameter PARAM_SUBSYS_PRICE_ITEM_LIFE_TIME. The default value for PriceListItemCacheLifeTime is -1.
PricerMappingCacheSize	Not Applicable. **Note:** This is the number of records to be cached for each mapping factor. The default is 100. If this value is set to zero, then mapping information will not be cached, and the database will immediately be searched.
PricerPriceItemCacheSize	Not applicable. **Note:** This applies to the price list line item records for each price list cache.
PricerPriceListCacheSize	Not applicable. **Note:** This applies to the price list records at the header level.
PricerPriceModelCacheSize	Not applicable. **Note:** This applies to the model cache.

The default value for PriceListCacheLifeTime and PriceListItemCacheLifeTime is -1, which means that the cache remains unrefreshed indefinitely. You can change this value to any number between zero and 86400. Zero means to turn off the cache. 86400 is the number of seconds in a full day.

If you want to reload new list prices for products without having to log out of and back in again to your Siebel Business Application, then you must set the following pricing cache parameters to, for example, 60:

```
PriceListCacheLifeTime = 60

PriceListItemCacheLifeTime = 60
```

Setting both parameters (PriceListCacheLifeTime and PriceListItemCacheLifeTime) to low values results in more frequent refreshing and slows down performance, but keeps the pricing information up to date.