SIEBEL® eBusiness 7

PRICING ADMINISTRATION GUIDE

VERSION 7.5, REV. A

12-EECGU0

MARCH 2003

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Printed in the United States of America

Proprietary Information
Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Introduction

Chapter 1. Overview

Chapter 2. Creating and Assigning Price Lists

Chapter 3. Creating and Assigning Rate Lists

Chapter 4. Creating and Using Cost Lists

Chapter 5. Creating Volume Discounts

Chapter 6. Creating Promotions and Deals

Chapter 7. Creating Pricing Reports

Chapter 8. Setting Up Component-Based Pricing

Chapter 9. Setting Up Attribute-Based Pricing

Chapter 10. Creating Pricing Models and Pricing Factors

Chapter 11. Matrix-Based Pricing Factors

Chapter 12. Bundling and Aggregate Pricing Factors

Chapter 13. Script-Based Pricing Factors

Chapter 14. Using Mappings in Pricing Factors

Chapter 15. Using the Pricing Factor Flow Chart Designer

Chapter 16. Testing and Validating Pricing Models

Appendix A. Siebel ePricer Deployment and Integration

Introduction

This guide provides information about using the Siebel ePricer application.

This book will be useful to people whose title or job description matches one of the following:

Siebel Application Administrators Persons responsible for planning, setting up, and maintaining Siebel applications.

Siebel Application Developers Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.

Siebel ePricer Administrators Persons responsible for administering the Siebel ePricer application.

Siebel ISS Application Administrators Persons responsible for administering one of the Siebel applications in the Siebel ISS suite.

Siebel System Administrators Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

To use some features of Siebel ePricer effectively, you should have some familiarity with Siebel Tools; you should be able to navigate to business objects, business components, and specific fields, and you should understand how data related to pricing topics is organized. To implement script-based pricing factors, you should be able to write scripts in one of the Siebel-supported scripting languages, or be able to specify requirements for such scripts to a person with scripting skills.

How This Guide Is Organized

This guide provides information necessary to implement, configure, and monitor pricing within Siebel eBusiness Applications. If you work with applications that use pricing functionality, you will need to know about pricing configuration, administration, and reporting.

Revision History

Pricing Administration Guide, Version 7.5, Rev. A

March 2003 Bookshelf

Table 1. Changes Made in Rev. A for March 2003 Bookshelf

Topic	Revision
Chapter 3, “Creating and Assigning Rate Lists”	Added this chapter.
Chapter 4, “Creating and Using Cost Lists”	Added this chapter.

Overview 1

This chapter gives a general overview of pricing administration for Siebel applications, and it describes how you log on as Siebel Administrator. It also discusses the processing order of price adjustments.

This chapter covers the following topics:

- "About Pricing Administration" on page 16
- "Logging On as the Siebel Administrator" on page 18
- "The Processing Order of Price Adjustments" on page 19

About Pricing Administration

Siebel ePricer allows you to work with:

- **Price Lists.** Create price lists and assign them to users. Price lists are created separately from products, so you can use different prices for the same products. For more information, see Chapter 2, "Creating and Assigning Price Lists."

- **Volume Discounts.** Give customers a discount if they buy in volume. For example, if a customer buys more than one hundred units, the customer can get a 10% discount. For more information, see Chapter 5, "Creating Volume Discounts."

- **Promotions and Deals.** Create promotional deals for product. For example, if you are introducing a new product, you can offer a 10% discount during the first month that the product is available. For more information, see Chapter 6, "Creating Promotions and Deals."

- **Pricing Reports.** Create a variety of reports to track your pricing policies. For more information, see Chapter 7, "Creating Pricing Reports."

- **Customizable Products.** Siebel ePricer allows you to define pricing for two types of customizable products:

  - **Component-Based Products.** Customers can customize products by choosing components. For example, when a customer buys a computer, the customer can customize it by choosing to add peripherals such as a CD drive. For more information, see Chapter 8, "Setting Up Component-Based Pricing."

  - **Attribute-Based Products.** Customers can customize products by choosing attributes. For example, when a customer buys a tee-shirt, the customer can customize it by choosing its color. For more information, see Chapter 9, "Setting Up Attribute-Based Pricing."

In addition, Siebel ePricer lets you work with pricing models and pricing factors, which allow you to create more complex pricing adjustments. Siebel ePricer allows you to create the following types of pricing factors:

- **Single.** Includes a single rule that applies a price adjustment to individual line items in a quote, order, agreement, or component-based product. For example, you can give the customer a discount if the extended price of a line item is more than $1,000. For more information, see Chapter 10, "Creating Pricing Models and Pricing Factors."

- **Matrix-Based.** Equivalent to several single-type pricing factors. You create a Siebel business component with records specifying multiple rules for price adjustments, and the pricing factor searches this business component to determine which price adjustments to apply. You can import information into this business component from a pricing table in another application, such as an Excel spreadsheet. For more information, see Chapter 11, "Matrix-Based Pricing Factors."

- **Bundling.** Allows you to apply price adjustments that depend on the selection of bundles of products. For example, if a customer buys a dining room table and four chairs, the customer can get a 10% discount on a sideboard. For more information, see Chapter 12, "Bundling and Aggregate Pricing Factors."

- **Aggregate.** Allows you to apply price adjustments that depend on the entire quote or order. For example, you can give a customer a 10% discount if the total price of a quote is over $100,000. For more information, see Chapter 12, "Bundling and Aggregate Pricing Factors."

- **Script-Based.** Allows you to use a script (in Siebel eScript, Siebel Visual Basic, or other scripting languages) to define more complex pricing factors. These might include complex mathematical calculations, multiple business components, or a compound rule on multiple fields in the same business components. For more information, see Chapter 13, "Script-Based Pricing Factors."

Siebel ePricer also lets you price items that are not products in your Siebel application. For example, you can price literature or products that exist in other systems but not in your Siebel application. For more information about nonproduct pricing, see the section "External Integration" on page 273.

Logging On as the Siebel Administrator

The Siebel installer creates a Siebel administrator account that can be used to perform the tasks described in this guide. For more information, see *Siebel Server Administration Guide* and the Siebel server installation guide for the operating system you are using.

To log on as the Siebel administrator, start the application and log on using the user name and password assigned by your database administrator. Generally, the Siebel administrator connects to the server database.

CAUTION: Do not perform system administrative functions on your local database. Although there is nothing to prevent you from doing this, it can create problems such as data conflicts, an overly large local database, and a large number of additional transactions to route.

The Processing Order of Price Adjustments

If you create more than one type of price adjustment for a price list, the order in which the 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% volume discount if the customer buys more than ten. If a customer qualifies for both discounts:

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

This section describes the processing order of pricing adjustments. You should refer to this section if you create a price list that includes more than one of the pricing adjustments described in later chapters.

Because the processing order for component-based products is different, it will be described separately.

Siebel ePricer applies pricing adjustments for all products except component-based products in the following order:

- **The Promotional Price Takes Precedence over the List Price.** If a price list includes a promotional price for a product (in the Promotional Price field), all price adjustments apply to the promotional price. If a price list does not include a promotional price for a product, all price adjustments apply to the list price (in the List Price field).
- **Promotional Deals.** If there is a deal-based price, it modifies the list or promotional price. This deal-based price is used as the basis of any other price adjustments.
- **Attribute Pricing.** Pricing adjustments based on the attributes that users select for customizable products are processed before other pricing adjustments. Customizable products are individual line items in the quote or order, and the price of each must be resolved before Siebel ePricer can decide whether other types of price adjustments apply.
- **Volume Discounts.** Volume discounts are applied after attribute pricing is resolved and before pricing models are applied.

- **Pricing Models for Line Item Products.** Pricing models for line item products are applied after other price adjustments. You determine the order in which the pricing factors of a pricing model are applied by specifying a sequence number of a logical flow. However, you must specify the following order:
    - **Single Pricing Factors.** All pricing factors that apply to a single line of the quote or order must be applied first. These include single type, matrix -based type, and script-based type pricing factors.
    - **Aggregate Pricing Factors.** All pricing factors that apply to multiple lines of the quote or order must be applied last. These include bundling type and aggregate type pricing factors.

Siebel ePricer applies pricing adjustments for component-based products in the following order:

- **The Promotional Price of Components Takes Precedence Over the List Price.** If a price list includes a promotional price for a product (in the Promotional Price field) that is a component of the product, all price adjustments apply to the promotional price. If a price list does not include a promotional price for a product that is a component of the product, all price adjustments apply to the list price (in the List Price field).
- **Attribute Pricing.** Components of the product can have attributes. For example, the product may be a computer, and you may be able to select the color of the mouse and keyboard. Pricing adjustments based on the attributes of components that users select for customizable products are processed before other pricing adjustments. The attribute-based price of each component must be resolved before Siebel ePricer can decide whether other types of price adjustments apply.
- **Component-Based Pricing Model.** Pricing models that apply to customizable products are processed before other pricing adjustments. Customizable products are individual line items in the quote or order, and the price of each must be resolved before Siebel ePricer can decide whether other types of price adjustments apply.

- **Volume Discounts for the Entire Component-Based Product.** After the pricing of the entire component-based product is resolved, Siebel ePricer can apply volume discounts that are based on purchase of the entire component-based product. For example, a user may be purchasing computers, which are a customizable product. After resolving the price of each computer based on which components the user chooses, Siebel ePricer can apply a volume discount based on the total number of computers the customer purchases.

- **Pricing Models for the Entire Component-Based Product.** Pricing models for all line item products, including component-based products, are applied after other price adjustments. You determine the order in which the pricing factors of a pricing model are applied by specifying a sequence number of a logical flow. However, you must specify the following order:

  - **Single Pricing Factors.** All pricing factors that apply to a single line of the quote or order must be applied first. These include single type, matrix -based type, and script-based type pricing factors.

  - **Aggregate Pricing Factors.** All pricing factors that apply to multiple lines of the quote or order must be applied last. These include bundling type and aggregate type pricing factors.

Creating and Assigning Price Lists 2

This chapter begins with an overview of price lists. Then it describes the most common way of creating a new price list. It also describes how to assign price lists to users.

This chapter covers the following topics:

About Price Lists

A price list is a set of standard prices for products or services. Price lists generally contain the first prices that a buyer sees, before any pricing adjustments take place.

A product must be assigned a price in a price list to appear 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 ePricer. You should test the price list thoroughly before applying more advanced price adjustments.

This chapter looks at 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 EIM

After you create a price list, you must assign it to users, to make it control their prices.

Price List Prerequisites

Before you create any price list, you must create the products the prices apply to. You must:

- Define products
- Associate each product to its product lines

For more information, see *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 this other data before you can use these optional fields.

For example, if you want a price list to use volume discounts, 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:

- Assets, cost lists, and services. For more information, see *Product Administration Guide*, *Applications Administration Guide* and *Siebel Field Service Guide*.
- Rate lists. For more information, see Chapter 3, “Creating and Assigning Rate Lists” in this book.
- Volume discounts. For more information, see Chapter 5, “Creating Volume Discounts,” in this book.
- Product classes and attributes and their domains. For more information, see *Product Administration Guide*.
- Component-based pricing adjustments. For more information, see Chapter 8, “Setting Up Component-Based Pricing,” in this book.

- Attribute-based pricing adjustments. For more information, see Chapter 9, “Setting Up Attribute-Based Pricing,” in this book.
- Pricing models. For more information, see Chapter 10, “Creating Pricing Models and Pricing Factors,” in this book.

The Process of Creating a New Price List

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

You can create different price lists for the same products. For example, you can create one price list with the wholesale prices for all of your products, and another 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.

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

1. "Creating a Price List Header" on page 27
2. "Creating Price List Line Items" on page 29

NOTE: Creating a price list line item for a customizable product is more complex than creating a line item for a simple product. Consult with a Siebel Administrator before creating price lists for customizable products. For more information about customizable products, see Chapter 8, "Setting Up Component-Based Pricing."

NOTE: This book covers price lists for products. For information about creating Service Pricing line items, see *Siebel Field Service Guide*.

Creating a Price List Header

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, which are described later in this chapter.

To define a price list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

2 In the Price Lists list, click New.

A new Price List record appears.

3 Enter information in the new record and in More Info form, as described in Table 1.

Table 1. Price List Header Fields

Field	Explanation
Name	Required. Enter unique, meaningful name for this price list. Overwrite any system-generated name. If many price lists will appear in your system, use a consistent naming convention to name all your Price Lists.
Cost List	Optional. Select the cost list to be associated with this price list.
Currency	Required. Enter the currency for all amounts that will be in this price or be associated with it. All cost lists, terms, pricing models, and pricing adjustments associated with this price list or with line items in it must use this currency.
Shipping Method	Optional. Select one default shipping method to be associated with this price list. This appears as the default Shipping Method when a runtime user creates a quote or order that specifies this price list.
Pricing Model	Optional. Select the pricing model to be associated with this price list. Only one pricing model can be linked to this price list (at the header data level) at a time. The model's effective dates must fall within the effective dates of the price list for the model to be effective.
Payment Terms	Optional. 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 runtime user creates a Quote or Order that specifies this price list.
Organization	Required. Select all the organization which may have prices controlled by this price list. For more information see, "Assigning a Price List to a User" on page 45.
Shipping Terms	Optional. Select one default shipping charge protocol to be associated with this price list. This appears as the default Shipping Terms data when a runtime user creates a quote or order that specifies this price list.

Table 1. Price List Header Fields

Field	Explanation
Effective From	Required. Enter the date and time when this price list will become effective. By default, ePricer assigns the current system date and time when you first create the Price List record.
Effective To	Optional. 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	Required. By default, ePricer assigns the user name used to log in to the current session in which this Price List record is created.
Last Updated	Required. By default, ePricer assigns the current system date and time when you most recently saved this Price List record.
Integration ID	Optional. If this system-generated ID appears, it can be used as a unique identifier for this price list to assist with system integration tasks. The Integration ID field is used for system integration with external systems. This field is populated by Siebel EAI connectors.
Description	Optional. Enter a description of the purposes, unique characteristics and limitations of this price list.

Creating Price List Line Items

A price list line item contains price data for a specific product. It includes the product name and related information.

When you create a price list line item, you first select the product (which may be a service). This product provides the price list line item its name. Adding a price list line item is equivalent to adding a product to a price list.

In a given price list, you may have only one line item for a product.

If you have multiple price lists, the same product can have a line item in each price list. 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 customizable product 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 customizable products, see Chapter 8, "Setting Up Component-Based Pricing."

To define a new price list line item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

   The Price List 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 Price List Line Items view tab.
4. In the Price List Line Items list, click New.

   The Add Products dialog box appears.

5. In the Add Products dialog box, use the Query or Find options, if necessary, to locate the product you want to add to the price list. Select the product record and click Add.

   The product that you selected appears in the new Price List Line Item record.

6. If the product is a customizable product with component products, see the instructions in Chapter 8, "Setting Up Component-Based Pricing."

7 Complete the fields in the Price List Item Detail form, as described in Table 2.

Table 2. Price List Line Item Fields

Field	Explanation
Name	Required. The product name that you select from the Add Products dialog box. A price list line item uses this product name as its unique ID.
Customizable	Read-only; based on product definition. If this check box is selected, it indicates that the product has components.
List Price	Optional, but recommended. Enter a list price, the standard price used for most transactions and a most commonly used target price for price adjustments. If a promotional price is entered, it is used instead of this list price. If the list price is omitted and no promotional price or price override price is provided, the product may be offered with no price—the equivalent of a zero 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, then the list price is validated against them.
Promotional Price	Optional. If you enter a promotional price, it will be used instead of the list price in all functions that do not explicitly identify the list price as the target price.
Attribute Pricing	Optional. Specifies the attribute pricing table that provides attribute-based pricing adjustments for this line item product. This attribute pricing table is also applied to this same product when it is a component in a customizable product. This attribute pricing table uses attribute data from the Class to which the product belongs.
Customizable Product Pricing Model	Optional. If this line item is for a customizable product that has components, you can select the name of a pricing model developed for this line item.
Description	Optional. Enter a description of this line item and its unique qualities, especially if important for ePricer processing.

Table 2. Price List Line Item Fields

Field	Explanation
Product Line	Optional. Read-only. This value is taken from the product information. Specifies the line of products to which the line item product belongs.
Part Number (Part #)	Optional. Read-only. This value is derived from the product information. Specifies the part number assigned to the line item product.
Cost	Optional. Enter the cost of the line item product using the standard cost formula for this price list. This can be used as the target price for some price adjustment calculations.
MSRP	Optional. Enter the MSRP for reference purposes. In some cases, this can be used as the target price for price adjustment calculations.
Purchase Price	Optional. Enter the purchase price of the line item product. This can be used as the target price for some price adjustment calculations.
Unit of Measure	Optional. Select the unit of measure for the product item.
Minimum Price	Optional but recommended. If a minimum price value is provided, ePricer does not offer a lower price for this line item, unless a runtime 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. Minimum price does not apply to components of customizable products.
Maximum Price	Optional. If a maximum price value is provided, ePricer does not offer a higher price for this line item, unless a runtime 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 customizable product, then this maximum will apply to that component product as well.
Service Price %	Optional. Enter a service price percentage used for determining the service pricing associated with this line item. The service price percentage is applied to the target specified in the Service Pricing Method field.

Table 2. Price List Line Item Fields

Field	Explanation
Service Pricing Method	Optional. Specify a service pricing target for the service price percentage. This is used to determine the service pricing associated with this line item.
% Margin	Optional. Read-only. This is a calculated field that calculates margin based on the following formula: ((List or promotional price) – cost)/List*100

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

Creating a Price List Line Item for a Simple Product Bundle

A simple 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.

When you create a simple bundle, the product bundle is treated as a single product. You assign a price to the bundle as a whole in the same way that you assign a price to an individual product.

For information about creating simple bundle products, see *Product Administration Guide*.

You can create more complex bundle discounts by using bundling pricing factors, described in Chapter 12, “Bundling and Aggregate Pricing Factors.” The two methods of creating bundling discounts have the following difference:

- **Simple bundle prices.** Allow you to set a single price for the entire bundle. For example, a dining table and four chairs might cost $500. Define the bundle in the Product Administration screen. Then set the price for the entire bundle in the Pricing Administration screen, as you would set the price for any other product.

- **Bundle type pricing factors.** Allow you to set discounts for individual components of the bundle. For example, if you buy a dining table you might get a 10 percent discount on four chairs. You do not define the bundle in the Product Administration screen, and you do not define a list price for the entire bundle. Instead, you use the pricing factor to associate products that are in the bundle and to define discounts on their list prices.

NOTE: Before performing the following procedure, follow the instructions in "To define a new price list line item" on page 30.

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

1. Create a new product line item, as described in "Creating Price List Line Items" on page 29.

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 that suits different types of customers or markets. For example, you might use a price list named United States Price List as the starting basis for creating new price lists for different countries, locations, organizations, currencies, accounts, or demographics.

First you copy the price list, which retains the original price list and recreates all of its data in a price list with a new name. The new price list must have a different name.

Then you modify the data in the copy as necessary.

To modify the data in more elaborate ways, you should use the Transform button, described in the following section, rather than modifying the data manually. For example, you can use the Transform function to convert all the currencies in a price list to a different currency.

NOTE: Do not edit the Currency or Pricing Model field when you copy and modify a price list. To modify these fields, use the Transform function, described in "Copying and Transforming a Price List" on page 37.

To copy and modify a price list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.
2. In the Price Lists list, select the price list you want to copy.

3 In the Price Lists list, click the menu button, 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, described in Table 1 on page 28 and the price list line items, described in Table 2 on page 31.

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

NOTE: Do not edit the Currency or Pricing Model field directly. To modify these fields, use the Transform function, described in the following section.

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 allows you to 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% 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 model to a price list, changing some or all product prices depending on the model. For more information about pricing models, see Chapter 10, “Creating Pricing Models and Pricing Factors.”
- Change certain types of attribute-based price adjustments by the percentage you specify, using the Deep Copy option. This option applies only to attribute-based pricing for customizable products. For more information, see Chapter 9, “Setting Up Attribute-Based Pricing.”

The Transform function transforms the list price as well as the promotional price fields when the transformations do not include a pricing model. If a transformation includes a pricing model, then it applies to the list price field only. You can change the promotional prices manually.

NOTE: If you use a pricing model to transform a price list, that pricing model should contain only single, matrix-based, or script-based type factors; it should not include any bundling or aggregate type factors. The purpose of a pricing model in the Transform operation is to modify the prices of single price list line items. Bundling or aggregate type pricing factors cannot do this because they apply to multiple line items.

CAUTION: To transform a price list, first make a copy of the Price List, and then transform the copy.

To transform a price list

1. Copy the price list you want to transform, using the procedure in the section "Copying and Modifying a Price List" on page 35, but do not modify the price list.

2. Click the Transform button.

   The Transform Price List form appears.

3 Enter information in the Transform Price List form, as described in Table 3, and then click the Transform button.

Table 3. Transform Price List Form Fields

Field	Explanation
Transformed Price List Name	Required. The name of the transformed price list. The default is the name of the copied price list.
Currency Code	Required. The currency to be used for prices in the transformed price list. The default is the currency of the copied price list; if you are not changing currencies, keep the default.
Exchange Date	Required, if you are converting currencies. Enter the date for the currency exchange rate. The system will use the exchange rate on that day to convert currency.
Prorate %	Required. 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%, you would enter 130. To decrease all list prices by 20%, you would enter 80. The default is 100%, and if you do not want to change prices, you should keep this default.
Deep Copy Attribute Tables	Optional. Check this box to the attribute-based pricing tables associated with this price list and apply the Prorate % and currency conversion fields to them. For more information, see "Using the Deep Copy Option to Transform Price Lists" on page 40.
Transforming Pricing Model	Optional. Select the name of the pricing model to be applied to all line items in the specified price list. For more information about pricing models, see Chapter 10, "Creating Pricing Models and Pricing Factors."

Table 3. Transform Price List Form Fields

Field	Explanation
Account	Optional. If you used the Transforming Pricing Model field to specify a pricing model that includes pricing factors based on an account, you must use this field to select the specific account.
Additional Data for Transformation	Optional. The extra space on this form is provided so that you configure the form to add drop-down lists for selecting additional fields to be used by Pricing Factors, similar to the list provided for specifying Accounts. For more information, see "Configuring the Transform Price List Form to Use Additional Data" on page 42.

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

Using the Deep Copy Option to Transform Price Lists

The Deep Copy option of the Transform Price list form is only relevant to price lists that include customizable products that use attribute-based pricing. For more information, see Chapter 9, "Setting Up Attribute-Based Pricing."

If you use the Deep Copy option, ePricer makes modified attribute-based pricing data available to the new price list by:

- Copying the attribute-based pricing tables attached to the original price list items into new attribute tables attached to the new price list
- Applying the Prorate % and the currency conversion rate to the new attribute table items (as appropriate)

The new, deep-copied attribute-based pricing tables are identical to the old tables except that:

- Each new row has a new unique row ID appended to it

- If a row specifies an adjustment type of Markup Amount, Discount Amount, or Price Override, then the Adjustment Amount value in that row is multiplied by the Prorate % and converted to the new currency (as appropriate).

NOTE: Deep Copy does not have any effect on attribute-based pricing table item adjustments if the price adjustment type is Markup % or Discount %. These adjustments apply to an amount in the price list that is already adjusted by the prorating, so there is no reason to change them again with Deep Copy.

These new price adjustments are applied in addition to the fundamental price list changes that result from the Prorate % and currency conversion during the transformation process.

The changed attribute tables also affect any customizable product component product that is linked to an attribute pricing table. Component products use the attribute tables identified in their product line items in the price list.

Examples: Transforming List Prices with Deep Copy

In the first example, the attribute-based adjustment is of the Markup Amount type.

A Widget product normally sells for $100. If a runtime user selects the Chrome Plating attribute, the list price of the $100 Widget increases to $110, based on a $10 amount markup (Adjustment Type = Markup Amount, Adjustment Value =$10).

If you enter a Prorate % of 120 and select the Deep Copy option, then the attribute-adjusted price of this chrome-plated Widget in the transformed price list would be $132, the result of two Prorate % adjustments: (120% x $100) + (120% x $10)=$120 + $12=$132.

In the next example, the attribute-based adjustment is of the Discount Amount type.

Again, a Widget product normally sells for $100. If a runtime user selects the Reconditioned attribute, the list price of the $100 Widget to decreases to $70, based on a $30 amount discount (Adjustment Type = Discount Amount, Adjustment Value =$30).

If you enter a Prorate % of 120 and select the Deep Copy option, then the attribute-adjusted price of this reconditioned Widget in the transformed price list would be $84, the result of two Prorate % adjustments: (120% x $100) – (120% x $30) = $120 – $36 = $84.

In both cases, the 120% change is applied both to the original price and to the price adjustment.

If you do not use Deep Copy, the Transform function still copies all of the existing attribute table adjustments, and the attribute pricing tables are not changed.

Configuring the Transform Price List Form to Use Additional Data

Using Siebel Tools, you can configure the Transform Price List form by adding new drop-down lists that function like the Account drop-down list.

The Account drop-down list and other drop-down lists that you might add to the form are only relevant to price lists that use pricing models. For more information about pricing models, see Chapter 10, “Creating Pricing Models and Pricing Factors.”

You must use the Account drop-down list to specify the account, if you used the Transforming Pricing Model field to specify a pricing model that includes pricing factors based on an account.

For example, many pricing models include logical statements such as: “If account name = A. K. Parker, then apply 10% discount.” To apply such a pricing model within the Transform process, you would have to select A. K. Parker in the account selector. Otherwise, the pricing model will not have the access to the required Business Component information.

If you use pricing models that include Pricing Factors based on other Business Components, you must add additional drop-down lists that work in the same way.

If you do not use this sort of drop-down lists, the transformation can only check data in the following Business components:

- Price List
- Price List Copy
- Price List Item

Use Siebel Tools to add drop-down lists to the Transform Price List form, that let you access Siebel business component data required by the pricing models you use regularly. Once you have customized this form, the added drop-down lists will always be present.

The basic procedures for adding the new field to the Price List Transformation form are listed below. Expertise with Siebel Tools is required to configure the product in this way.

To add a new business component field drop-down list to the Price List Transformation form

1. In Siebel Tools, add a new field called "Customer Type" to the Price List Transformation applet. For details on adding a new field to a business component, see *Siebel Tools Reference*.

2. In the Pricing Model Manager, create a pricing model to be used for the Transform processing session.

3. In the pricing model to be used for transformations, add the single type pricing factors that refer to this new field. For details, see the following pricing factor example.

4. Run the Transform process, using the instructions in the section "Copying and Transforming a Price List" on page 37.

Example: Adding Product-Line Specific Prices when Transforming a Price List

You want to create a new price list your product line of monitors, reducing prices by 10%.

First, create a pricing factor with the fields listed below. This pricing factor identifies customers whose Product Line field has been set to Monitors and discounts their prices by 10%.

Pricing factor fields:

- Pricing Factor Type: Single
- Business Component: Price List Item Copy
- Business Component Field: Product Line

- Operator: =
- Field Value: Monitors
- Type of Calculation: % Discount
- Adjustment Value: 10.

Then, use Siebel Tools to add a Product Line drop-down list to the Transform Price List form.

When you transform the price list, use this pricing factor and choose Monitors in the Product Line drop-down list. The resulting price list will include the 10% price reduction for monitors.

Importing Price Lists Using Siebel EIM

It is common to use Siebel Enterprise Integration Manager (EIM) to import Price Lists into the Siebel application, to deal with large numbers of Price Lists that have been 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. This allows for organizational visibility of Price Lists and associated Price List Items.

If you use Siebel EIM to import Price Lists and Price List Items, 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 will become enterprise visible and will be routed to all users regardless of organization.

For more information about using Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

Assigning a Price List to a User

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

- Organization 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. A price list can be assigned to many organizations, and many price lists can be assigned to an organization.
- Account controls which price list is active for the user. The active price list is the price list associated with the user's account record. You can only associate one price list with an account record, so the user's prices are controlled by one price list.

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.

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

To assign a price list to a user

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.
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 Organizations dialog box appears.
4. If the user's organization is not already listed in this dialog box, click New and add it. Then click OK.
5. From the application-level menu, choose View > Site Map > Accounts > All Accounts Across Organizations.
6. In the Accounts list, select the user's account.

7. In the More Info form, click the show more button to expand the form.
8. In the Price list field of the More Info form, click the select button.

   The Pick Price List dialog box appears.

9. In the Pick Price List dialog box, select the price list you want to use for that account, and click OK.

Creating and Assigning Rate Lists 3

This chapter begins with an overview of rate lists. Then it describes the most common ways of creating a new rate list. It also describes how to assign rate lists to users.

This chapter covers the following topics:

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, 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 a price in a rate list, it can appear in catalogs, just as it can if you assign it a price in a price list.

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 are used to manage projects that use professional services, in order to determine the billing rates for professional services for that project. For more information, see *Siebel Professional Services Automation Guide*.
- **Field Service.** Rate lists are used to 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 Pricing Administration 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, you may charge different rates to customers who use the help for short-term projects and long-term projects, or you may charge different rates to larger and smaller customers. You would create different rate lists and assign them to these customers to make their rates visible to them.

The Process of Creating a New Rate List

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 items records contain rates for specific resources.

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

1. "Creating the Resources as Products"
2. "Creating a Rate List Record"
3. "Creating Rate List Line Items" on page 52

Creating the Resources as Products

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, 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 checkbox when you create the product in the Product Administration screen.

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

Creating a Rate List Record

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

To create a rate list record

1. From the application-level menu, choose View > Site Map > Pricing Administration > Rate List.
2. In the Rate Lists list, click New.

   A new Rate List record appears.
3. Enter information in the new record and in the More Info form, as described in Table 4.

Table 4. Rate List Header Fields

Field	Explanation
Name	Required. Enter a unique, meaningful name for this rate list.
Description	Optional. Enter a description of the rate list for your own use.
Cost List	Optional. Select the cost list to be associated with this rate list.
Effective From	Required. 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	Optional. Enter the date and time when this rate list will become ineffective. After this time, Siebel applications will not be able to use this rate list. If no value is entered here, the rate list remains in effect indefinitely.
Updated By	Required. 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	Required. By default, the application assigns the current system date and time when you most recently saved this Rate List record.
Organization	Required. Select all the organization which may have rates controlled by this rate list. For more information see, "Assigning a Rate List to a User" on page 57.

Creating Rate List Line Items

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.

In a given rate list, you may have only one line item for each resource.

If you have multiple rate lists, 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, your Canadian rate list may have the rate charged for the resource in Canada, and so on.

To create a rate list line item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Rate List.
2. In the Rate List list, select the rate list to 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. In the Add Position Types dialog box, use the Query or Find options, if necessary, to locate the type of resource you want to add to the rate list. Select the product record and click Add.

   The product that you selected appears in the new Rate List Line Item record.
6. Complete the fields of the Rate List Item record, as described in Table 5.

Table 5. Rate List Line Item Fields

Field	Explanation
Resource	Required. The type of resource that you selected from the Add Position Types dialog box is displayed here. 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.
Contract Rate Per Hour	Enter the hourly rate you charge for this resource as a contractor.

Table 5. Rate List Line Item Fields

Field	Explanation
Premium Rate Per Hour	Enter the premium hourly rate you charge for this employee, if applicable. For example, this premium rate may be charged to customers who get special service.
Material Mark-Up	Enter the percentage mark-up 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 that suits 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 you copy the rate list. This retains the original rate list and creates a new rate list with the same data.

Then you modify the data in the copy as necessary.

To modify the data in more elaborate ways, you should use the Transform button, described in the following section, rather than modifying the data manually. For example, you can use the Transform function 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" on page 55.

To copy and modify a rate list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Rate List.
2. In the Rate Lists list, select the rate list you want to copy.
3. In the Rate Lists list, click the menu button, 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, described in Table 4 on page 51 and the rate list line items, described in Table 5 on page 52.

   Either edit the fields directly or click the menu button, then click Change Records, and use 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 following section.

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 allows you to 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% 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.

Caution: To transform a rate list, first make a copy of the Rate List, and then transform the copy.

To transform a rate list

1. Copy the rate list you want to transform, using the procedure in the section "Copying and Modifying a Rate List" on page 54, but do not modify the rate list.
2. Click the Transform button.

   The Transform Rate List form appears.
3. Enter information in the Transform Rate List form, as described in Table 6, and then click the Transform button.

Table 6. Transform Rate List Form Fields

Field	Explanation
Transformed Rate List Name	Required. The name of the transformed rate list. The default is the name of the copied rate list that you are transforming.
Currency Code	Required. The currency to be used for prices in the transformed rate list. The default is the currency of the copied rate list; if you are not changing currencies, keep the default.

Table 6. Transform Rate List Form Fields

Field	Explanation
Exchange Date	Required if you are converting currencies. Enter the date for the currency exchange rate. The system will use the exchange rate on that day to convert currency.
Prorate %	Required. 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%, you would enter 130. To decrease all list prices by 20%, you would enter 80. The default is 100%, and if you do not want to change prices, you should keep this default.

Assigning a Rate List to a User

The rate list that an employee sees depends on that employee's organization. The user 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. One organization can have only one rate list, which determines the rates that its employees see.

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

To assign a rate list to a user

1. From the application-level menu, choose View > Site Map > Pricing Administration > Rate List.
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, click New and add it. Then click OK.

Creating and Using Cost Lists 4

This chapter begins with an overview of cost lists. Then it discusses how to create costs lists and use them with either price lists or rate lists to view your profit margin.

This chapter covers the following topics:

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 in order to determine the profit margin.

The margin is the difference between the list or promotional price of a product or service and the cost of that product or service. The margin is stated as a percentage that is calculated as follows:

```
Margin = (List or Promotional Price - Cost) / List Price
```

For example, if a product's list price is $100 and its cost is $80, the margin is 20 percent. If a product's list price is $100 and its cost is $60, the margin is 40 percent.

There are three cost methods that can be used to determine the cost of a product or service:

- **Standard Cost.** A predetermined operating cost, which is compared with the actual cost in order to measure the performance of a given department or operation.
- **Last Cost.** 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.** A costing method that allows the user to maintain the cost manually.
- **Average Cost.** A costing method that calculates product cost as the average (arithmetic mean) of all the purchase costs of an inventoried product.

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.

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

The Process of Implementing Cost Lists

To implement costs lists, you go through the following process:

1. "Creating the Price List or Rate List the Cost List Will Reference" on page 61
2. "Creating the Cost List Record" on page 61
3. "Creating the Cost List Line Item Records" on page 63
4. "Entering the Indirect Costs" on page 64
5. "Associating the Cost List with a Price List or Rate List" on page 65
6. "Updating a Price List's Costs" on page 66
7. "Viewing the Product Cost and Margin" on page 66

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

As a prerequisite for creating a cost list, you should create the price list or the rate list that the cost list will be associated with. 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 Chapter 2, "Creating and Assigning Price Lists."

For more information about creating rate lists, see Chapter 3, "Creating and Assigning Rate Lists."

Creating the Cost List Record

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. From the application-level menu, choose View > Site Map > Pricing Administration > Cost List.
2. In the Cost Lists list, click New.

3 Enter information in the new Cost List record and the More Info form, as described in Table 7.

Table 7. Cost List Record Fields

Field	Explanation
Cost List	Required. Enter a unique name for this cost list. If you will have many cost lists, it is important to use meaningful names and a consistent naming convention for them all.
Description	Optional. Enter a description of this cost list for your own use.
Effective From	Required. 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	Optional. Enter the date and time when this cost list will become ineffective. After this time, Siebel applications will not be able to use this price list.
Cost Method	Select one of the four cost methods, Standard, Average, Next, or Last. These cost methods are described in "About Cost Lists" on page 60.
Created By	The application enters the user name of the person creating this record.
Created	The application enters the date and time when the record was created.
Organization	Click the select button and use the Organizations dialog box to add the organizations that will use this cost list.
Updated By	The application enters the user name of the last person who updated this record.
Updated	The application enters the date and time when the record was last updated.

Creating the Cost List Line Item Records

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" on page 60. Which one of these four cost methods is used depends on which cost method you choose in the Cost List record.

If the line item is a resource that bills by the hour, you can also enter a standard overtime rate and extended overtime rate for the resource. For example, you would do this if you are a temporary employment agency that places employees whom you pay by the hour and whom you must pay overtime if they work more than forty hours a week.

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

To create a cost list line item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Cost List.
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, you can use a Query or Find to locate the product you want to add to the price list. Select the product record and click Add.

6 Complete the fields in the Cost List Line Item record, as described in Table 8.

Table 8. Cost List Line Item Fields

Field	Explanation
Product	The name of the product or service that you selected from the Add Products dialog box is automatically entered here.
Part #	The part number of the product or service that you selected from the Add Products dialog box is automatically entered here. This number uniquely identifies the product or service.
Standard Cost	Enter the standard cost, described in "About Cost Lists" on page 60.
Last Cost	Enter the last cost, described in "About Cost Lists" on page 60.
Next Cost	Enter the next cost, described in "About Cost Lists" on page 60.
Average Cost	Enter the average cost, described in "About Cost Lists" on page 60.
Standard OT %	If the product is a resource that bills by the hour and qualifies for overtime pay, enter the standard overtime rate as a percentage of the base rate. For example, if the resource gets time and a half for overtime, enter 150.
Extended OT %	If the product is a resource that bills by the hour and qualifies for overtime pay, enter the extended overtime rate as a percentage of the base rate. For example, if the resource gets double pay for extended overtime, enter 200.

Entering the Indirect Costs

After entering the direct cost of the product, you can also enter indirect costs associated with product, such as your cost of ordering products, your cost of receiving products, and your cost of shipping products to customers.

These indirect costs apply to all the products in the cost list. Enter an average cost per unit for all your products.

To enter indirect costs

1 From the application-level menu, choose View > Site Map > Pricing Administration > Cost List.

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 Table 9, and continue to add these records until you have entered all indirect costs.

Table 9. Cost List Indirect Cost Fields

Field	Explanation
Expense Object	Select the type of indirect cost, such as shipment, receipt, or demonstration.
Cost Per Unit	Enter the cost per 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

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.

To associate a cost list with a price list

1 From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

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, in the Cost List field, select the cost list.

To associate a cost list with a rate list

1 From the application-level menu, choose View > Site Map > Pricing Administration > Rate List.

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, in the Cost List field, select the cost list.

Updating a Price List’s Costs

If the cost list is associated with a price list, you must update the cost for that price list in order 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. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.
2. In the Price Lists list, select the price list whose cost you want to update.
3. In the Price Lists list, click the menu button, and select Update Cost from the menu.

Viewing the Product Cost and Margin

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

The margin is displayed as a decimal, as described in “About Cost Lists” on page 60.

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

To view a product cost and margin in a price list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.
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 Scroll 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 products cost is in the Cost field and the profit margin is in the % Margin field of the Price List Line Item Detail form.

To view a product cost and margin in a rate list

1 From the application-level menu, choose View > Site Map > Pricing Administration >Rate List.

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.

Creating Volume Discounts 5

This chapter describes the two different types of volume discounts. Then it describes the process of setting up a volume discount. It also covers special features of volume discounts for customizable products.

This chapter covers the following topics:

About Volume Discounts

A volume discount is an adjustment to the price of a product based on the quantity of that product that is requested in the quote line or order line item. For example, you can set up a simple volume discount for a product that gives the customer:

- A 5% discount when the user requests five to 10 of the item
- A 10% discount when the user requests more than 10 of the item

Volume discounts apply to the quantity in the quote line or order line, not the total quantity in the entire quote or an entire order. If a user splits an order for a product into two or more lines on a quote, the volume discount calculation for that product would not be based on the total of the two lines.

In some circumstances volume discounts are not allowed. For example, if a sales agent manually enters a price discount, it is usual to have the Keep Discount option in force, which prevents volume discounts from being applied.

The volume discount applies to the list price, except:

- If the product is customizable, the volume discount applies to the price established by the customizable product price adjustments.
- If a promotional price has been defined for the product, the volume discount applies to the promotional price.

NOTE: If you use volume discounts with other types of price adjustments, the final price depends on the order in which Siebel ePricer applies the pricing adjustments. For more information, see “The Processing Order of Price Adjustments” on page 19.

About Simple and Tiered Volume Discounts

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

- **Simple.** Applies a discount percent to every instance of the item in a quote or order line item if the quantity meets the volume discount quantity requirement. If the line item quantity satisfies this requirement, the entire quantity qualifies for the volume discount.
- **Tiered.** Applies the discount percentage 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% discount when the quantity is five to 10 items
- Apply a 20% discount when the quantity is 11 to 20 items
- Apply a 30% discount when the quantity is 21 items or more

If this were a simple volume discount, and if the customer bought 23 items, the customer would get a discount of 30% on all 23 items.

If this were a tiered volume discount, and if the customer bought 23 items, the customer would get no discount on items one to four, a 10% discount on items five to 10, a 20% discount on items 11 to 20, and a 30 percent discount on items 21 to 23.

The Process of Setting Up a Volume Discount

To set up a volume discount, you go through the following process:

1. "Creating a Volume Discount Record"
2. "Creating Volume Discount Line Items" on page 74
3. "Linking the Volume Discount to a Product" on page 75

Creating a Volume Discount Record

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, and the start and end dates.

CAUTION: If you delete a volume discount record, 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. From the application-level menu, choose View > Site Map > Pricing Administration > Volume Discount.
2. In the Volume Discounts list, click New.

   A new Volume Discount record appears.
3. Enter information in the new record and in More Info form, as described in Table 10.

Table 10. Price List Header Fields

Field	Explanation
Name	Required. Enter a unique, meaningful name for this volume discount.
Discount Method	Required. Select a method from the picklist. The options are Simple and Tiered, as described in the section "About Simple and Tiered Volume Discounts" on page 71.
Start Date	Required. Enter the date when this volume discount will become effective.
End Date	Optional. Enter the date when this volume discount will become ineffective. If the volume discount will not expire, leave this field blank.
Description	Optional. Enter a description of the purposes and unique characteristics of this price list.
Integration ID	Optional. If this system-generated ID appears, it can be used as a unique identifier for this price list to assist with system integration tasks. The Integration ID field is used for system integration with external systems. This field is populated by Siebel EAI connectors.

Creating Volume Discount Line Items

Volume discount line items include information about quantities and percentage values of a volume discount.

Add a volume discount line item for each discount rate in the volume discount. For example, if you want to apply a 10% discount when the quantity is five to 10 and a 20% 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 will create the following errors:

- If you leave a gap in a simple discount, quantities that fall within the range of the gap receive no discount at all.
- If you leave a gap in a tiered discount, quantities that fall within the range of the gap default to the lower tier.
- If you create overlapping discount items, quantities that fall within the range of the overlap default to the lower tier.

How the discounts in the line items are applied depends on whether this is a simple or tiered volume discount, as described in the section "About Simple and Tiered Volume Discounts" on page 71.

To create a volume discount line item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Volume Discount.
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, click New.

   A new Volume Discount Line Item record appears.

5 Enter information in the new record, as described in Table 11.

Table 11. Price List Header Fields

Field	Explanation
Name	Required. Enter a unique name for this volume discount line item. This name appears in the quotes that your sales representatives create for customers, so you should use a name that describes the discount, such as “10% Discount for Buying 5 to 10.”
Min Qty	Required. Enter the minimum quantity that must be purchased to get this discount rate.
Max Qty	Optional. Enter the maximum quantity that gets this discount rate. If this field is blank, this discount rate applies to all quantities above the minimum quantity.
Discount %	Required. Enter the percentage discount for purchasing these quantities.
Description	Optional. Enter a description of this volume discount line item.

Linking the Volume Discount to a Product

After a volume discount record has been created and the line items that describe the discount have been defined, you link it to a product. After you link a volume discount to a product, the discount is applied automatically when a sales representative or other user enters the product in a quote.

To apply a discount to a product

1 From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

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 In the Price List Line Items list, in the Volume Discount field for a product, select the volume discount you want to apply to that product.

Volume Discounts and Customizable Products

Customizable products are discussed in Chapter 8, “Setting Up Component-Based Pricing,” and Chapter 9, “Setting Up Attribute-Based Pricing.” This section explains how volume discounts apply to customizable products.

Before ePricer executes volume discounts, it first completes any customizable product pricing adjustments, establishing a price for the configured version of the customizable product before any volume discounting takes place.

Volume discounts do not apply to component products within customizable products. If several customizable products in a quote include the same component product, those component products do not qualify for a volume discount based on the total quantity of the component product in the quote. The volume discounts are based on the quantity of each customizable product.

Volume discounts do apply to different configurations of a customizable product. For example, there is a volume discount for buying 20 Compaq Deskpro computers. If a customer buys 20 Compaq Deskpro computers and configures them all with different amounts of RAM and different peripherals, the customer gets the volume discount.

When you create a volume discount for a component-based customizable product, the discount amount is spread among components of the product. The price of each component is discounted in proportion to the overall discount.

Creating Promotions and Deals 6

This chapter begins with a general description of promotions and deals. Then it describes the process of creating promotions. Finally, it describes the customer's experience of promotions.

This chapter covers the following topics:

About Promotions and Deals

A promotion is a special discount that you create to market a specific product. For example, if you want to increase the sales of a specific model of computer, you might offer a promotional discount of 10% on that model for one month.

In Siebel ePricer, there are two types of promotions:

- **Corporate promotions.** Company-wide promotions of products which can be made available to any account, created by brand managers and corporate marketing managers. Corporate promotions are optional. They can be used as templates to make it easier to create account promotions.
- **Account promotions.** Account-specific promotions that apply to one account, many accounts, or all accounts. created by account managers. Account promotions are necessary to make a promotional discount available to customers.

After you create an account promotion, customers from those accounts can view the promotional deal in catalog views, while the promotion is available. When the promotion period expires or when the fund that was created to pay for the promotion becomes inactive, the promotion is no longer visible to customers.

NOTE: You cannot create promotions for customizable products.

NOTE: If you use promotions and deals with other types of price adjustments, the final price depends on the order in which Siebel ePricer applies the pricing adjustments. For more information, see "The Processing Order of Price Adjustments" on page 19.

The Process of Creating Promotions

To create a promotion, you go through the following process:

1. **"Creating a Fund" on page 80.** A brand manager, corporate marketing manager, or account manger must create a fund to cover the costs of the promotion.
2. **"Creating a Corporate Promotion" on page 81.** Optionally, a brand manager or corporate marketing manager may create a corporate promotion, which account managers can use as a template for their account promotions.
3. **"Creating an Account Promotion" on page 83.** Account managers must create account promotions to make promotions available to accounts.
4. **"Adding Products to the Account Promotion" on page 85.** Account managers must add products to the account promotions they create. The promotion will be visible when the accounts views these products in the catalog.
5. **"Creating Promoted Product Deals" on page 86.** Account managers must create promoted product deals for the account promotions they create, to specify what discounts will be available for the promoted products, and to specify which fund will be used to pay the cost of these discounts.
6. **"Creating Bundled Product Deals" on page 87.** Optionally, account managers may create bundled product deals, to provide promotional discounts to customers who purchase a specific combination of products.
7. **"Creating Payments Against Promoted Product Deals" on page 89.** Optionally, after promotions and deals have been set up, sales representatives can create payment records against funds, so they can reimburse accounts for running promotions.

You can configure your application using Siebel Tools so that deals funded by that fund are no longer be visible to customers when that fund is used up. For more information about Siebel Tools, see *Siebel Tools Reference*.

Creating a Fund

Corporate managers or account managers create funds to be used to cover the cost of promoted product deals.

Promotions are only visible if there is a valid fund associated to the promoted product. In addition, you can configure the application using Siebel Tools so that:

- Each time a customer buys a promoted product, a deal payment is created for the discounted amount, and that payment is deducted from the fund.
- Once the fund has been spent, the promotions are no longer available.

For more information about Siebel Tools, see *Siebel Tools Reference*.

To create funds

1. From the application-level menu, choose View > Site Map > Promotions Administration > Funds.
2. In the Funds list, click New.
3. In the new record and the More Info form, enter the information described in Table 12.

Table 12. Fields in the Funds List

Field	Description
Fund ID	System-generated ID number for the new fund.
Created By	System-generated field with the login name of the person who created the fund.
Start Date	Date when the fund becomes available. Entered automatically when you enter the Period field.
Description	Enter a description of the fund.
Name	Enter a name for the fund.
End Date	Date when the fund is no longer available. Entered automatically when you enter the Period field.
Amount	Enter the total amount of money to be provided by the fund.

Table 12. Fields in the Funds List

Field	Description
Active	There must be a check mark in this field to make the fund active. You can remove the check mark if you want to make the fund temporarily inactive.
Type	Select the fund type from the picklist. The available types are Fixed, Accrual, and Mixed. You can select Accrual or Mixed, only if your Siebel administrator has defined accrual rules using Siebel Business Process Designer. For more information about Siebel Business Process Designer, see *Siebel Business Process Designer Administration Guide*.
Period	Click the select button, and use the Pick Period dialog box to select the period when this fund is active. The Start Date and End Date fields of the Fund record are automatically filled in with the Start Date and End Date fields of the period record that you select.

Creating a Corporate Promotion

Corporate promotions are company-wide promotions of products. Corporate marketing managers and brand managers can create corporate promotions for products.

The corporate promotion is a template that account managers can use as the basis of account promotions

For example, a corporate marketing manager for an electronics company may create a corporate promotion for the launch of a new PC in North America during the first month that it is available. Then an account manager can use the information in this corporate promotion to create account promotions for his own accounts.

Corporate promotions provide information such as objective of the promotion and the period of the promotion, which account managers can use to create promotions for their accounts.

NOTE: Creating corporate promotions is optional. Account promotions do not need to be based on corporate promotions.

To create a corporate promotions

1. From the application-level menu, choose View > Site Map > Promotions Administration > Corporate Promotions.
2. In the Corporate Promotions list, click New.
3. In the new record and the More Info dialog box, enter the enter the information described in Table 13.

Table 13. Fields in the Corporate Promotions List

Field	Description
Name	Enter a name for the corporate promotion.
Promotion #	System-generated ID number for the new fund.
Period	Click the select button, and use the Pick Period dialog box to select the period when this promotion is active.
Advertising Description	Enter a description of the promotion to be used in advertising.
Category	You can use Siebel Tools to add logic for this field or remove it from the display.
Products	Click the select button, and use the Associated Products dialog box to choose the product or products that this promotion applies to.
Objective	Enter the objective of this promotion, the goal it is meant to achieve.
Packaging Description	Enter a description of the promotion to be used on product packaging.
Start Date	Date when the promotion becomes active. A default date is entered automatically when you enter the Period field.
End Date	Date when the promotion is no longer active. A default date is entered automatically when you enter the Period field.
Consumption Start Date	The default value is the Start Date. The value must be between the Start Date and the End Date.
Consumption End Date	The default value is the End Date. The value must be between the Start Date and the End Date.

Table 13. Fields in the Corporate Promotions List

Field	Description
Shipment Start Date	The default value is the Start Date. The value must be between the Start Date and the End Date.
Shipment End Date	The default value is the End Date. The value must be between the Start Date and the End Date.

Creating an Account Promotion

Account managers can create promotions for individual or multiple accounts. Creating these account promotions is necessary to make the promotion visible to customers at the accounts.

Account managers may use corporate promotions as templates for their account promotions.

After creating an account promotion, the account managers must add products to the promotion and create promoted product deals to cover the cost of the promotion.

NOTE: The Status field can be used to track the account promotion as it goes through the approval process. You can configure the Status field using Siebel Business Process Designer to make the approval workflow automatic.

To create an account promotion

1. From the application-level menu, choose View > Site Map > Promotions Administration > Account Promotions.
2. In the Account Promotions list, click New.
3. In the new record, click the select button in the Promotion field.

   The Promotions dialog box appears.

4 In the Promotions dialog box:

a If you want to base the new account promotion on an existing promotion, select the existing promotion and click OK.

b If you do not want to base the new account promotion on an existing promotion, click New to add a new record to the Promotions dialog box, enter information about the new account promotion in this record as described in Table 14, and click OK.

The new record in the Account Promotions list is filled in with information from the Promotions dialog box and with other defaults.

5 In the new record of the Account Promotions list and in the More Info dialog box, change the defaults as necessary, by entering the information described in Table 14.

Table 14. Fields in the Account Promotion List

Field	Description
Name	Enter a name for the account promotion.
Promotion	Optionally, select a promotion to be used as the basis or template for the account promotion.
Status	Select the status of the promotion. Typical statuses are New, Pending Approval, Approved 1, Approved 2.
Active	Check this box to make the promotion active.
All Accounts	Check this box to make the promotion available to all accounts.
Account	Select one or more accounts that the promotion will be available to.
Period	Select the time period during which the promotion is valid.
Type	Enter the type of promotion.
Revision	This system-generated field displays the number of times this promotion record has been revised.
Category	Select the product category of the promoted product.

Adding Products to the Account Promotion

When you create account promotions, you must add products to these promotions. Customers in these accounts will see these promotions when they browse catalogs for these products.

NOTE: Customizable products cannot be added to account promotions. All products defined in the Products Administration screen are available to be added to promotions except for products of the type Customizable.

To add a product to an account promotion

1. From the application-level menu, choose View > Site Map > Promotions Administration > Account Promotions.
2. In the Account Promotions list, select the promotion to which you want to add products.
3. Click the Promoted Products view tab.
4. In the Promoted Products list, click New.
5. In the new record in the Promoted Products list, click the select icon in the Product field.

   The Pick Product dialog box appears.
6. In the Pick Product dialog box, click Go.

   The Pick Product list appears.
7. In the Pick Product list, select a product and click OK.

   The product you selected appears in the Promoted Products list.

Creating Promoted Product Deals

Account managers must create promoted product deals to specify:

- What kind of price adjustments will be available for each promoted product, a fixed amount off the unit list price, a percentage discount, a percentage markup, a markup amount, or a price override
- Which fund will be used to pay the cost of the discounts.

They can also use promoted product deals to specify that the promotions are stackable or available for bundled products.

To create promote d product deals

1. From the application-level menu, choose View > Site Map > Promotions Administration > Account Promotions.
2. In the Account Promotions list, click the hyperlink in the Promotion field of the promotion you want to create promoted product deals for.
3. In the Promoted Products list, click the hyperlink in the Product field of the product you want to create the deal for.

   The Deals List appears.
4. In the Deals list, click New.
5. In the new record in the Deals list, enter the information described in Table 15.

Table 15. Selected Fields in the Deals List

Field	Description
Fund Name	Click the select icon, and use the Pick Funds dialog box to select the fund that pays for this promoted product deal.
Cost Type	Select the type of cost the deal covers. Options might be Bill Back or Off Invoice.
Promotion Message	Enter a message that describes the promotion. This message appears when customers view the promotion in product catalogs.
Status	Select the status of the deal. Options might be submitted, pending, and paid.

Table 15. Selected Fields in the Deals List

Field	Description
Minimum Quantity	Optionally, enter the minimum quantity a customer must purchase to receive the deal price.
Maximum Quantity	Optionally, enter the maximum quantity a customer can purchase to receive the deal price.
Incremental Quantity	Optionally, enter the increment in which customers must purchase products in order to receive the deal price.
Stackable	Enter a check mark in this field, to make the deal stackable. This means that customers can use more than one more promotion deals for the same promoted product.
Bundle	Enter a check mark in this field, if the deal is for a bundled product. For more information, see "Creating Bundled Product Deals" on page 87.
Price Adjustment Type	Select the type of price adjustment used for the discount. Typical options are Discount Amount and % Discount.
Price Adjustment Amount	Enter the amount of the price adjustment.

Creating Bundled Product Deals

Bundled product deals allow account managers to offer discounts for a specific combination of products.

When customers select bundled product deals in product catalogs, they receive the promoted product and bundled items at the promotional deal price.

For example, an account manager may create a bundled product deal in which customers can get a monitor for one-half price when they buy a PC. When a customer selects this bundled product deal, the promoted product (the PC) and the bundled item (the monitor) are automatically added to the shopping cart and the appropriate price adjustments are made.

NOTE: Bundled product promotions are not stackable.

NOTE: Bundled product promotions cannot include products of the type Customizable.

To create bundled product deals

1. Create a promoted product deal, as described in "Creating Promoted Product Deals" on page 86, and select the Bundle check box.
2. In the Bundled Products list, click New.

   A new record appears in the Bundle Products list.
3. In the Product field, click the select icon.

   The Pick Product dialog box appears.
4. Query for a product.

   Products appear in the pop-up list.

5 Select a product and click OK.

6 Fill in the other fields as described in Table 16.

Table 16. Fields in the Bundled Product List

Field	Description
Product	Select the product that will be bundled with another product as part of this deal.
Qty	Enter the quantity of the bundled product that will be added to the shopping cart.
Adjustment Type	Enter the adjustment type of the discount on the bundled product.
Adjustment Value	Enter the value of the price adjustment.

Creating Payments Against Promoted Product Deals

Sales representatives can create payment records against funds, so they can reimburse accounts for running promotions.

For example, an account may run a promotion of a product as a newspaper circular. To pay for this, a sales representative writes a check to the account and then creates the payment record against the fund associated to this promoted product deal.

NOTE: You can configure your application using Siebel Tools so that a payment record is created whenever a deal is executed on a quote. In addition, you can configure the application so deal payments are deducted from the fund the deal is associated to. Finally, you can configure the application so the fund is set to inactive when it becomes empty or at a certain level above empty. With these configurations, deals would no longer be visible to customers when the funds that pay for them are used up.

To create payments against promoted product deals

1. From the application-level menu, choose View > Site Map > Promotions Administration > Account Promotions.

   The Account Promotions list appears.

2. In the Account Promotions list, select the record that contains the promoted product deal against which you want to submit payments.
3. Click the Promoted Products view tab.
4. Click on a Product hyperlink in the Promoted Products list.
5. In the Bundled Products list, from the Show drop-down list, select Deal Payments.

   The Deal Payments list appears.

6. In the Deal Payments list, click New.
7. In the new record in the Deal Payments list, enter the information described in Table 17.

Table 17. Selected Fields in the Deal Payments List

Field	Description
Deal ID	System-generated number identifying the deal.
Status	The status of the deal, taken from the Deal record.
Amount	Required. Enter the amount of the deal payment.
Currency Code	The currency of the deal, taken from the Deal record.
Created	The date the deal was created.
Created Name	The username of the deal creator, taken from the Deal record.
Comments	Optional. Enter comments about this deal payment.

The Customer's Experience of Promotions

Promotions can be used by anyone who views products and their prices. Your employees, such as salespeople and call center employees, can view them when they view your product catalogs to deal with customers. Your customers can view them when you use Siebel eSales to make your product catalogs visible through the Web.

To administer promotions, it is useful to understand how customers view and use them.

Customers browse product catalogs to view and buy products. For some of these products, deals may be available. When customers view these products, they can click Browse Deals to view all the deals available for a product. They can take advantage of these deals by selecting a product deal and adding it to the shopping cart at the adjusted price.

To browse product catalogs and view deals

1. In a Siebel eSales Web site, the customer click the Catalog tab on the Home page.

   The product catalog appears.

2. The customer clicks on a product category hyperlink.

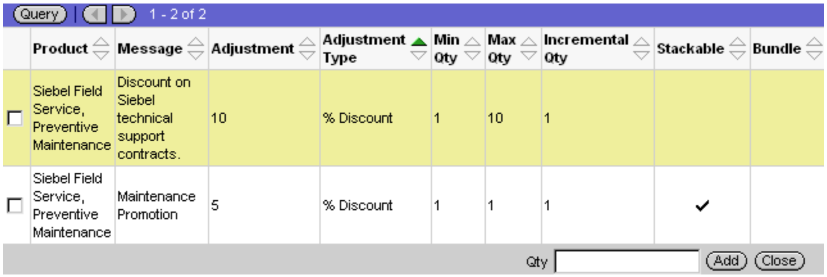
   All the products in the category appear.

3. The customer clicks on a product hyperlink.

   The Product form appears.

4 The customer clicks Browse Deals to view product promotions.

The Promotion Deals pop-up box appears, displaying all deals available for this product.



	Product	Message	Adjustment	Adjustment Type	Min Qty	Max Qty	Incremental Qty	Stackable	Bundle
□	Siebel Field Service, Preventive Maintenance	Discount on Siebel technical support contracts.	10	% Discount	1	10	1		
□	Siebel Field Service, Preventive Maintenance	Maintenance Promotion	5	% Discount	1	1	1	✓	

5 In the Promotion Deals pop-up box, the customer selects a deal by clicking the check box next to it and entering a quantity in the Qty field.

6 The customer clicks Add.

The product form appears, with the adjusted deal price for the product in the Your Price field.

7 The customer clicks Add To Cart.

The items selected appear in the Shopping Cart.

8 The customer clicks View Cart.

The Shopping Cart appears.

9 The customer clicks Save Cart to save the cart as a quote, or the customer clicks Check Out to place an order.

Creating Pricing Reports 7

This chapter describes available pricing reports. It gives instructions about how to run reports. It also gives information about creating and modifying reports.

This chapter covers the following topics:

About Pricing Reports

Siebel ePricer 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.

NOTE: The Pricing Engine Log file includes a variety of useful data that may be manually reformatted into reports. When you generate a Pricing Engine Log file, it is located in the Siebel server installation directory in a TEMP file. For information about this log file, see Chapter 16, “Testing and Validating Pricing Models.”

Descriptions of Available Reports

Predefined pricing reports that are available are described in this section.

Price Lists Report

Accessible from the Pricing Administration > Price List view, this report provides a list of all price lists in the database. Since this is a summary report, it contains Price List header data information, including the following information for each price list:

- Price list name
- Price list description
- Base currency
- Terms
- Shipping methods
- Effective dates

Price List Based Price Book Report

Accessible from the Pricing Administration > Price List view, this report shows line item (product-specific) pricing information for the selected price list.

For each line item in the selected price list, this report includes the following:

- Price list line item data
- Attribute-based pricing data
- Catalog data (product line, part number, vendor number)
- Cost data

- Margin %
- Minimum and maximum sales price

NOTE: For customizable products, only the price list line item data is included. Listing all the possible combinations of component product is not practical.

NOTE: To view pricing associated with bundled products and aggregate type pricing factors, refer to the appropriate pricing factor definition data under the Pricing Model Manager view.

Pricing Models - All Report

Accessible from the Pricing Administration > Pricing Model Manager view, this report provides a list of pricing factor information for the selected Pricing Model.

The high-level data in this report includes the pricing model name, its creation date, and its effective start date.

For each pricing factor in the selected model, this report presents pricing factor information that includes the following:

- The condition that determines whether the algorithm gets executed
- The algorithm used to adjust the target price
- The reason that is displayed to the runtime user if this pricing factor is applied
- Any comments regarding this factor that have been entered by Pricing Administrators

For more information about pricing models, see Chapter 10, “Creating Pricing Models and Pricing Factors.”

Quote Reports

The following reports also provide information useful to pricing administrators:

- **Quote List Report.** Accessible from the Pricing Administration > Quote Pricing Model Validation view, this report contains a listing of all the quotes.
- **Current Quote Report.** Accessible from the Pricing Administration > Quote Pricing Model Validation view, this report contains formatted information about the quote line item currently selected. You must select a line item from the quote before running this report.
- **Summary Quote Report.** Accessible from the Pricing Administration > Quote Pricing Model Validation view, this report contains summary information for the quote, such as taxes and terms. It also includes fundamental line item information.
- **Package Quote Report.** Accessible from the Pricing Administration > Quote Pricing Model Validation view, this report contains package information and summary information for the quote. It also includes fundamental line item information. Information about items that are not in packages is classified as Other.
- **Account Specific Price Book.** Accessible from the Quotes view, this report lists attribute-based pricing information for the selected Quote line items by showing which attributes have been selected, including the attribute name and value, along with the list price and net price of the product. To generate this report, you have to create a quote that contains all the products to be included in the price book. You must then update the account name in the report, and click Reprice All prior to generating the report.

Running Reports

You can run all the reports described in this chapter in the same way.

To run the Price Lists report

1. Navigate to the screen and view that the report is accessible from, as specified in the description of the report.
2. From the application level menu, choose View > Reports.
3. In the Reports dialog box, select the report name from the drop-down menu, and click Run Now.

   The selected report appears in the Siebel Report Viewer.
4. You can print a copy of the report by clicking the printer icon.

About Creating and Modifying Reports

You are not limited to working with these standard, predefined reports. The standard Siebel ePricer reports are part of a large set of Siebel application reports. You can modify these reports or add new reports in two locations:

- **Siebel Tools.** Used to define the of the data exported from the Siebel application to the Actuate report, and to attach reports to the Reports menus of specific views.
- **Actuate e.Report Designer Professional.** Used to define report behavior, appearance, and data acquisition.

For information about defining and working with reports, see *Siebel Reports Administration Guide*.

Setting Up Component-Based Pricing 8

This chapter describes pricing for component-based products, one of the two types of customizable products used by Siebel ePricer. It gives instructions about how to view a product’s components and how to set up component-based pricing. It also covers the processing order of price adjustments for component-based products.

This chapter covers the following topics:

Component-Based Versus Attribute-Based Pricing

Siebel ePricer allows you to work with two types of customizable products:

- **Component-Based Products.** Customers can customize products by choosing components. For example, when a customer buys a computer, the customer can customize it by choosing to add peripherals such as a CD drive.
- **Attribute-Based Products.** Customers can customize products by choosing attributes. For example, when a customer buys a tee-shirt, the customer can customize it by choosing its color.

In most cases, it is clear that a customizable product is one of these two, but in some cases, the product administrator must decide whether to let users select options as product components or as attributes.

For example, if a PC monitor comes in three sizes, 14-inch, 17-inch, and 21-inch, then those monitors could be offered as:

- Three separate products with three different prices, which are used as three component products within a customizable PC product.
- One monitor product with a "Screen Size" attribute that includes three values (14, 17, and 21), which is used as one component product within a customizable PC Product.

You should decide how to set up customizable products, in a way that will be as the product hierarchy and its circumstances change.

NOTE: In this example, you can see that if you are working with a component-based product, you may need to set up attribute pricing for the component products. A component of a component-based product is also a separate product in the price list, so you can define attribute pricing for it in the same way as for any other product, as described in Chapter 9, "Setting Up Attribute-Based Pricing." If components have attributes, Siebel ePricer first applies the component-based item pricing, then applies any attribute-based price adjustments for the components.

About Component-Based Pricing

Customizable products can contain components that are themselves products in the product table.

For example, you sell computer monitors, CD-ROMs, and disk drives. You also sell several models of desktop computer that include these products as components.

This chapter explains how to work with this sort of component-based product:

- **Viewing a product in the Pricing Designer.** You can use the Pricing Designer to view a product's components. It is also used for the other functionality described in this chapter.

- **Setting 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.

- **Creating 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.

- **Pricing models.** Pricing models for component-based products are created in the same way as pricing models for other products, but there are some added constraints, which are covered in this chapter.

If you create several types of price adjustments for component-based products, the order in which price adjustments are processed can affect the final price. In general, price adjustments for each component-based product are processed before other price adjustments, because Siebel ePricer must resolve the initial price of each line item in an order before applying further price adjustments. For more information, see "The Processing Order of Price Adjustments" on page 19.

NOTE: This chapter assumes that you have general background knowledge about component-based customizable products. For more information about customizable products, see *Product Administration Guide*.

Viewing a Product’s Components in the Pricing Designer

You can use the Pricing Designer to display the relationships and components of a customizable product. You can expand or collapse the relationship hierarchy to see the components of each relationship.

NOTE: If the List Price column for a component is blank, this means the component has not been added to a price list. Enter a price in the Price List field. This automatically adds the component as a price list line item to the current price list.

To display a product’s components in the pricing designer

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

   The Price List view appears.

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 customizable product’s name.

   The Pricing Designer view appears.

5. To expand a relationship and view its components, click the relationship name in the Outline Number column.

   NOTE: You can also expand customizable products that are components of the current customizable product. The indentation level of each item displays to the right of its icon. The first level is 1, the second is 1.1 and so on.

About Setting Up Component-Based Pricing

There are two types of component-based pricing for customizable products:

- **Base pricing.** Used when customizable products share a base configuration. Customers who purchase the base product can customize it by selecting options. Cars, aircraft, and many types of financial services are examples.
- **Summation or roll-up pricing.** Used when customizable products do not share a common base configuration. Customers can choose all of the components. Desktop computers and simple product bundles are examples.

Setting Up Base Pricing

Use this pricing method when the user purchases some base product and can customize it by adding options. The total price is the sum of the base price plus the adjusted price of the components.

For example, a car has a base price of $25,000. The user can also purchase additional options, such as air conditioning, sun roof, and so on.

To set up base pricing you would do the following:

- Set the list price of the car to $25,000.
- 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, the price increases.

For example, a car has a base price of $25,000 and has the options shown in Table 18.

Table 18. Car Options

Item	List Price
2.1 Chrome Wheels (Default)	$2000
2.2 Gold Wheels	$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, 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 0.
- Create a price override adjustments for related nondefault options that sets their price to the difference between the default option and the nondefault option.

In the example shown in Table 18 on page 105, you can 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 0 for the Chrome wheels.
- Create a price override to $1000 for the Gold wheels. This is the difference in price between the gold and chrome wheels.

To set up base pricing product

1. Display the components of the customizable product in the Pricing Designer, as described in “Viewing a Product’s Components in the Pricing Designer” on page 104.
2. Define any needed relationship-based pricing adjustments as described in “Creating a Relationship-Based Price Adjustment” on page 110.
3. In Price List Item, enter the base price in the List Price field.
4. Enter Prices for all the components.
5. If you are using the alternative method of setting up base pricing, enter the necessary price overrides.

Setting Up Summation or Roll-Up Pricing

Use this pricing method when all of the product’s components are configurable. The product has a default configuration, and the user can select nondefault components.

When you use the Pricing Designer, you must set a list price for the base product of a customizable product as well as defining pricing adjustments on components. The price that displays to the user is the sum of the list price plus the adjusted price of the customizable product’s default components.

This is true for Quote, Agreement, Order, and the selection pages in a configuration session. In an eSales catalog, the user sees only the list price of the base product.

When you are using summation or roll-up pricing, there is no base product, so you set the list price of the base product to 0. Thus, 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 roll-up 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 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, 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 shown in Table 19.

Table 19. Desktop Computer Components

Item	List Price	Adjustment Type	Adjusted Price
1.0 Chassis (default)	$1200	Price Override	$1000
2.0 Monitor			
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, 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 roll-up pricing for a customizable product

1. Display the components of the customizable product in the Pricing Designer, as described in "Viewing a Product's Components in the Pricing Designer" on page 104.

2. Define any needed relationship-based pricing adjustments as described in "Creating a Relationship-Based Price Adjustment" on page 110.

3. Do not enter a price in the List Price field in the Price List Item applet. Do not enter a price for the customizable product (also called the root product) 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 ePricer 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.

Siebel ePricer 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, Siebel ePricer adds the list price and the price of all the default components. This calculation does not take account of 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 top applet. Then you can 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.

To view the reference price

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.

   The Price List view appears.

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 a customizable product’s name.
5. 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 is displayed in the Reference Price field.

Creating a Relationship-Based Price Adjustment

Relationship-based price adjustments allow you to adjust the list price of a product when it is a component of a customizable product.

For example, you may sell a model of computer monitor for $300. If the customer who is buying one of your desktop computers chooses this monitor model as an options, you can reduce its price 10% from this list price.

The types of adjustments you can make are described in Table 20.

Table 20. Adjustment Types for Relationship-Based Pricing

Adjustment Type	Explanation
Discount Amount	Reduces price by: amount entered in the Adjustment Value field.
% Discount	Reduces price by: (amount entered in the Adjustment Value field/100) multiplied by the list price.
Markup Amount	Increases price by: amount entered in the Adjustment Value field.
% Markup	Increases price by: (amount entered in the Adjustment Value field/ 100) multiplied by the list price.
Price Override	Overrides the original list price with: amount entered in the Adjustment Value field.

To define a relationship-based pricing adjustment

1. Display the components of the customizable product in the Pricing Designer, as described in "Viewing a Product's Components in the Pricing Designer" on page 104.
2. In the Adjustment Type field of the component, select the adjustment type.
3. In the Adjustment Value field of the component, enter the adjustment value.
4. Collapse the relationship, then expand it, in order to refresh the display.
5. Verify that the component's Item Price reflects the pricing adjustment.

NOTE: You can modify the relationship-based adjustment by editing the amounts in the Adjustment Type and Adjustment Value field, and you can delete the relationship-based adjustment by deleting the amounts in these fields.

About Pricing Models for Component-Based Products

Pricing models are advanced features of Siebel ePricer, which allow you to define your own price adjustments.

This section provides background about the special features of pricing models for component-based products. You refer to this section when you follow the instructions for creating pricing models in Chapter 10, “Creating Pricing Models and Pricing Factors” and in the chapters that follow it.

If a price list includes line items for component-based products, then each of those price list line items may be linked to a pricing model.

A pricing model for a component-based product has the same fundamental structure as other pricing model. You use the same procedures and work with the same Pricing Model Manager forms and functions to create a pricing model for component-based products and for other products.

The key difference is that product components are the line items in a pricing model for a component-based product, while products are the line items in other pricing models. Customizable product pricing models work with the prices of components of a product, whereas other list models work with the prices of products in a price list.

Because the component-based product is a single line item, pricing models for component-based products are processed before other pricing models. All of the price adjustments for a customizable product—including the pricing model price adjustments—must be completed before that product’s price can be considered as a list line item, so Siebel ePricer can decide whether other price adjustments apply to it.

With a few exceptions, you can use a customizable product pricing model to adjust the prices of the line items in a customizable product in the same ways that you use a price list pricing model to adjust the prices of line items in a price list.

Constraints on Pricing Models for Component-Based Products

Pricing models for component-based products have the following constraints:

- All factors in a pricing model for a component-based product must refer to some element in the component-based product, and must not reference any product or element outside of that product. The flexibility of the Pricing Model Manager user interface allows you to specify an inappropriate business component when creating a pricing factor for a pricing model for a product with components, which will result in errors.
- Aggregate type pricing factors may not be used in a pricing model for a component-based product. If they are used, the model will base logical tests on the prices of components and of noncomponent items, and to use the prices on these items to determine the price of your component-based product. The result will be incorrect and unpredictable pricing.
- The pricing model for the component-based product must have a start date and end date that are within the start date and end date of any price list that includes the customizable product.
- If a single pricing model includes elements of a component-based product and other elements of a price list, there will be errors because of double-counting problems and crossover problems. To avoid errors:
  - If you are creating a pricing model for a price list, do not reference the customizable product (CP) business components shown in Table 21.
  - If you are creating a pricing model for a component-based product, reference only the customizable product (CP) business components shown in Table 21.

Table 21. Business Components Used in Component-Based Pricing Factors

Business Component Name	Single	Bundle Type	Aggregate Type
Customizable Product Header	4		
Customizable Product	4	4	
Customizable Product XA	4		

Setting Up Attribute-Based Pricing 9

This chapter describes pricing for attribute-based products, one of the two types of customizable products used by Siebel ePricer. It gives instructions about how to design and create attribute-based pricing tables and about how to use the adjustment item generator.

This chapter covers the following topics:

- "About Attribute-Based Pricing" on page 114
- "About Attribute-Based Pricing Tables" on page 115
- "Decisions About Attribute-Based Pricing Table Design" on page 116
- "The Process of Creating Attribute Pricing Tables" on page 124
- "The Process of Creating Attribute Pricing Tables with the Adjustment Item Generator" on page 132
- "Troubleshooting Attribute Pricing Adjustments" on page 138
- "Configuring Automatic Navigation for the Adjustment Item Generator" on page 139

About Attribute-Based 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 would see 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 would see 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 would apply to different target prices.

When a user looks at any customizable product in a product catalog, the user sees the reference price in the Starting At Price field. This is the price of the product with its default options. For more information about the reference price, see "Viewing and Changing the Reference Price" on page 109.

NOTE: If you use attribute-based pricing with other types of price adjustments, the final price depends on the order in which Siebel ePricer applies the pricing adjustments. For more information, see "The Processing Order of Price Adjustments" on page 19.

About Attribute-Based Pricing Tables

Pricing Administrators define attribute-based pricing by setting up pricing adjustment tables.

Each row in the table specifies a product attribute (or a combination of attributes) and the price adjustment for that attribute. These rows are called *adjustment items*.

To apply these adjustments to a product, you associate the attribute pricing table with the product in a price list line item.

An attribute pricing table is based on data in one class, so an attribute pricing table can only be applied to a product or products within that class. Depending on your approach to pricing, you may be able to use the same attribute pricing table for all products within a class, or you may have to create attribute pricing tables used for individual products.

CAUTION: Attribute-based pricing can only use Attribute values that have been defined as discrete elements in a List of Values (LOV). It cannot use attributes that have been defined with a Range of Values.

NOTE: This chapter assumes that you have general background knowledge about attribute-based customizable products, classes and attribute values. For more information about customizable products, see *Product Administration Guide*.

Decisions About Attribute-Based Pricing Table Design

To plan an attribute-based pricing table, begin by identifying the products and product attributes associated with the class for which you are developing the attribute pricing table. You may want to create a spreadsheet showing the products associated with each class, and the attribute pricing tables that use the attributes of each class.

Determine which attributes will be used for pricing. Some attributes 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. Then, you would not use the color attribute in the pricing table.

NOTE: When you use the class and attribute structure for pricing, the class structure and its associated attributes and lists of values (LOVs) should not be changed without coordination between the Siebel Product Administrators and ePricer Administrators.

After determining which attributes affect pricing, you must decide whether you need a single-attribute or multiple-attribute pricing adjustment table.

Single-attribute pricing-adjustment tables are relatively simple, and if this is what you need, you can go on to create it.

Multiple-attribute pricing-adjustment tables are more complex, and if this is what you need, you will have to make more decisions about their design.

Single Attribute and Multiple Attribute Pricing Tables

When you create an attribute pricing table, you use the Type field to specify that it is a single-attribute or a multiple-attribute table:

- **Single Attribute.** The user selects a single attribute value to trigger a price adjustment. Use this type if the product has only one attribute that affects pricing, such as its size.

- **Multiple Attribute**. The user selects a unique combination of attribute values to trigger a price adjustment. Each row in the adjustment items list must specify the multiple attributes needed to trigger the price adjustment. Use this type if the product has several attributes that affect pricing, such as its size and its color.

If you are using a single-attribute table, you do not have to make any more decisions about table design, and you can go directly to "The Process of Creating Attribute Pricing Tables" on page 124.

If you are using a multiple-attribute table, before creating the table, read "Decisions About Design of Multiple Attribute Pricing Tables" on page 117.

Decisions About Design of Multiple Attribute Pricing Tables

Before you begin creating a multiple-attribute pricing table, you should make decisions about the following features of the table:

- **Order of Attributes.** You must plan the order in which you list attributes, because the order affects the adjusted price.
- **Display Error When Invalid Combination.** You can select this option to prevent the user from choosing an invalid combination of attributes.
- **Allow Any.** This option allows you to insert null value wildcards into the pricing table, so one record represents many value of an attribute.
- **Adjustment Item Generator.** This feature allows you to automatically populate an attribute pricing table with price adjustment items representing all possible combinations of attributes.
- **Zero-Effect Adjustment Items.** You may want to create adjustment items that have no effect on pricing to make the table easier to maintain.

Order of Attributes in Multiple-Attribute Pricing Tables

If you are creating a multiple-attribute pricing table, the order in which you add attributes to the Attributes list determines the order of attributes in each pricing adjustment item, and:

- The order of attributes affects the pricing calculation
- If certain product attributes must override other attributes for pricing purposes, the order of attributes determines precedence

- The order of attributes affects processing speed

In single-attribute pricing tables, the order of attributes in the Attributes list is not important. Each price adjustment item includes only one attribute, so the price adjustment for that attribute is based on one calculation applied to the list price or promotional price.

In multiple-attribute pricing tables, each price-adjustment item includes multiple attributes, so the price adjustment is based on several calculations applied to the list price promotional price or list price. The order of attributes determines the order of the calculation and affects the result of the calculation.

For example, a product has a list price of $100, and there is a 50% markup for choosing the large size, and a $10 markup for choosing the color gold.

- If you calculate the size markup first, the large gold product will cost $160 (100 + 50% = 150, and 150 + 10 = 160).
- If you calculate the color markup first, the large gold product will cost $165 (100 + 10 = 110, and 110 + 50% = 165).

In addition, the selection order also determines the relative importance of attributes in case of ties. If two or more adjustment item rows in a table are both valid, the row with more explicitly defined attribute values (not blanks) in the leftmost columns takes precedence. For example, if there are four fields per record, and the first two fields of both records match what has been selected, the engine will return the record that matches first going from left to right.

Finally, the first attribute in each adjustment item row has the greatest impact on processing speed. If it does not make a difference in final price, in order to speed processing, follow these general rules:

- Select the attributes with the fewest values first.
- Do not select attributes first if they will be flagged as Allow Any.
- Do not select attributes first if you will allow the use of a wildcard in the Attribute field.

You must plan the order of attributes before you create a price adjustment table. The order in which you add attributes to the Attribute list determines their order in Price Adjustment Items, and it cannot be changed.

The Display Error When Invalid Combination Option

To prevent users from selecting combinations of attributes combinations that are not valid, you can select the Display Error When Invalid Combination check box in the Pricing Table Header record.

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 chose both large and green.

If you select this option, you must create an adjustment item record representing every valid combination of attributes, even if the combination has zero effect on pricing. The application will display an error message if the user selects an combination of attributes that is not represented by an adjustment item.

For example, even if gold is the only color that requires a price markup, you will have to add attribute item records for all the colors that the user can select. If there are 100 combinations of other attributes, and 10 possible colors, you will have to create 1000 attribute item records, even if only one color affects pricing.

To decide whether to use this option, you should weigh its benefits against its costs. There is a benefit to preventing users from choosing invalid combinations of attributes. But the added administrative cost is high because there are many combinations of attributes that do not effect pricing.

Most important, if you leave this check box blank, the user can proceed after choosing an invalid combination of attributes, and will not receive any price adjustment for the product. You must have some business plan to deal with these invalid orders.

If you select the Display Error When Invalid Combination option, you must create a pricing adjustment item representing every valid combination of attribute values, and you must not create a pricing adjustment item for any invalid combination of attribute values. To create the correct pricing adjustment items, you can use one of the two following features of Siebel ePricer:

- **Allow Any option.** Allows you to create adjustment items with null value wildcards that accommodate many attribute value combinations from the runtime interface, so you can use fewer adjustment items to represent all valid combinations. "The Allow Any Option and Null Value Wildcards" on page 120.

- **Adjustment Item Generator.** Allows you to automatically generate a full set of adjustment item records for all possible combinations of values of attributes that affect pricing. You can delete the adjustment items for invalid combinations of attributes. For more information, see "About the Adjustment Item Generator" on page 122.

These two features can also be useful if you do not select the Display Error When Invalid Combination option.

NOTE: If you select the Display Error When Invalid Combination option, you must be sure that the default attributes for the product are a valid attribute pricing combination. If it is not a valid combination, your Siebel application will not add the product when users click the Add Item button to add this product to a quote. It will seem to users that the Add Item button is not working.

The Allow Any Option and Null Value Wildcards

When you create a Multiple Attributes table, selecting the Allow Any option allows you to enter a blank instead of entering every different domain value for a given attribute. This blank is a *null value wildcard* that represents all possible values in the given attribute's domain.

The Allow Any option is useful when you use the Display Error When Invalid Combination option, so you must have an adjustment item to represent every valid combination of attributes. It may allow you to use a wildcard instead of entering each separate value for an attribute.

For example, suppose you sell tee-shirts that come in the colors White, Gray, and Red and come in the sizes Small, Medium, Large, and Extra-Large, with the pricing shown in the following table:

Color	Size	Adjustment Type	Adjustment Value
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 table shown above, you can represent this with the following short table:

Color	Size	Adjustment Type	Adjustment Value
Red		Markup Amount	$2.00
Red	Extra Large	Markup Amount	$5.00

In the short 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 you must be careful to avoid the problems that it can create.

- You may mistakenly enter wildcards that allow the user to choose invalid combinations of products.
- You may also mistakenly create several pricing adjustment items that represent the same combination of attributes. Make sure that every Attribute Value combination in an attribute pricing table is unique.

If you do create several pricing adjustment items that represent the same combination of attributes, Siebel ePricer uses the adjustment item with that has the most exact matches (not blanks) in the leftmost position in the attribute pricing table. If one adjustment item had a blank for the third attribute and another had a blank in the fourth attribute, the latter would win. You should put the most important attributes in the leftmost position.

Avoid using null value wildcards in the first (leftmost) columns. This increases the processing task for ePricer and causes the processing to take longer. It is also contrary to the planning principle that the required, higher priority attributes should appear in the leftmost columns.

NOTE: If you use the Allow Any option and null value wildcards, it is especially important to test your results after creating the attribute pricing table.

About the Adjustment Item Generator

The Adjustment Item Generator allows you to automatically generate adjustment item records for all possible combinations of attribute values that affect pricing.

After adjustment item records are generated, you must enter the adjustment type and the adjustment value in each. If any combinations of attributes are invalid, you must delete those adjustment items.

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 make the decision, consider that:

- If you use the adjustment item generator, you must do the extra administrative work, entering adjustment type 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, you may have a difficult conceptual task of analyzing the combinations of attributes to decide where you can should use wildcards.
- If you chose the Display Error When Invalid Combination 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 example above). If you did, there would be no error message when the user chose extra large and gray.

If you decide not to use the adjustment item generator, go to "The Process of Creating Attribute Pricing Tables" on page 124

If you decide to use the adjustment item generator, go to "The Process of Creating Attribute Pricing Tables with the Adjustment Item Generator" on page 132.

Zero-Effect Adjustment Items

If you choose the Display Error When Invalid Combination Option, you must create adjustment items representing every valid combination of attributes, even if they have no effect on pricing, as noted in "The Display Error When Invalid Combination Option" on page 119.

Even if you do not choose this option, you may want to create zero-effect adjustment items for the following reasons:

- To make the table easier to maintain. If a pricing attribute is not represented in an attribute pricing table, it can be difficult or impossible to add it later and integrate it with the adjustment items already in the table.
- To create a comprehensive, well structured table. If an attribute pricing table has a structured place for everything, it can be easier to work with, even if it has items with zero effect.

The Process of Creating Attribute Pricing Tables

To define an attribute pricing table, you go through the following process:

1. **"Creating the Attribute Pricing Table Header" on page 125.** The header includes general information about the attribute pricing table.

2. **"Selecting Attributes That Will Trigger Price Adjustments" on page 127.** When you created the header, you specified the class that the attribute pricing table applies to. Next, you must specify the attributes in the class that will trigger pricing adjustments.

3. **"Adding Adjustment Items" on page 129.** The adjustment items specify what price adjustment the attributes trigger.

4. **"Attaching the Attribute Pricing Table to a Price List Line Item" on page 131.** After creating the attribute pricing table, you must apply it to a product in a price list.

After creating an attribute pricing table, review and test it to make sure that it adjusts prices as expected. For example, make sure that there are not multiple rows that apply different price adjustments based on the same product attribute.

NOTE: This process describes how use the Adjustment Items form to create individual adjustment items. If you are creating a multiple-attributes table, you can also use the Adjustment Item Generator form to produce sets of adjustment items. To do that, see "The Process of Creating Attribute Pricing Tables with the Adjustment Item Generator" on page 132.

Creating the Attribute Pricing Table Header

The attribute pricing table header data includes data that specifies your strategic approach to attribute pricing for a product or class of products. Before you create the table header, you should do the planning described in the section "Decisions About Attribute-Based Pricing Table Design" on page 116.

CAUTION: If you delete an attribute pricing table header, all references to that attribute pricing table in any price list line item become invalid, and all price adjustments associated with that attribute pricing table are lost. Do not delete a table unless you want this result. Create a complete backup of your Siebel ePricer system data before you delete a table.

To create an attribute pricing table header

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, click New.

   The New Attribute Pricing list appears.
3. Enter information in the new record and in More Info form, as described in Table 22.

Table 22. Attribute Pricing List Fields

Field	Entry
Name	Required. Enter a name for the attribute pricing list; 60 characters maximum.
Class	Required. Click the select button, and select a class. The attributes in this class can be used in this attribute-based pricing table.
Description	Optional. Enter a longer description of this attribute pricing table. Describe the characteristics that distinguish this table from any other tables that apply to products with the same Class.

Table 22. Attribute Pricing List Fields

Field	Entry
Type	Required. Select Single Attributes to create pricing adjustments based on one attribute value. Select Multiple Attributes to create pricing adjustments based on combinations of attribute values. For more information, see "Single Attribute and Multiple Attribute Pricing Tables" on page 116.
Display Error when Invalid Combination	Check this box to send an error message if the user selects a combination of attributes that is not represented in the attribute adjustment table. If you leave this check box blank, the user can proceed after choosing an invalid combination of attributes, and will not receive any price adjustment for the product. For more information, see "The Display Error When Invalid Combination Option" on page 119.
Default Starting Adjustment Type	Select the Adjustment Type value to be assigned by default when you create or generate adjustment items. You can overwrite this default when you work with each adjustment item. Price Adjustment Type values include Discount Amount, % Discount, Markup Amount, % Markup. These are described in detail in Table 25 on page 135.
Currency Code	Select the currency code of the price list these adjustments will apply to. The currency code in the attribute pricing table header must be the same as the currency code for any associated price list. All attribute-based price adjustments are calculated in the currency used for the list price of the product.
Start Date	Enter the date when this attribute pricing table becomes effective. This date must be coordinated with the Effective From date of any price list with a line item to which this attribute table will be attached.
End Date	Optional. Enter the date when this attribute pricing table stops being effective. This date must be coordinated with the Effective To date of any price list with a line item to which this attribute table will be attached.

Selecting Attributes That Will Trigger Price Adjustments

After creating the attribute pricing table header, you must use the Attributes list to select all the product attributes that this table will use as pricing attributes to trigger price adjustments. The attributes that you select now can be used when you create adjustment items.

The attributes that you can select are limited to the attributes in the Class that you specified in the attribute pricing table header data. To use an attribute in a Attribute Pricing table, it must be associated with the Class that you selected when you created the header for this attribute pricing table.

Whether you are defining a table based on single or multiple attributes, the steps in this task remain the same.

However, if you are creating a multiple-attribute pricing table, the order in which you select attributes is crucial. For more information, see "Order of Attributes in Multiple-Attribute Pricing Tables" on page 117.

NOTE: If you are creating a Multiple Attributes type attribute pricing table, do not change the selection of pricing attributes after you start to create adjustment item rows for the table.

To select attributes that will trigger price adjustments

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, select the attribute pricing table for which you want to specify attributes.
3. Click the Attributes view tab.

4 In the Attributes list, click New.

5 Enter information in the new record, as described in Table 23.

Table 23. Attribute Fields

Field	Entry
Sequence	The sequence number is entered automatically based on the order in which you select attributes. The attribute that you select first is assigned sequence number 1 and will appear first (in the leftmost position) in each adjustment item row, and so on. For more information about what order to select attributes in, see "Order of Attributes in Multiple-Attribute Pricing Tables" on page 117.
Name	Select the attribute name from the drop-down list. This list includes all attributes associated with the class specified in the attribute pricing table header.
Allow Any:	If you select the Allow Any box for an Attribute, then you will be able to use a null value wildcard for this attribute when you create adjustment items. This option is used in setting up multiple attributes tables. For more information, see "The Allow Any Option and Null Value Wildcards" on page 120.

Adding Adjustment Items

This section covers the Adjustment Items list, which allows you to create adjustment items one at a time. To use the Adjustment Item Generator, see "The Process of Creating Attribute Pricing Tables with the Adjustment Item Generator" on page 132.

If you are creating an attribute pricing table of the Single Attributes type, add an adjustment item for each attribute that can affect pricing.

If you are creating an attribute pricing table of the Multiple Attributes type, which attribute items you add depends on other options you have chosen. To decide which attribute items to add, see "Decisions About Design of Multiple Attribute Pricing Tables" on page 117.

NOTE: For a multiple-attributes pricing table, no two adjustment items can have the same set of attribute values. An error message appears if you try to create an adjustment item with the same attribute values as an existing adjustment item.

To create adjustment items using the Adjustment Items list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, select the attribute pricing table for which you want to create attribute pricing adjustment items.
3. Click the Adjustment Items view tab.

   The Adjustment Items list appears, and it includes all the attributes that you added to the Attributes list.
4. Select the pricing attribute with which you want to work.

5 In the Items list (which is below the Adjustment Items list), click New.

6 Enter information in the new Items record, as described in Table 24.

Table 24. Items Fields

Field	Entry
Adjustment Type	Select the type of adjustment to be applied. The options are Discount Amount, % Discount, Markup Amount, and % Markup.
Adjustment Value	Enter the value of the adjustment. If you selected Discount Amount or Markup Amount in the Adjustment Type field, the price will be decreased or increased by the value you enter here. If you selected % Discount or % Markup in the Adjustment Type field, the value you enter here will be treated as a percentage: the price will be decreased or increased by the value divided by one hundred multiplied by the price.
Description	Optional. Enter a description of this adjustment item.

NOTE: If you checked the Allow Any option for an attribute when you added it to the Attributes list, you have the option of leaving the Attribute Item record blank, creating a null value wildcard. Do not do this unless you understand the consequences. For more information about the null value wildcard, see "The Allow Any Option and Null Value Wildcards" on page 120.

Attaching the Attribute Pricing Table to a Price List Line Item

After creating the attribute pricing table, you must apply it to the product that it was designed for, by attaching it to the price list line item that represents the product.

You can use the same attribute price list for the product in all your price lists, by assigning it to the line item that represents that product in each price list. Or you can create different attribute adjustment tables for different price lists, in order to give different attribute price adjustments to different customers.

To attach an attribute pricing table to a price list line item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price List.
2. In the Price Lists list, select the price list the adjustment will apply to.
3. Click the Price List Line Items view tab.
4. In the Price List Line Items list, in the record of the product that the attribute price adjustment applies to, in the Attribute Pricing Field, click the select icon.

   The Pick Attribute Pricing dialog box appears.
5. In the Pick Attribute Pricing dialog box, select the attribute pricing table, and click OK.

The Process of Creating Attribute Pricing Tables with the Adjustment Item Generator

If you are working with a multiple-attribute pricing table, you can use the Adjustment Item Generator rather than adding each adjustment item individually to the Adjustment Item list. For information about whether you should use the Adjustment Item Generator, see "Decisions About Design of Multiple Attribute Pricing Tables" on page 117.

To define an attribute pricing table using the adjustment item generator, go through the following process:

1. **"Creating the Attribute Pricing Table Header" on page 125.** The header includes general information about the attribute pricing table. (This step is the same, whether or not you are using the Adjustment Item Generator.)
2. **"Selecting Attributes That Will Trigger Price Adjustments" on page 127.** When you created the header, you specified the class that the attribute pricing table applies to. Next, you must specify the attributes in the class that will trigger pricing adjustments. (This step is the same, whether or not you are using the Adjustment Item Generator.)
3. **"Generating Adjustment Items" on page 133.** The Adjustment Items Generator automatically creates an adjustment item record for every possible combination of the attributes that affect pricing.
4. **"Completing the Generated Adjustment Items" on page 135.** In each valid adjustment item that was generated, you must enter actual price adjustment type and price adjustment value.
5. **"Deleting Invalid Adjustment Items" on page 137.** If any of the generated adjustment items contains an invalid combination of items, you must delete it.
6. **"Attaching the Attribute Pricing Table to a Price List Line Item" on page 131.** After creating the attribute pricing table, you must apply it to a product in a price list. (This step is the same, whether or not you are using the Adjustment Item Generator.)

After creating an attribute pricing table, review and test it to make sure that it adjusts prices as expected. For example, make sure that there are not multiple rows that apply different price adjustments based on the same product attribute.

NOTE: The first two steps in this process are the same as the first two steps in "The Process of Creating Attribute Pricing Tables" on page 124. Follow the instructions for the first two tasks there before continuing with the remaining tasks.

Generating Adjustment Items

If you are working with a Multiple Attributes type table, you can use the Adjustment Item Generator function to populate the entire table with a well-ordered set of records that includes all possible combinations of the attributes that affect pricing.

When you use the Adjustment Item Generator, you work with two lists:

- **Adjustment Item Generator list.** Displays the attributes in the order in which you added them to the Attributes form. The displayed sequence, from top to bottom, indicates the sequence in which the attributes will be listed, from left to right, in each Multiple Attributes type adjustment item.
- **The Item Generator list.** Displays all the values in the domain of the attribute selected in the Adjustment Item Generator list (every value the attribute can have). You use the Items list to select the attribute values that affect pricing. These values will appear in the generated attribute pricing table.

NOTE: The Adjustment Item Generator list is designed to create one set of adjustment items after another. Because of this, when you leave the Adjustment Item Generator list to display another view, Siebel ePricer deletes all the data in the Items list. This feature prevents inadvertent use of leftover data when you use the adjustment item generator repeatedly, but it can also cause you to lose data. To avoid data loss, select values for all attributes in your table and click Generate immediately. To modify this feature, you can configure Siebel ePricer using Siebel Tools as described in "Configuring Automatic Navigation for the Adjustment Item Generator" on page 139.

To generate adjustment items for a multiple attributes table

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, select the attribute pricing table for which you want to generate adjustment items.
3. Click the Attributes view tab.
4. In the Attributes list, make sure that all attributes are listed that are significant for pricing.
5. In the Attributes list, make sure that the Allow Any flag is set correctly for each attribute in the list. If the value for an attribute can ever be omitted in a Multiple Attributes type table—that is, if you will ever allow a blank space (a null value wildcard) to be used—leave the Allow Any box checked for that attribute.

   For more information about the Allow Any option and null value wildcards, see "The Allow Any Option and Null Value Wildcards" on page 120.
6. Click the Adjustment Item Generator view tab.
7. In the Adjustment Item Generator List, make sure all attributes are listed that are significant for pricing purposes, and then do the following:

   a. In the Adjustment Item Generator List, select each attribute in turn.

      When you select an attribute, its domain values appear in the Item Generator list below.

   b. In the Item Generator List, for each value that has significance for pricing, add a check mark to the Affects Pricing field.

      NOTE: If you doubt whether an attribute value affects pricing, it is best to checkmark an attribute value and include it in the Adjustment Items list. If you generate extra rows in the Adjustment Item list, they are easy to delete.

   c. Continue to select attributes in the Adjustment Item Generator list, and specify the values that affect pricing in the Item Generator list, until you have gone through all the records in these lists.

8 Click Generate.

The Adjustment Items list appears, displaying all adjustment items you just generated. All data in the Adjustment Item Generator list is deleted.

Completing the Generated Adjustment Items

The adjustment item records you generate contain the minimum data required by the system. They do not include the actual price adjustment data that they will use or Description text.

When you use the Adjustment Item Generator, Siebel ePricer always inserts the Default Starting Adjustment Type defined in the attribute pricing table header in each record's Adjustment Type field, and it enters zero (0) in Adjustment Value field, because this is a required field.

To complete the generated table, you must replace these defaults with the actual values needed in the Adjustment Type and Adjustment value fields, described in Table 25.

Table 25. Adjustment Types in Attribute Pricing Tables

Adjustment Type	Explanation
Discount Amount	Reduces price by the amount entered in the Adjustment Value field. Valid in Single Attributes or Multiple Attributes adjustment items.
% Discount	Reduces price by the (percentage entered in the Adjustment Value field)/100 multiplied by the original list price. Valid in Single Attributes or Multiple Attributes adjustment items.
Markup Amount	Increases price by the amount entered in the Adjustment Value field. Valid in Single Attributes or Multiple Attributes adjustment items.
% Markup	Increases price by: (percentage entered in the Adjustment Value field)/100 multiplied by the original list price. Valid in Single Attributes or Multiple Attributes adjustment items.
Price Override Amount	Overrides the original list price with the amount entered in the Adjustment Value field. Valid only in Multiple Attributes type tables.

You may also use the Description field to enter text that describes the purpose of the adjustment item.

NOTE: If you have just generated the table, it is automatically displayed in the Adjustment Items list, so you can skip steps 1 through 3 of the following procedure.

To specify pricing adjustment data for an adjustment item

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, select the attribute pricing table that contains the adjustment item for which you want to specify a pricing adjustment.
3. Click the Adjustment Items view tab.
4. Look through the Adjustments Items list and Items list to make sure that the table has been generated as expected, and then do the following:

   a. In the Adjustment Items List, select each adjustment item in turn.

      When you select an adjustment item, its domain values appear in the Item list below.

   b. In the Item List, in each record enter an adjustment type and an adjustment value, as described in Table 25 on page 135, and optionally enter a description.

   c. Continue to select adjustment items in the Adjustment Item list, and to enter price adjustment data in the Item list, until you have gone through all the records in these lists.

Deleting Invalid Adjustment Items

If you are using the Display Error When Invalid Combination option, you must delete any generated adjustment item that contains an invalid combination of attributes.

For example, if you have generated adjustment items for all the colors and sizes of your products, and if your small product does not come in gray, you must delete the record that includes the combination of small and gray.

If your product is available with every combination of attributes, or if you are not using the Display Error When Invalid Combination option, skip this task.

For more information about the Display Error When Invalid Combination option, see "The Display Error When Invalid Combination Option" on page 119.

To delete invalid adjustment items

1. From the application-level menu, choose View > Site Map > Pricing Administration > Attribute Pricing.
2. In the Attribute Pricing list, select the attribute pricing table whose invalid combinations of attributes you must delete.
3. Click the Adjustment Items view tab, and do the following:

   a In the Adjustment Items List, select each adjustment item in turn.

   When you select an adjustment item, its domain values appear in the Item list below.

   b In the Item List, delete any record that contains an invalid combination of attributes by selecting the record, clicking the menu button, and then clicking Delete Record.

   c Continue to select adjustment items in the Adjustment Item list, and to delete invalid records in the Item list, until you have gone through all the records in these lists.

Troubleshooting Attribute Pricing Adjustments

If there are errors in attribute-adjusted prices, to track down their source, you can:

- **Determine which price list the user was using.** To track down the price list for a user you may need that user's login ID and login information (date and time). If the user is a sales agent, then you also need any details about the buyer that might change the price list used.

- **Look up the original list price for the product.** In the Price List screen, select the price list that the user was using, and look at the Price List Line Item record for the product. Note whether the start price includes attribute pricing adjustments.

- **Determine whether there was a promotional price for the product.** Look in the price list to see if there is an entry in the Promotional Price field. If there is one, the attribute-based adjustment is applied to the promotional price, not the list price.

- **Determine whether there was a deal-based price for the product.** Display the quote or order and click the Deals view tab. If there is a deal-based price, the attribute-based adjustment is applied to the deal-based price, not the list price or promotional price.

- **Determine whether there was a volume discount or pricing model for the product.** Look at the Line Item Detail view to see if there was a volume discount. Look at the Pricing Engine Log file to see what volume discount and pricing model price adjustments have been applied. If these exist, they are applied to the attribute-based price after it is calculated.

- **Identify the attribute pricing table that was used.** In the Price List screen, look at the Attribute Pricing field of the Price List Line Item record for the product.

- **Display this attribute pricing table.** In the Attribute Pricing view, select the table header.

- **Find the price adjustment item that was used.** Look at the product attributes the user selected, and compare them with the attributes in the price adjustment item rows in the attribute pricing table, until you find a match.

- **Correct the error.** If there is an error in the price adjustment item, correct it.

Configuring Automatic Navigation for the Adjustment Item Generator

By default, when the Adjustment Item Generator generates a new set of adjustment items, it also prepares the workspace for a new Attribute value selection session by deleting existing data, as described in "Generating Adjustment Items" on page 133.

As a side effect of this standard configuration, if you activate any other view, form, or list that causes the Adjustment Item Generator form to disappear (to be replaced in the foreground of your screen), you lose your work in the Adjustment Item Generator. Even if you clicked Save for each row, ePricer deletes your data input.

You can change the configuration of ePricer so that, after you use the Generate button on the Adjustment Item Generator form, this form does not close or clear away the data you have entered. The form remains as is, with all data, set up for the next use of the Generate function.

This is useful if you generate many attribute pricing tables in one session, and you do not wish to leave the Adjustment Items forms or lose the values you have entered in the Affects Pricing field.

In Siebel Tools, alter the User Property called "Result Items View" in the Price Book Pricing Item Maker Single List applet.

This User Property identifies the view and forms that the system opens after you use the Generate button. If the view is set to an empty field, then the Generate button does not automatically activate another view, deleting existing data. The data is preserved, so you can use it as the basis of the next set of attribute pricing items that you generate.

For more information about Siebel Tools, see *Siebel Tools Reference*.

Creating Pricing Models and Pricing Factors 10

This chapter gives a general overview of pricing models and of the different types of pricing factors that they can use. It looks in more detail at single type pricing factors. Finally, it describes the process of creating and applying a pricing model, using a pricing model with a single-type pricing factor as an example.

This chapter covers the following topics:

About Pricing Models and Pricing Factors

You can design your own price adjustments using pricing models and pricing factors:

- **Pricing model.** A collection of pricing factors, used to work with the pricing factors as a group.
- **Pricing factor.** A rule to create a specific price adjustment. Each factor is an if ... then ... statement, with a logical condition and a price adjustment that is applied if the condition is true.

Siebel ePricer allows you to create several types of pricing factors. One pricing model may include pricing factors of different types.

This chapter discusses:

- **Types of pricing factors.** A summary of the different types of pricing factors, so you can decide which you need.
- **Single type pricing factor.** A discussion of how to design the single type pricing factor, which is the simplest type of pricing factor.
- **Process of creating and applying a pricing model and pricing factor.** Describes how to create and apply a pricing model that includes a single type pricing factor. A similar process is used with the other types of pricing factors described in other chapters.

NOTE: If you use pricing models with other types of price adjustments, the final price depends on the order in which Siebel ePricer applies the pricing adjustments. For more information, see "The Processing Order of Price Adjustments" on page 19.

Types of Pricing Factors

Siebel ePricer allows you to create the following types of pricing factors:

- **Single.** Includes a single rule that applies a price adjustment to individual line items in a quote, order, agreement, or component-based product. For example, you can give the customer a discount if the extended price of a line item is more than $1,000. For more information, see "About Single Type Pricing Factors" on page 147.

- **Matrix-Based.** Equivalent to several single-type pricing factors. You create a Siebel business component with records specifying multiple rules for price adjustments, and the pricing factor searches this business component to determine which price adjustments to apply. You can import information into this business component from a pricing table in another application, such as an Excel spreadsheet. For more information, see Chapter 11, "Matrix-Based Pricing Factors."

- **Bundling.** Allows you to apply price adjustments that depend on the selection of bundles of products. For example, if a customer buys a dining room table and four chairs, the customer can get a 10% discount on a sideboard. For more information, see Chapter 12, "Bundling and Aggregate Pricing Factors."

- **Aggregate.** Applies the pricing factor to the entire quote or order. For example, you can give a customer a 10% discount if the total price of a quote is over $100,000. For more information, see Chapter 12, "Bundling and Aggregate Pricing Factors."

- **Script-Based.** Allows you to use a script (in Siebel eScript or Visual Basic, for example) to define more complex pricing factors. For example, these might include complex mathematical calculations, multiple business components, or a compound rule on multiple fields in the same business components. For more information, see Chapter 13, "Script-Based Pricing Factors."

Siebel ePricer also gives you tools to manage pricing factors:

- **Managing the Order in which Pricing Factors are Executed.** The order in which pricing factors are executed affects the final price. Siebel ePricer provides a feature for controlling this order and creating complex logic flows. For more information, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."
- **Testing Pricing Factors.** Siebel ePricer provides features for testing pricing factors. For more information, see Chapter 16, "Testing and Validating Pricing Models."

NOTE: Pricing factors for component-based products use the same general procedures as pricing factors for other products. However they involve additional constraints. You should not work with them until you have read "About Pricing Models for Component-Based Products" on page 111.

Business Components Used with Pricing Factors

Pricing factors include rules to determine whether a price adjustment is applied depending on the value of a field of a business component. For example, the customer might get a discount of the value of the extended price field of the line item of a quote is greater than $1,000.

Table 26 and the notes that follow it summarize which business components can be used with each type of pricing factor.

Table 26. Business Component Names Used for Pricing Factor Types

Business Component Name	Single, Script, and Matrix Types	Bundle Type	Aggregate Type
Agreement Item XA	✔		
FS Agreement Item	✔	✔	✔
Order Entry - Line Item	✔	✔	✔
Order Entry - Orders	✔		
Order Entry XA	✔		
Price List	✔		
Price List Copy	✔		
Price List Item	✔		
Quote	✔		
Customizable Product Header	✔		
Quote Item	✔	✔	✔
Customizable Product	✔	✔	
Quote Item XA	✔		
Customizable Product XA	✔		
Service Agreement	✔		

Notes on Table 26 on page 145:

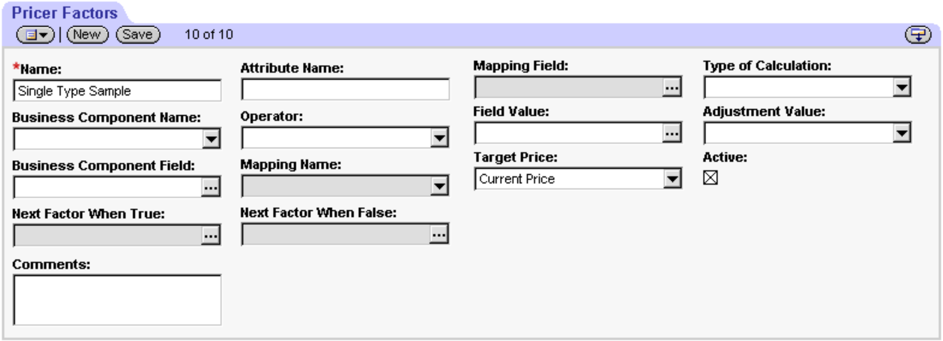
- The business components named Customizable Product Header, Customizable Product, and Customizable Product XA can be used only for pricing models for component-based product.
- The business components named Price List, Price List Copy, and Price List Item can be used only for transforming a price list. The business component named Price List Copy contains an account field. The business component named Price List has no account field. If you intend to reference an account in a pricing model when you transform a price list, use Price List Copy rather than Price List.
- If you create a pricing factor that references a business component with attribute data (identifiable by the characters XA in the buscomp name), do not enter a value in the Business Component Field.

NOTE: Table 26 on page 145 refers only to the business components used in the Business Component Name field of the pricing factor.

About Single Type Pricing Factors

A single type pricing factor includes a condition and a price adjustment that will be applied if the condition is true.

You define a single type pricing factor using the Pricing Factors form shown in Figure 1.



Pricer Factors

*Name: Single Type Sample	Attribute Name:	Mapping Field:	Type of Calculation:
Business Component Name:	Operator:	Field Value:	Adjustment Value:
Business Component Field:	Mapping Name:	Target Price: Current Price	Active: ☒
Next Factor When True:	Next Factor When False:		
Comments:			

Figure 1. The Pricer Factors Form for Single Type Pricing Factors

The condition compares the value in a field (or optionally, a field attribute) with a value that you specify. Specify the condition using the following fields:

- **Business Component Name, Business Component Field, Attribute Name.** Specify a business component, a field, and optionally a field attribute.
- **Operator.** Specify an operator such as =, <, or >.
- **Field Value.** Specify a value to compare the field’s value with.

CAUTION: If the business component field is a date-time field, the field value must be entered in it using a 24-hour clock (military time). The pricing factor will not run if the time includes A.M. or P.M.

The price adjustment is performed if the condition is true. Specify the price adjustment using the following fields:

- **Type of Calculation.** Specify what type of adjustment should be done, for example, a % discount, a markup amount, or a price override.
- **Adjustment Value.** Specify the amount of the adjustment.

For example, you can create a single type pricing factor saying that, if the product line is desktops, the customer gets a 5% discount on that line item. Enter the following information:

- **Business Component Name.** Quote Item
- **Business Component Field.** Product Line
- **Operator.** =
- **Field Value.** Desktops
- **Type of Calculation.** % discount
- **Adjustment Value.** 5

NOTE: In addition to this simple comparison of a field with a value, single type pricing factors can compare a field with a field in another business component to determine if the price adjustment should be applied. For more information, see Chapter 14, “Using Mappings in Pricing Factors.”

The Process of Creating and Applying a Pricing Model

To create and apply a pricing model and pricing factor, go through the following process:

1. "Creating a Pricing Model"
2. "Creating Pricing Factors" on page 151
3. "Linking the Pricing Model to a Price List or Customizable Product" on page 157
4. "Testing the Pricing Model" on page 158
5. "Reloading the Pricing Model into the Cache" on page 159
6. "Unlocking the Pricing Model" on page 159

NOTE: You can also create a new pricing model by using the Pricing Model Manager's menu to copy an existing one. This copies all pricing factors in the Pricing Model as well. Then you can rename the model and modify it as needed.

Creating a Pricing Model

Use the Pricing Model Manager list to create a new pricing model and enter general information about it. The pricing model record includes the following information:

- **Locked.** Pricing models include a locking mechanism so that only one pricing administrator can edit a pricing model and its factors at a time. The model is automatically locked when you create it. If you unlock it, it becomes available to customer.
- **Currency.** The base currency you assign to a pricing model must always be the same as the base currency of the price list with which the model is associated. If you recreate a price list in a foreign currency, you must also recreate the pricing model for that price list with the same base currency.
- **Start and End.** The pricing model has a Start Date and End Date field, which controls when it can execute. Each pricing factor also has a Start Date and End Date, which controls when it can execute.

NOTE: You can use the same pricing model for several price lists, but each price list can have only one pricing model associated at the topmost header level.

To create a new pricing model

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.
2. In the Pricing Model Manager list, click New.
3. Enter information in the new record and in More Info form, as described in Table 27.

Table 27. Pricing Model Manager List Fields

Field	Entry
Name	Required. Enter a unique, meaningful name for the pricing model; 30 characters maximum. Blanks and hyphens may be used, but other special characters may not be used.
Customizable Product	If a component-based product is associated with this model, select it here. There are special constraints for pricing models for component-based products. For more information see "About Pricing Models for Component-Based Products" on page 111.
Description	Optional. Enter a description of the pricing model.
Locked	By default, a pricing model is locked when it is created. A pricing model must be locked whenever it is being modified. Unlocking a pricing model makes it available for use by the runtime users.
Locked By	The system updates this field when the pricing model is locked. Only the person who locked the model or a system administrator can modify the pricing factors within the pricing model.
Locked Date	The system updates this field when the pricing model is locked. When the pricing model is unlocked, it is blank.
Currency	Required. Select the currency used by the model. The pricing model can only be associated with price lists that use the same currency. The default is USD (U.S. Dollars).
Start	Required. Enter the date when the pricing model becomes activate. The default is the date created. The Start Date must be later than or equal to Date Created.
End	Optional. Enter the date when the pricing model becomes inactive. If it is not specified, the pricing model remains active indefinitely. The End Date must be later than or equal to the Start Date.

Creating Pricing Factors

You use two applets to create a pricing factor:

- **Pricing Factor Designer list.** Used to create the new pricing factor record, select its Type, and enter general information about it.
- **Pricer Factors form.** Used to define the quantitative price adjustment. This form changes depending on the Type you select in the Pricing Factor Designer list, because you define the price adjustment differently for different types of pricing factors.

NOTE: If you do not select the pricing factor type, then a generic version of the pricing factor form appears. The generic version may contain fields that do not apply to the type of pricing factor you want to create.

To create a new pricing factor record

1. From the application-level menu, choose View > Site Map > Pricing Administration >Pricing Model Manager.
2. In the Pricing Model Manager list, select the Pricing Model in which you want to create a new pricing factor.
3. If the Locked field does not already contain a checkmark, click this field to lock the pricing model.
4. Click the Pricing Factors Designer view tab.

   The Pricing Factors Designer list appears.

5 Click New.

A new pricing factor record appears in the Pricing Factors Designer list, and a new Pricer Factors form appears below.

6 Enter information in the new record in the Pricing Factors Designer list, as described in Table 28.

Table 28. Pricing Factor Designer Form Fields

Pricing Factor Designer Form Field	Entry
Sequence	A number indicating the order in which factors execute. The system assigns a default value, which you can edit. You can also change the order using the Pricing Factor Flow Chart Designer. Single, matrix-based, and script-based type factors should come before bundling and aggregate factors.For more information about the order in which factors should execute, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."
Name	Enter a unique, meaningful name for the new pricing factor, to replace the system-generated character string.
Type	Select the type of pricing factor you need to create. For information about which type you need, see "Types of Pricing Factors" on page 143. This chapter covers Single type pricing models, so to continue this process, select "Single."
Start	Required. Enter the date on which the pricing model becomes active.
End	Optional. Enter the date on which the pricing model becomes inactive. If no date is entered, the pricing model remains active indefinitely.
Comments	Optional. Enter an explanation of this pricing factor for the end user. The comment text entered here appears in the Pricing Comments field of quotes and orders that use this pricing factor.

To complete the Pricer Factors form

1 In the new Pricer Factors form, enter the information described in Table 29.

NOTE: Table 29 describes the fields that appear in the Pricer Factors form when you choose Single in the Type field in the Pricing Factor record. Pricer Factors forms for other types of pricing factors are described in later chapters.

Table 29. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Name	Replace the system-generated name with a unique, meaningful name.
Business Component Name	Select a Siebel business component containing the data that the pricing factor will compare with the value you enter in the Field Value field to determine whether to apply this pricing factor.
Business Component Field	Select the field in the Siebel business component containing the data that the pricing factor will compare with the value you enter in the Field Value field.
Attribute Name	For single type pricing factors that reference an attribute (XA) business component, enter the name (not the Display Name) of the attribute to be used in the pricing factor. This name is assigned in the Applications Administration > Class Administration > Dynamic Attributes list.
Operator	Select an operator to be used in the comparison. The operators are: =, < >, <, < =, >, > =, EXISTS IN, DOES NOT EXIST IN. If you select an arithmetic operator, the Field Value field becomes available. If you select EXISTS IN or DOES NOT EXIST IN, Mapping Name and Mapping Field become available.

Table 29. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Field Value	If you are using an arithmetic operator, enter a character string or a number that will be compared to the value in the Siebel BusComp Field. Character strings and numeric constants do not need to be enclosed in quotes. You can enter multiple values in the Field Value field. Click the field to display a dialog box where you can enter a list of values. The pricing engine connects these values with an implied OR. If the business component field is a date-time field, the field value must be entered in it using a 24-hour clock (military time). It may not contain A.M. or P.M. For single pricing factors: ■ If the Pricer Rule Operator is = or < >, all values in the dialog box are considered. ■ If the Pricer Rule Operator is >, > =, <, or < =, only the first value in the dialog box is considered. For aggregate pricing factors only the first value in the dialog box is considered, regardless of the specified operator.
Mapping Name	If the operator is EXISTS IN or DOES NOT EXIST IN, select a mapping from the drop-down list. For more information, see Chapter 14, “Using Mappings in Pricing Factors.”
Mapping Field	If the operator is EXISTS IN or DOES NOT EXIST IN, select a mapping field from the drop-down list. For more information, see Chapter 14, “Using Mappings in Pricing Factors.”
Next When True	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as true. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”

Table 29. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Next When False	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as false. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”
Type of Calculation	Select a calculation that will be used to adjust the target price. The options are: ■ Discount Amount decreases the target price by the specified amount ■ % Discount decreases the target price by the specified percentage value ■ Markup Amount increases the target price by the specified amount ■ % Markup increases the target price by the specified percentage value ■ Price Override replaces the existing current price with the specified price ■ Power (Target Price, Power) raises the target price to the specified exponential value. For example, if the adjustment value is 2, the target price will be raised to the power of 2, which would yield 4. A target price of $100 with an exponent of 1.176 results in a selling price of $224.9055. ■ Multiplicative Amount multiplies the target price by a specified amount. ■ Round (Current, Decimal Places) sets the number of decimal places to which the price will be rounded.

Table 29. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Adjustment Value	Enter the amount to be used in the calculation to adjust the target price. For example, if the type of calculation is a % Discount and the adjustment value is 10, a 10% discount is applied to the target price. If the type of calculation is Discount Amount and the adjustment value is 10, then a $10 discount is applied to the current price (assuming currency is USD). If the type of calculation is Price Override and the adjustment value is 10, then the price becomes $10 (assuming currency is USD).
Target Price	Select the price that will be used as the basis of the price adjustment calculation. The options are: ■ Base = MSRP. The MSRP in the price list line item detail. ■ Current Price. The price in its current state, which includes all adjustments that ePricer has completed when this pricing factor executes. ■ Base = Cost. The cost in the price list line item detail. ■ Base = List Price. The list price in the price list. If a promotional price is defined for an item, the promotional price is used instead. ■ Base = Purchase. The Purchase Price in the price list line item detail. If the adjustment applies a static amount, the target price is not a part of the adjustment calculation. If the adjustment is a price override, the target price is not relevant.

Table 29. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Active	Click the check box to activate or deactivate the pricing factor. A pricing factor will not execute when it has been deactivated. If it appears in a decision flow path, it will be interpreted as a factor that did not meet the conditions for execution.
Comments	Optionally, enter comments explaining this pricing factor to the end user. The comment entered here appears in the quote's or order's Pricing Comments field.

Linking the Pricing Model to a Price List or Customizable Product

Before you can test or release a pricing model, you must link it to a price list or customizable product.

Each price list can have only one pricing model linked to it at a time.

Pricing factors for component-based products involve additional constraints. For more information, see "About Pricing Models for Component-Based Products" on page 111.

For testing, you should leave the model locked and link it to a price list or component-based product. While it is locked, it can be tested using the validation screen, but it does not apply to actual quotes and orders that use the price list.

NOTE: If you modify either a pricing model or a price list or component-based product after the two are linked, you may need to exit and restart Siebel ePricer to see the effects of your changes. Whether this is necessary depends on how your price list and pricing model caching is set up. Your Siebel system administrator can change the cache refresh rules and reset the timing of cache date updates.

To link a pricing model to a price list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price Lists.

   The Price Lists list appears.

2. Select the price list to which the pricing model should be linked.
3. In the Pricing Model field, use the drop-down list to select a pricing model.

To link a pricing model to a customizable product

1. From the application-level menu, choose View > Site Map > Pricing Administration > Price Lists.

   The Price Lists list appears.

2. In the price list item that represents the customizable product, in the Customizable Product Pricing Model field, select the pricing model to which the customizable product should be linked.

Testing the Pricing Model

Before you unlock the pricing model to make it available to users, you should test it using the pricing model validation screens, which are described in Chapter 16, "Testing and Validating Pricing Models."

Reloading the Pricing Model into the Cache

To enhance performance, pricing models are cached when they are initially executed by a quote or order or initially unlocked.

After the cache has been created once, you must reload the model to cache any changes.

Before unlocking a pricing model to release it to users, you should save it and reload it into the cache, to make sure that users have the latest version of that model in the cache.

To reload a pricing model into the cache

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model you want to reload.
3. In the Pricing Model Manager list, click the menu button, and then Reload.

Unlocking the Pricing Model

You make a pricing model available to users of the price list or component-based product by unlocking the pricing model.

A model can be unlocked only by the Pricing Administrator who locked it.

To release (unlock) a pricing model

1. From the application-level menu, choose View > Site Map > Pricing Administration> Pricing Model Manager.
2. In the Pricing Model Manager list, select the pricing model you want to unlock.
3. In the Locked field, click to remove the checkmark from the check box.

Matrix-Based Pricing Factors 11

This chapter describes matrix-based pricing factors, which allow you to create a business component with pricing information that is referenced by the pricing factor. It gives instructions about how to set up the pricing factor and business component for matrix-based pricing.

This chapter covers the following topics:

About Matrix-Based Pricing

To use matrix-based pricing, you must set up a business component in addition to the pricing factor:

- **Matrix-pricing business component.** Allows you to define an entire list of pricing adjustments, which are can depend on other fields in addition to the product.
- **Matrix-based pricing factor.** Specifies which matrix-pricing business component to use, and which fields to search in to see if the discount applies.

For example, you want to give your customer accounts different discounts on a variety of products. Among other discounts, you want to give the A. K. Parker account 10% off when they purchase the product called Laptop 123. You want to give other accounts different discounts on this product.

You create a a matrix pricing business component that lists the discount rates that you will give to each account for each product. It includes the following:

Account	Product	Adjustment Type	Adjustment Amount
A. K. Parker	Laptop 123	% Discount	10

You also create a matrix-based pricing factor that identifies this business component and specifies that Siebel ePricer should search the Account field and the Product field to see whether a discount applies.

When a user purchases products using Siebel quotes, orders, or shopping cart, Siebel ePricer searches through the matrix to see if there is a match. If the Account is A. K. Parker, and the Product is Laptop 123, it gives a 10 percent discount.

NOTE: You must be an advanced user to work with business components. If you are not familiar with Siebel Tools and Siebel application architecture, work with a Siebel system administrator. For information about Siebel business components, see *Siebel Tools Reference*.

About Matrix Pricing Business Components

To use a matrix-based pricing factor, you must a create a Siebel business component with the matrix-based pricing data. You can import pricing information from external systems like spreadsheets, databases, or mainframe files into the business component by using Siebel Enterprise Application Integration (eAI).

Matrix pricing business components that are used to store matrix pricing data contain multiple records in a format like that of a spreadsheet or data table. Each record includes key data that is used to search for the record, and the corresponding price adjustment.

Some existing Siebel business components are preconfigured to work with matrix-based pricing factors. You can also develop your own business components to store pricing data.

This section begins by describing a predefined Siebel business component that provides matrix pricing data, FS Entitlement Pricing Details. Then it describes the creation of customized business components.

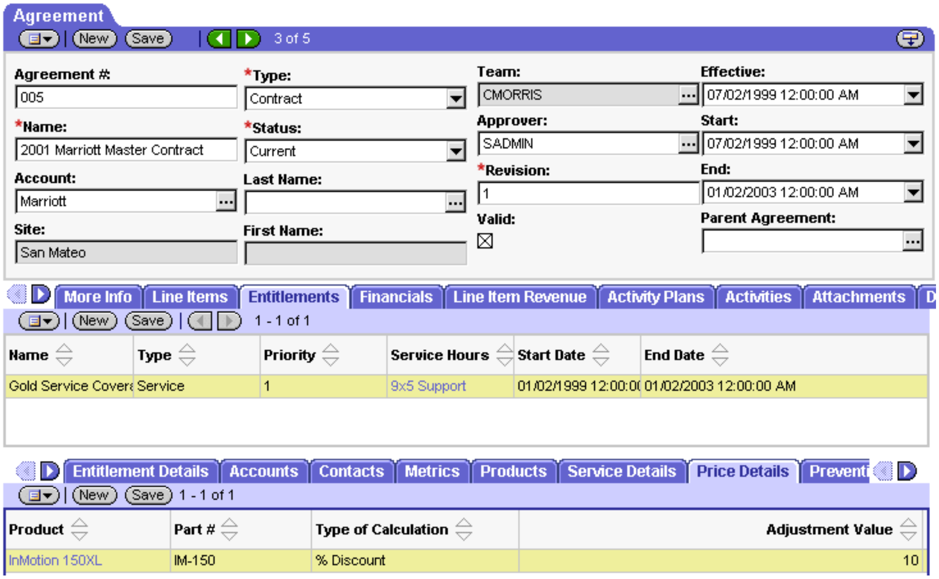
A Preconfigured Matrix-Pricing Business Component

Siebel ePricer system is shipped with business components that are preconfigured for use with matrix-based pricing factors, including the following business components, which are part of the Service Agreement business object.

- Service Agreement
- Agreement Entitlement
- FS Entitlement Pricing Details

You can see the data in these business components by using the Agreement Entitlement Pricing Details view, shown in Figure 2. This includes the Entitlements and Pricing Detail lists, the views corresponding to the business components that we are using to illustrate matrix-based pricing.

NOTE: In the default configuration of Siebel ePricer, entitlement data appears only in the order view, so in the example, the search specification for entitlement matrix data refers to order data, not quote data.



Agreement #: 005	*Type: Contract	Team: CMORRIS	Effective: 07/02/1999 12:00:00 AM
*Name: 2001 Marriott Master Contract	*Status: Current	Approver: SADMIN	Start: 07/02/1999 12:00:00 AM
Account: Marriott	Last Name:	*Revision: 1	End: 01/02/2003 12:00:00 AM
Site: San Mateo	First Name:	Valid: ☒	Parent Agreement:

Name	Type	Priority	Service Hours	Start Date	End Date
Gold Service Covera	Service	1	9x5 Support	01/02/1999 12:00:0	01/02/2003 12:00:00 AM

Product	Part #	Type of Calculation	Adjustment Value
InMotion 150XL	IM-150	% Discount	10

Figure 2. Agreement View with Entitlements List and Price Detail List

In this example, Entitlement ID and Product ID are used as key fields that identify the discounts for which an account qualifies. When users provide data for the Entitlement ID and Product ID in an Order, Siebel ePricer applies the applicable price adjustments to the order.

In this example, the matrix-based pricing factor’s search criteria seek matching data in the FS Entitlement Pricing Details business component. If an order specifies an entitlement ID and a product ID, and the same agreement entitlement and product values are in the FS Entitlement Pricing Details business component, then the match of these values triggers a pricing adjustment.

The standard FS Entitlement Pricing Details business component records contain the Type of Calculation and Adjustment Value field data from the Entitlement Price Details form. The matrix factor SearchSpec causes this adjustment data to be applied to the item price.

For more information on maintaining the information in these fields, see *Applications Administration Guide*.

A New Matrix-Pricing Business Component

You can build your own matrix-pricing business components to suit your purposes. For example, you can design a custom business components so it can be populated with data imported from spreadsheets developed by different pricing specialists, distributing the price administration workload.

The matrix-pricing business component must contain the following fields:

- **Adjustment Amount.** The business component must identify one column that holds the adjustment amount and designate it as the adjustment amount through its user properties, which the pricing engine will use to identify the field. To do this, create a new business component user property named *Pricer Adjustment Value*. As the value of this user property, specify the name of the column in the business component that stores the adjustment amount.

- **Type of Calculation.** The business component must identify one column that holds the type of calculation, also known as the adjustment type. To do this, create a new business component user property named *Property Adjustment Method*. As the value of this user property, specify the name of the column in the business component that stores the adjustment type.

  For more information about defining data types and tables, see the section about creating tables in *Siebel Tools Reference*.

About Matrix-Based Pricing Factors

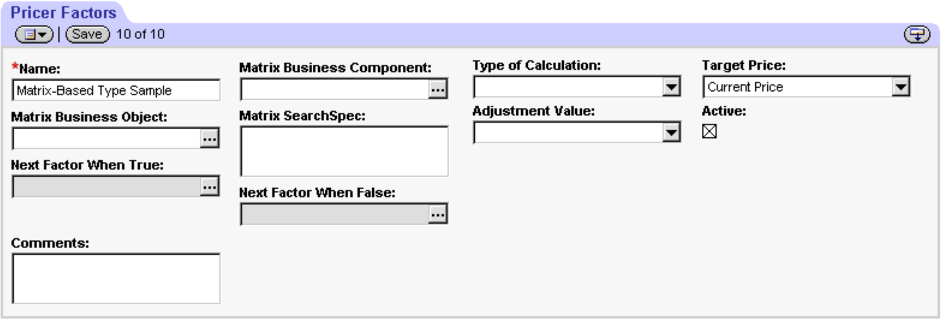
Because a matrix-based pricing factor refers to all the data in a matrix pricing table, it can replace many single pricing factors.

NOTE: Because matrix-based pricing factors search data residing in Siebel business components, their performance depends on the efficiency with which data can be retrieved when a search specification is executed. Matrix-based pricing factors can be tuned by ensuring proper indexes exist for the underlying database tables.

The Pricer Factors Form for Matrix-Based Type Pricing Factors

The form you use to create a matrix-based pricing factor, shown in Figure 3, does not specify the criteria that determine whether a price adjustment will be applied. Instead, it has fields to specify the business component and search specifications that Siebel ePricer will use to find the pricing rules. The type of calculation and the price adjustment value comes from records in the matrix pricing business component.

However, the target price value for a matrix-based pricing factor (the base price that the adjustment is applied to) is specified in the Pricing Factor form.



Pricer Factors

Save 10 of 10

*Name: Matrix-Based Type Sample

Matrix Business Object:

Next Factor When True:

Comments:

Matrix Business Component:

Matrix SearchSpec:

Next Factor When False:

Type of Calculation:

Adjustment Value:

Target Price: Current Price

Active: ☒

Figure 3. The Pricer Factors Form for Matrix-Based Type Pricing Factors

Search Specifications for a Matrix-Based Pricing Factor

Search specifications are entered in the Matrix SearchSpec field of the Pricing Factor form. They are logical conditions that use fields in the quote or order as variables. For any quote or order, only one row in the matrix-pricing table should satisfy the condition, and the price adjustment in this row is applied to the order.

NOTE: If the search specification in this field can describe more than one record, then the search returns the first matching record that it finds, which may vary from one session to another. You should test search specifications to make sure that they cannot be satisfied by multiple records.

Search specifications use the standard Siebel query syntax and formatting rules, as follows*:*

- Enclose the field name in square brackets, for example, `[Entitlement Id]`
- Specify all names (business object, business component, field) with the same capitalization, spacing, and punctuation used in Siebel Tools.
- Join multiple search criteria with AND or OR.
- Repeat the field name when you are comparing a field to multiple values.

For more information about the standard syntax used in search specifications and queries, see *Siebel Tools Reference*.

The following examples will give you a general idea of search specification formatting and syntax. These examples were created to function in a specific context, and they may not to work in your environment.

- The following searchspec would be used to give discounts to customers based on their entitlement and the product ordered; it would be used with the sample business component described in "A Preconfigured Matrix-Pricing Business Component" on page 163.

  ```
  [Entitlement Id]={Order Entry - Orders.Entitlement Id} AND
  [Product Id]={Order Entry - Line Items.Product Id}
  ```

- The following search spec would be used to give discounts to customers based on the Service Agreement's price list, the service product and the covered product (the product for that service is for).

  ```
  [Price List Id]={Service Agreement.Price List Id} AND [Parent
  Product Id] = {FS Agreement Item.Product Id} AND [Product Id] =
  {FS Agreement Item.Covered Asset Product Id}
  ```

The Process of Setting Up Matrix-Based Pricing

To set up matrix-based pricing, you go through the following process:

1. "Creating a Matrix-Based Pricing Factor"
2. "Setting Up a Matrix-Pricing Business Component" on page 171

Creating a Matrix-Based Pricing Factor

This procedure assumes that you are familiar with the overall process for creating and applying a pricing model, described in "The Process of Creating and Applying a Pricing Model" on page 149.

NOTE: When you define matrix-based pricing factors, the Comment field is especially important. The purpose and details of the matrix are difficult to summarize using just the name of the pricing factor. This field is a good place to record such data as the filenames for spreadsheets in which pricing specialists record original matrix pricing data.

To create a matrix-based factor

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model in which you want to create a new aggregate type pricing factor.
3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark in order to lock it.
4. Click the Pricing Factor Designer view tab.
5. In the Pricing Factor Designer list, click New.

6 In the new record in the Pricing Factor Designer list, enter the information described in Table 28 on page 152, and select Matrix-Based in the Type field.

7 In the new Pricer Factors form, enter the information described in Table 30.

Table 30. Matrix-Based Pricing Factor Fields

Field	Entry
Name	Required. Replace the system-generated character string in the Name field with a meaningful name that is unique within the current pricing model.
Matrix Business Object	Required. Click the select button and choose the business object that contains the business component that holds the matrix pricing data.
Matrix Business Component	Required. Click the select button, and choose the business component that holds the pricing matrix data.
Matrix SearchSpec	Required. Create a search specification using the standard Siebel syntax and formatting rules, which are explained in *Siebel Tools Reference*. For more information, see "Search Specifications for a Matrix-Based Pricing Factor" on page 167.

Target Price	Required. Select the target price (the base price that will be modified by any price adjustment data returned from the matrix search of the matrix business component).
	Base = MSRP
	Current Price
	Base = Cost
	Base = List Price
	Base = Purchase

Active	Click the check box to activate or deactivate the pricing factor.
Comments	Optional. Enter such data as the filenames for spreadsheets in which pricing specialists record original matrix pricing data.

Setting Up a Matrix-Pricing Business Component

This section includes general guidelines for setting up matrix-based business components. It does not include specific instructions. Only people who are experienced in working with Siebel Tools should set up matrix-based business components.

To set up a matrix-pricing business component, you go through the following process:

1. "Designing a Matrix-Pricing Table"
2. "Creating a Matrix-Pricing Table" on page 172
3. "Creating a Matrix-Pricing Business Component" on page 172
4. "Loading the Matrix-Pricing Data" on page 173 (Necessary only if you created the matrix pricing table in an external application.)

For more information about using Siebel Tools, see *Siebel Tools Reference*.

Designing a Matrix-Pricing Table

Matrix-pricing tables are similar to spreadsheets or database tables where different price adjustments can be defined for each row that has a unique set of key values. These key values are the pricing criteria.

If Siebel ePricer searches finds a match when it searches these key values, then it can apply that row's price adjustment. The search criteria are always based on values in a quote or order, and the format of the key values in the search criteria must be logically comparable to the values in the quote or order.

For example, a distributor of consumer goods needs to apply a simple price markup in order to cover shipping charges. The markup could depend on the following fields: City of Origin, Destination City, Shipment Method, and Shipment Priority. The matrix must be set up with one row for each possible combination of these values, and the markup for that combination of values. This table could be maintained in a spreadsheet and periodically imported to a Siebel component.

A simpler example has just two key values, Entitlement ID and Product ID in the Product Entitlements table. The discount that a customer gets depends on the customer's entitlement and on the product the customer is purchasing. You must set up a matrix with rows for every possible combination of customer and product, and with the discount for each combination.

Creating a Matrix-Pricing Table

For business components that are not based on existing tables, you will need to create a table using Siebel Tools.

When you create database tables to support Matrix business components, in addition to the key values described previously, you must include two fields:

- **Adjustment Amount.** Must be numeric and can hold a value that represents a markup, a discount, or a price override.
- **Type of Calculation.** Also known as adjustment type. Tells the pricing engine how to apply the adjustment amount. This field should be based on the PRICE_CALCULATION_TYPE LOV, which limits the selection of price calculation types to a valid list of values. The options are: % discount, amount discount, % markup, amount markup, price override, multiplicative amount, power (target price, power) and round (current, decimal places). For more information about these options, see Chapter 10, "Creating Pricing Models and Pricing Factors."

For more information about defining data types, see the section about creating tables in *Siebel Tools Reference*.

Creating a Matrix-Pricing Business Component

For business components that are not based on existing tables, you will need to create a business component using Siebel Tools.

When you create a matrix-pricing business component, you must include two fields corresponding to the matrix-table fields:

- **Adjustment Amount.** The business component must identify one column that holds the adjustment amount and designate it as the adjustment amount through its user properties, which the pricing engine will use to identify the field. To do this, create a new business component user property named *Pricer Adjustment Value*. As the value of this user property, specify the name of the column in the business component that stores the adjustment amount.

- **Type of Calculation.** The business component must identify one column that holds the type of calculation, also known as the adjustment type. To do this, create a new business component user property named *Pricer Adjustment Method*. As the value of this user property, specify the name of the column in the business component that stores the adjustment type.

For more information about defining data types and tables, see the section about creating tables in *Siebel Tools Reference*.

Loading the Matrix-Pricing Data

If you created the matrix-pricing table in a Siebel business component, this data is already present in the component, and you do not have to load it.

You do have to load the data when you are:

- Loading rules from another system that stores pricing rules in a matrix-like form
- Synchronizing rules from another system that stores pricing rules in a matrix-like form

For example, if you want to integrate Siebel ePricer with SAP, you must load the pricing rules that are stored in SAP tables into Siebel tables.

One way of loading the rules is to map the SAP tables to Siebel tables. Then the load is handled through Siebel Enterprise Integration Manager (EIM) or Siebel eBusiness Application Integration (eAI).

For help with Siebel EIM or eAI, contact your Siebel system administrator.

For more information about Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

For more information about Siebel eAI, see *Overview: Siebel eBusiness Application Integration Volume I*.

Improving Performance of Matrix Pricing

Because pricing matrices often have many rows of data, it is important to set up the business components for matrix pricing in the most efficient way.

The underlying table containing the pricing matrix data should contain integration ids that are indexed. If you need to export this pricing matrix information between systems, you must use ids that can be used by all of these systems; these common ids need to be indexed.

To improve performance, you can create two business components for this table, one used for administration and one used in run time. The Admin bus comp would contain the links, joins, pick lists and associated views, applets, pick applets, and pick maps that the administrator uses to create the pricing matrix data. The Run time bus comp, would not contain any of the links, joins, or pick lists that the administrator uses. The Run time bus comp would contain only the columns needed to execute the matrix look up, such as product integration id and account type.

Bundling and Aggregate Pricing Factors 12

This chapter discusses bundling type pricing factors and aggregate type pricing factors, the two types of pricing factors that apply to groups of products. It describes and gives instructions about how to create both these types of pricing factors. It also covers bundling type pricing factors for component-based products.

This chapter covers the following topics:

About Bundling and Aggregate Type Pricing Factors

Bundling type pricing factors and aggregate type pricing factors work with groups of items:

- **Aggregate type factors.** Address all the line items in the quote, order, or agreement as a complete set; that is, the aggregate of line items.
- **Bundling type factors.** Address specified combinations of line items—either the components within a component-based product or the line items in the quote, order, or agreement.

For example, you can use aggregate type pricing factor create package deals such as the following, which are based on all the items in the quote, order, or agreement:

- Apply a discount when 10 or more different items are in the ordered
- Apply a discount when the total for all items ordered is greater than $1,000,000.

You can use bundling type pricing factor to create package deals such as the following, which are based on only some items in the quote, order, or agreement:

- Buy A and B, get $x or x% off B
- Buy A and B, get C for free
- Buy A and B, get $x or x% off C

Processing Order of Bundling and Aggregate Type Pricing Factors

Because these two pricing factor types apply to multiple items within the quote, order or agreement, they are applied at a different time from other types of pricing factors:

- **Single, matrix-based, or script-based pricing factors.** Applied when a user adds an item to the shopping cart or changes the quantity of the item in a quote, order, agreement, or component-based product.
- **Aggregate and bundling pricing factors.** Applied when the user saves a quote, clicks Reprice All, or checks out. (Not applied if the system saves a quote—for example, because a data connection is broken.)

Siebel ePricer must apply all the single, matrix-based, or script-based type pricing factors in the model before it can apply the bundling or aggregate type factors. It must apply adjustments to individual product prices (single line items) before it can apply aggregate adjustments to groups of products (multiple line items).

In addition, Siebel ePricer must apply bundling type pricing factors before aggregate type pricing factors. Bundling type factors can change the price of a subset of products in the quote, order, or agreement, which affects the total price. For this reason, they should be applied before aggregate type pricing factors that are based on the total price.

When you are specifying the sequence of pricing factors, be sure that bundling type factors come before aggregate type factors, be sure they are executed in the following order:

- Single, matrix-based, or script-based type pricing factors
- Bundling type pricing factors
- Aggregate type pricing factors

Use this order when you create the pricing model and assign sequence numbers to pricing factors, and when you use the Pricing Factor Flow Chart Designer to specify the sequence of pricing factors.

If a bundling type pricing factor is in a component-based product pricing model, then ePricer must complete the single pricing factor adjustments on all components in that model before it automatically executes the bundling factors within the component-based product configuration session.

NOTE: Pricing models for component-based products can include bundling type pricing factors but cannot include aggregate type pricing factors.

Constraints on Using Bundling and Aggregate Type Pricing Factors

Aggregate and bundling type pricing factors must always reference a business component that contains data specific to the current quote, order, or agreement. This is because, at runtime, the aggregate factor adds up records in the current business component in its current state.

Neither bundling type pricing factors nor aggregate type pricing factors should ever appear in a pricing model that is used to transform a price list.

Aggregate type factors may not be used in pricing models for component-based products with components. If they do, the model will base logical tests on both component and noncomponent products, and it will use these items to determine the price of the component-based product. The result will be incorrect and unpredictable pricing.

If a bundling type pricing factor is applied to a component-based product, it evaluates the customizable product as it is defined in the current session.

Aggregate Type Pricing Factors

Aggregate type factors include a condition statement and a price adjustment that is applied if the condition is true.

The condition statement compares a value that you specify to aggregate-level field values, such as Line Total or Disc Amount.

The price adjustment applies to the subtotal (Item Total minus Total Item Discount) of the entire quote or order.

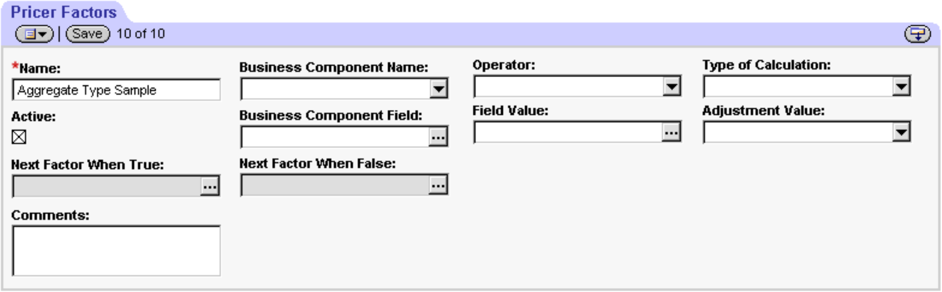
For example:

- If the sum of the Quantity fields of the line items in the quote or order is equal to or greater than 100, apply a specified discount.
- If the total price of the quote or order is greater than or equal to $100,000, apply a specified discount.

NOTE: In addition to this simple comparison of an aggregate-level field with a value, aggregate type pricing factors can compare an aggregate-level field with a field in another business component to determine if the price adjustment should be applied. For more information, see Chapter 14, “Using Mappings in Pricing Factors.”

Fields Used to Define the Aggregate Type Pricing Factor

You define a single type pricing factor using the Pricing Factors form shown in Figure 4.



*Name: Aggregate Type Sample

Active: ☒

Next Factor When True:

Comments:

Business Component Name:

Business Component Field:

Next Factor When False:

Operator:

Field Value:

Type of Calculation:

Adjustment Value:

Figure 4. The Pricer Factors Form for Aggregate Type Pricing Factors

Define the condition using the following fields:

- **Business Components.** Aggregate type factors work only with the following business components:
  - Order Entry - Line Item
  - Quote Item
  - FS Agreement Item
- **Business Components Fields.** Aggregate type factors work with many business component fields. The following are commonly referenced: Line Total, Disc Amount, Discount %, Unit Price, Net Price.
- **Operator.** Specify an operator such as =, <, or >.
- **Field Value.** Specify a value to compare the field’s aggregate value with.

For example, Net Price >1000 is true if the total value in all the Net Price fields of all the items in the quote, order, or, agreement are is greater than 1000.

The price adjustment is performed if the condition is true. Specify the price adjustment using the following fields:

- **Type of Calculation.** Specify what type of adjustment should be done, for example, a % discount, a markup amount, or a price override.
- **Adjustment Value.** Specify the amount of the adjustment.

Defining an Aggregate Type Pricing Factor

This procedure assumes that you are familiar with the overall process for creating and applying a pricing model, described in "The Process of Creating and Applying a Pricing Model" on page 149.

To define a new aggregate type pricing factor

1. From the application-level menu, choose View > Site Map > Pricing Administration >Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model in which you want to create a new aggregate type pricing factor.
3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark in order to lock it.
4. Click the Pricing Factor Designer view tab.
5. In the Pricing Factor Designer list, click New.

6 In the new record in the Pricing Factor Designer list, enter the information described in Table 28 on page 152:

   a In the Type field, select Aggregate.

   b In the Sequence field, enter a number so this aggregate type pricing factor is processed after all single, matrix-based, script-based, or bundling type pricing factors.

7 In the new Pricer Factors form, enter the information described in Table 31.

Table 31. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Name	Replace the system-generated name with a unique, meaningful name.
Business Component Name	Select a Siebel business component containing the data that the pricing factor will compare with the value you enter in the Field Value field to determine whether to apply this pricing factor. The following business components are valid for aggregate type pricing factors: Order Entry - Line Item, Quote Item, FS Agreement Item.
Business Component Field	Select the field in the Siebel business component containing the data that the pricing factor will compare with the value you enter in the Field Value field.
Operator	Select an operator to be used in the comparison. The operators are: =, <>, <, <=, >, >=. When you select one of these arithmetic operators, the Field Value field becomes available.

Table 31. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Field Value	Enter a character string or a number that will be compared to the value in the Siebel BusComp Field. Character strings and numeric constants do not need to be enclosed in quotes (" "). You can enter multiple values in the Field Value field. Click the field to display a dialog box where you can enter a list of values. The pricing engine connects these values with an implied OR. For single pricing factors: If the Pricer Rule Operator is = or < >, all values in the dialog box are considered. If the Pricer Rule Operator is >, > =, <, or < =, only the first value in the dialog box is considered. For aggregate pricing factors only the first value in the dialog box is considered, regardless of the specified operator.
Next When True	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as true. For more information, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."
Next When False	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as false. For more information, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."

Table 31. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Type of Calculation	Select a calculation that will be used to adjust the target price. The options are: ■ Discount Amount decreases the target price by the specified amount. ■ % Discount decreases the target price by the specified percentage value. ■ Markup Amount increases the target price by the specified amount. ■ % Markup increases the target price by the specified percentage value. ■ Price Override replaces the existing current price with the specified price. ■ Power (Target Price, Power) raises the target price to the specified exponential value. For example, if the adjustment value is 2, the target price will be raised to the power of 2, which would yield 4. A target price of $100 with an exponent of 1.176 results in a selling price of $224.9055. ■ Multiplicative Amount multiplies the target price by a specified amount. ■ Round (Current, Decimal Places) sets the number of decimal places to which the price will be rounded.
Adjustment Value	Enter the amount to be used in the calculation to adjust the target price. For example, if the type of calculation is a % Discount and the adjustment value is 10, a 10% discount is applied to the target price. If the type of calculation is Discount Amount and the adjustment value is 10, then a $10 discount is applied to the current price (assuming currency is USD). If the type of calculation is Price Override and the adjustment value is 10, then the price becomes $10 (assuming currency is USD).

Table 31. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Target Price	Select the price that will be used as the basis of the price adjustment calculation. The options are: ■ Base = MSRP. The MSRP in the price list line item detail. ■ Current Price. The price in its current state, which includes all adjustments that ePricer has completed when this pricing factor executes. ■ Base = Cost. The cost in the price list line item detail. ■ Base = List Price. The list price in the price list. If a promotional price is defined for an item, the promotional price is used instead. ■ Base = Purchase. The Purchase Price in the price list line item detail. If the adjustment applies a static amount, the target price is not a part of the adjustment calculation. If the adjustment is a price override, the target price is not relevant.
Active	Click the check box to activate or deactivate the pricing factor. A pricing factor will not execute when it has been deactivated. If it appears in a decision flow path, it will be interpreted as a factor that did not meet the conditions for execution.
Comments	Optionally, enter comments explaining this pricing factor to the end user. The comment entered here appears in the quote's or order's Pricing Comments field.

Bundling Type Pricing Factors

When you define a bundling type pricing factor, you must specify a list of items and the discount that applies if the user selects the required items in the list. For example, you might specify that, if customers buy a dining room table and four chairs, they get 10 percent off on the price of the chairs.

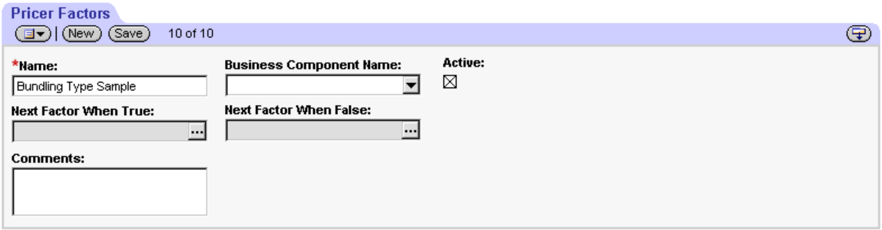
NOTE: Rather than using a pricing factor, you can create a simple product bundle, which allows you to set a price for the entire bundle. For example, you might specify that, if customers buy a dining room table and four chairs, they pay a special discount price for the entire set. For more information about simple bundle pricing, see "Creating a Price List Line Item for a Simple Product Bundle" on page 33.

The bundling type pricing factors makes adjustments based on line item prices, but these adjustments are visible as an adjustment to the total cost of the order. They are not visible to the user in each line item price.

Because you must specify the list of products that is in the bundle, you define bundling type pricing factors differently than other pricing factors. When you choose Bundling in the Type field:

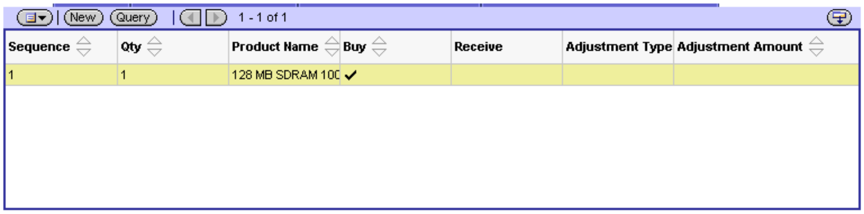
- The Pricer Factor form, shown in Figure 5 on page 187, no longer has the fields used to define the condition and amount of the adjustment.

- The name of the bundling factor in the Pricing Factor Designer list becomes a hyperlink, which you click to display a pricing factor items list, shown in Figure 6, which allows you to add products to this bundle associated with the pricing factor, to specify the price adjustment for each, and to specify whether each is a *Buy* or *Receive* product.



*Name: Bundling Type Sample	Business Component Name:	Active: ☒
Next Factor When True:	Next Factor When False:	
Comments:		

Figure 5. The Pricer Factors Form for Bundling Type Pricing Factors



Sequence	Qty	Product Name	Buy	Receive	Adjustment Type	Adjustment Amount
1	1	128 MB SDRAM 10C	✓			

Figure 6. The Pricing Factor Items List

About the Pricing Factor Items List

When you use the Pricing Factor Items List, you can define price adjustments for both Buy and Receive products.

To qualify for the price adjustment, the user must buy the specified quantity of all the Buy products in the list.

If the user meets the Buy product requirements, then Siebel ePricer applies the price adjustments for all the products in the list, whether they are Buy or Receive. However, that Siebel ePricer applies the price adjustments for Receive products only if they are ordered in the specified quantities.

For example, you can use the Pricing Factor Item list to create bundled discounts such as:

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

The discount is applied to the product that it is for, not to the root product. For example, if the discount is defined to give 50% off on an ergonomic chair, if you buy a desk and chair, then the discount is reflected in the price of the chair.

NOTE: The Buy product quantity is a minimum and the Receive product quantity is a maximum. For example, the pricing factor says that, if you buy 10 desks, you get 50% off on 10 ergonomic chairs for them. Then you must 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.

Defining a Bundling Type Pricing Factor

After you have defined them, you can associate customizable products with any available bundling-type pricing factor, even if that pricing factor was created for a different product and is associated with other products.

This procedure assumes that you are familiar with the overall process for creating and applying a pricing model, described in "The Process of Creating and Applying a Pricing Model" on page 149.

NOTE: Siebel ePricer application does not automatically add, or suggest the addition of, Receive products. You should use the pricing factor Comments field to insert upsell messages. An upsell message may say something like, "If you buy 3 Brand X printers, you can buy 3 printer cartridges for 50% off."

To define a new bundling type pricing factor

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model in which you want to create a new bundling type pricing factor.

3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark in order to lock it.

4. In the Pricing Factor Designer list, click New.

5. In the new record in the Pricing Factor Designer list, enter the information described in Table 28 on page 152:

   a. In the Type field, select Bundling.

   b. In the Sequence field, enter a number so this bundling type pricing factor is processed after all single, matrix-based, or script-based pricing factors, and before all aggregate type pricing factors.

6 In the new Pricer Factors form, enter the information described in Table 32.

Table 32. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Name	Replace the system-generated name with a unique, meaningful name.
Business Component Name	Select a Siebel business component. The following business components are valid for aggregate type pricing factors: Order Entry - Line Item, Quote Item, FS Agreement Item, Customizable Product.
Next When True	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as true. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”
Next When False	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as false. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”
Active	Click the check box to activate or deactivate the pricing factor. A pricing factor will not execute when it has been deactivated. If it appears in a decision flow path, it will be interpreted as a factor that did not meet the conditions for execution.
Comments	Optionally, enter comments explaining this pricing factor to the end user. The comment entered here appears in the quote’s or order’s pricing comments field. This may be an upsell message, such as “If you order Two MegaRam Video Memory cards and one DeathMogul IV CD, the DeathMogul CD is free!”

7 In the Pricing Factors Designer list, click the hyperlinked name of the bundling pricing factor.

A Pricing Factors Item list appears at the bottom of your screen.

8 In the Pricing Factors Item list, click New.

In the new Pricing Factors Item record, enter the information in described in Table 33.

Table 33. Pricing Factor Items Form Fields

Pricing Factor Items Form Field	Entry
Sequence	Enter a sequence number that helps you organize the items in this product bundle. You can sort the items from lowest to highest or highest to lowest.
Quantity	Enter the required quantity of this product for this product bundle. Required quantities of all Buy products must be ordered for the adjustment to be applied. The adjustment is applied to Receive products only if the required quantity is ordered. You cannot set a negative quantity for either a buy or a receive product. If this is a component-based product, then all instances of it must be identical to satisfy the quantity requirement.
Product Name	Click the select button, and select the products for a product bundle.
Buy	Add a check mark it this is a Buy product. The order must contain required quantities of all Buy products for any adjustment to be applied.
Receive	Add a check mark if this is a Receive product. If the order contains required quantities of the Buy products, then adjustments are also applied to the Receive products ordered in the required quantities.
Adjustment Type	Specify the adjustment type for this product, such as % Discount or Markup Amount. For more information, see Chapter 10, "Creating Pricing Models and Pricing Factors."
Adjustment Amount	Specify the adjustment amount for this product. For more information, see Chapter 10, "Creating Pricing Models and Pricing Factors."

9 Continue to add records to the Pricing Factors Item list, until you have added all the products in the bundle.

NOTE: If you are defining a bundling pricing factor for use with a component-based product, you must also reference the data in the Product Designer list, which appears when you click on the hyperlinked name of the bundling pricing factor in the Pricing Factors Designer.

Bundling Type Price Factors for “Buy One, Get One Free” Discounts

Bundling type pricing factors 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.

Bundling type pricing factors 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, do not define the Buy product and the Receive as separate bundle products. Instead, combine them in one Bundle adjustment 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 at 2 and set the receive rule to give 50% off the List price for each product.

Likewise, if customers can buy three dining room chairs and get one chair for free, set the Buy quantity at 4 and set the receive rule to give 25% off the List price for each product.

Bundling Type Pricing Factors for Component-Based Products

There are two ways that you can use a bundling type pricing factor with component-based products:

- Bundling type pricing factors for product components
- Bundling type pricing factors that include component-based products

NOTE: For information about the general constraints on all pricing factors for component-based products, see "About Pricing Models for Component-Based Products" on page 111.

Bundling Type Pricing Factors for Product Components

A bundling type pricing factor may refer to bundles of product components. For example, for customers buying personal computers, you might create a bundling type pricing factor saying that, if they buy both a CD drive and a floppy disk drive, they get a 10% discount on both drives.

Bundling type pricing factors for product components products work in the same general way as other bundling type pricing factors, but there are several added constraints:

- Bundling type pricing factors for products components must include only components of the product. Do not create a pricing factor that mixes components of one product line with other products. Siebel ePricer must process the pricing factor for the component based product first, to determine the price for that price list line item, before it can process pricing factors that contain price list line items.
- Siebel ePricer executes the bundling type pricing factor's price adjustments when a user selects a component that is a Buy product that completes the requirement of the pricing factor. If a Receive product is present in or is added to the component-based product, Siebel ePricer applies the Receive product price adjustment.

- If you are defining the required quantities for a component-based product component in a bundling factor, be aware of the component product constraints in the Products Designer list, especially the Max Cardinality limit. Do not define bundling factor rules that requires the user to buy more products than the cardinality limits; if the pricing rule exceeds cardinality, it will not fire.

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 bundling factor adjustment 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, the highest calculated price for that memory chip will be used as the target price for all bundling factor pricing adjustments on that chip.

Bundling Type Pricing Factors Including Component-Based Products

You can also create bundling type pricing factors that include entire component products. For example, if you buy 10 personal computers and 10 printers, you get a 10% discount on the printers.

In this type of pricing factor, all of the component-based products must be identical to meet the requirement. If there is any difference between instances of a component-based products, each unique product is considered part of a separate line item in the order.

In the preceding example, if a customer bought 10 computers but got CD drives with five and floppy disk drives with five, that customer would not qualify for the discount on the printers.

Script-Based Pricing Factors 13

This chapter covers script-based pricing factors, the most powerful type of pricing factors. It describes these pricing factors and the process of implementing them. It also gives an example of a script.

This chapter covers the following topics:

About Scripts and Script-Based Pricing Factors

You can create pricing factors that invoke a Siebel VB Script or a Siebel eScript. Script-based pricing factors are extremely powerful, limited only by the theoretical limits of the scripts.

Script-based pricing factors are potentially more powerful and more flexible than other factors, because scripts can use data from any available business component and any pricing factor variable that you have defined in a table.

Scripts can also modify business components outside of ePricer. A script may be able to update some target other than a quote, order, or agreement, or component-based product with the output of the pricing engine.

You can use script-based pricing factors for:

- **Complex mathematical calculations.** You can create a script with a complex mathematical calculation such as an optimization algorithm or a differential equation.
- **Multiple business components in a rule.** You can use a script to create a rule that uses multiple business components.
- **Compound rules.** You can use a script to create a compound rule that uses multiple fields in the same business component.

This chapter does not discuss how to write scripts, which requires significant training and is beyond the scope of this book.

For information about Siebel's scripting languages, see *Siebel Tools Online Help*.

For information about setting up Siebel applications, including Siebel ePricer, to use other scripting languages, see the chapter about data mapping using scripts in *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

NOTE: Script-based factors can use scripts written in different languages, including Siebel eScript and Siebel VBScript. Determine which scripting language or languages your system supports before writing a script.

The Process of Implementing Script-Based Pricing Factors

To implement a script-based pricing factor:

1. **"Defining User Properties for the Script" on page 197.** Before you create a script that will be used as a business service, you must define user properties to support attributes and customizable products, which the script will reference.
2. **"Planning and Writing the Script" on page 200.** As mentioned earlier, this chapter does not discuss how to write scripts. However, the section below gives you some necessary details about how scripts interact with the pricing engine.
3. **"Creating the Script-Based Pricing Factor" on page 206.** Use the Pricing Administration screen to define the pricing factor and associate it with the script.
4. **Testing the script-base pricing factor.** Testing and validation tools are discussed in the section about "Troubleshooting and Debugging the Script" on page 203 and in Chapter 16, "Testing and Validating Pricing Models."

NOTE: Pricing factors for component-based products use the same general procedures as pricing factors for other products. However they involve additional constraints. You should not work with them until you have read "About Pricing Models for Component-Based Products" on page 111.

Defining User Properties for the Script

A script-based pricing factor uses a business service or any process that can be called through a business service interface, such as a workflow process. These can be deployed either by being compiled into the SRF or as data in the Siebel database.

The purpose of defining user properties on the business service is to specify the information that is passed by Pricer to the business service.

The purpose of passing only the required information to the business service is to improve the performance of the pricing process by minimizing the amount of data that is transferred.

The format for business services requires you to define user properties to support:

- Attributes
- Customizable products
- Information about the current state of business components, such as the Account for the Quote or the Part # for the Product.

For every business service used as a pricing factor, one user property is required, as shown in Table 34.

Table 34. Property for Business Services Used as Pricing Factor

Name	Value
Use Variable Map	Y

In addition, properties similar to the ones shown in Table 35 can be used to specify information passed to the script.

Table 35. Properties Used to Pass Information to the Script

Name	Value
Field 1	Quote.Name
Field 2	Quote.Currency Code
Field 3	Quote Item.Product
Parameter 4	Quote Item XA.Color

When you are defining the properties used to pass information to the script, follow these recommended naming conventions:

- The properties must be named using sequential integers. These numbers are the keys you will use to pass parameters into the script.
- For the header (such as Quote) and item (such as Quote Item) business components, use a name composed of “Field” followed by an integer.

- For the XA business component (such as Quote Item XA), use a name composed of “Parameter” followed by an integer.
- Specify the values for the header and item using the syntax BusComp.Field, as shown in the examples in Table 35 on page 198.
- Specify the values for the XA buscomp using the syntax XABusComp.AttributeName, as shown in the examples in Table 35 on page 198.

The following properties are always present in the inputs property set to the script factor, supplied by the pricing engine. You should not set user properties for these inputs.

- Purchase Price
- Current Price
- Adjustment Value
- Target Price
- Source Price
- MSRP Price
- Cost
- Base Price
- Reason
- Type of Calculation

For a complete list of user properties, see Appendix A, “Siebel ePricer Deployment and Integration.”

Creating the User Properties

Before you write the script, create a business service and define the user properties that the script will reference.

To define user properties for the script

1. From the application level menu, choose View > Site Map > Business Service Administration > Business Service Details.
2. In the new record in the Business Service list, enter a name for the business service and optionally enter a comment describing the business service.
3. Add new records to the User Properties list, and enter the user properties, as described earlier.

Planning and Writing the Script

As mentioned earlier, this chapter will not include specific instructions for writing the script. For more information about Siebel's scripting languages, see *Siebel Tools Online Help*. Also, look at the sample script at the end of this chapter.

This section will give you basic information about the interaction between the script and the pricing engine.

Passing Information to the Script

For information to be passed to a script, the following conditions must all be met:

- You must have created user properties to specify the information, as described in the previous section.
- The business component field with this information must be active.
- For customizable products, the field must exists in the Customizable Product Header, or Customizable Product business components, as appropriate. In the Tools definition for the fields in these business components, the calculated field check box must be checked, and the calculation field must be empty. Do not specify anything in the Customizable Product XA business component.
- For pricing customizable products during runtime, the integration object used to launch the customizable product session must contain the fields required.

You should verify that these four conditions are met for every item in the script.

Within the script, you get information by referencing the data about user properties that you created for this script. You can reference this data by using a key name starting with *Var*, to retrieve the attribute values that you entered when you created the business service. For example, if you entered the values shown in Table 35 on page 198 when you created the business service, you can use those values in the script by writing the following code:

```
var QuoteName = Inputs.GetProperty ("Var 1");

var QuoteCurCode= Inputs.GetProperty ("Var 2");

var ProductName= Inputs.GetProperty ("Var 3");

var Color= Inputs.GetProperty ("Var 4");
```

The *Var #* syntax is used to represent variables, regardless of whether this item was in the user properties as a field or a parameter.

To avoid problems, observe the following cautions:

- Type carefully. No syntax checking is performed.
- Use the same business component names that you use when creating single pricing factors.
- After writing script factors, do not change the name of your attributes.
- If you specify customizable product user properties for a script that is used from a pricing model at the price list, the script will not receive any information.

Creating the Pricing Logic

Write the script to implement whatever business service functionality is required to accomplish your business process. The logic should determine:

- What the price for this item should be.
- What pricing comments are appropriate for the adjustments made.
- Whether the Pricing Factor Flow should take the Y branch or the N branch after this factor.

Getting Results Out of the Script

To get the results from the script into the Net Price field and the Pricing Comments field of a line item, you set the properties in the Outputs properties set that are shown in Table 36.

Table 36. Output Properties Used to Get Results Out of the Script

Property	Description
Script Applied	If the value is set to True, the pricing adjustment is considered and the flow takes the Y edge. If the value is set to False, any pricing adjustments are disregarded and the flow takes the N edge. The default value is False.
Current Price	The price result of the script factor. This is how you pass back the results of the script. This value is used in the calculation to determine what is entered in the Net Price field.
Reason	This string explains to the user what adjustments were made and why. This field is concatenated for all factors and entered in the Pricing Comments field.

Scripts for Multiple Business Objects

It is possible to write scripts that mix business objects, as long as they do not mix customizable products with other objects. For example, you can write a single script and have it work for Quotes, Orders, and wherever else scripting is used.

To do this, you must specify user properties for each field that data comes from. For example, instead of the user properties shown in Table 35, which all refer to one business object, you could specify user properties similar to those shown in Table 37.

Table 37. Properties Used to Pass Information to the Script

Name	Value
Field 1	Quote.Account Name
Field 2	Order Entry - Orders.Account Name

Whenever the script runs, one of these variables will be null and the other will have a value, depending on which business object the pricing is being invoked from. The logic of the script would have to take this into account and condition the logic to perform operations based only on those parameters that were actually passed.

CAUTION: Do not create scripts that mix customizable products with any other objects. If customizable products and simple products need the same pricing, write two physically separate business services and invoke these services with separate pricing factors.

Legacy Syntax

Do not use the following legacy syntax in scripts.

If you see a user property named __*BusComp Name* __ this is a script with legacy syntax from version 6.x. This syntax will continue to work for version 7.x for simple products only if you carefully followed the recommended algorithm in the 6.x Bookshelf.

CAUTION: This legacy syntax will not work for customizable products in version 7.x. If you use a script with this syntax, and a customizable product is present, Siebel ePricer will return an incorrect price.

Troubleshooting and Debugging the Script

If the script does not work properly, the problem may be caused by one of these common errors:

- Not specifying user properties.
- Attempting to use legacy syntax
- Not setting the *Use Variable Map* = *Y* property.
- Not specifying *Script Applied* = *true* in the Outputs property set.
- Mixing user properties for Customizable Product and any other product.

To help with debugging, you should implement error trapping and logging within the script. If you do not, any errors in the script will result in the script failing, applying no adjustment, with no indication of why this has happened.

Another debugging technique is to use the script debugging functionality of Siebel Tools.

An important debugging technique is to use the property set dumping function shown below. This function writes the full inputs and outputs property sets to a text file so you can see exactly what information is actually going in and out of your script. It creates a log file called PricingScriptFactor.log in the log directory for the application you are running.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)

{

PropertySetToFile(Inputs, "..\\log\\PricingScriptFactor.log", "Property Set with
Variable Map In");

// Pricing functionality goes here

PropertySetToFile(Outputs, "..\\log\\PricingScriptFactor.log", "Property Set with
Variable Map Out");

   return (CancelOperation);

}


function PropertySetToFile (PropSet, fileName, title)

{

var file = Clib.fopen(fileName, "at");


LogData(("\n---------------------------------------------------"), file);

LogData(("Start Process  " + Clib.asctime(Clib.gmtime(Clib.time()))), file);

LogData(title, file);

LogData("PROVIDED PROPERTY SET", file);

WritePropertySet(PropSet, file, 0);

Clib.fclose(file);

return (CancelOperation);
}

function WritePropertySet(PropSet, file, Level)
{
   if ((Level == "") || (typeof(Level) == "undefined")){
          Level = 0;
   }
   var indent = "";
   for (var x = 0; x < Level; x++){
          indent += "\t";
   }
   var psType = PropSet.GetType();
   var psValue = PropSet.GetValue();
   LogData((indent + "Type: " + psType + " Value: " + psValue), file);

   var propName = PropSet.GetFirstProperty();
   while (propName != ""){
      var propValue = PropSet.GetProperty(propName);
      LogData((indent + propName + " = " + propValue), file);
      propName = PropSet.GetNextProperty();
   }
   var children = PropSet.GetChildCount();
   for (var x = 0; x < children; x++){
         LogData(( indent + "CHILD PROPERTY SET " + x), file);
         WritePropertySet(PropSet.GetChild(x), file, (Level + 1));
   }
}

function LogData(DataString, file)
```

```
{
 try {
  Clib.fputs((DataString + "\n"), file);
  Clib.fflush(file);
 }
 catch (e){
   // no action
 }
}
```

Creating the Script-Based Pricing Factor

This procedure assumes that you are familiar with the overall process for creating and applying a pricing model, described in "The Process of Creating and Applying a Pricing Model" on page 149.

After adding the script to the Business Service Scripts list, you can create the script-based pricing factor.

You must use the Name field of the script-based pricing factor to identify the corresponding script. The value in the Name field must be in the following format, ScriptName::ScriptMethod, where the ScriptName is the script name you entered in the Name field in Step 2 on page 200.

NOTE: The script data may override some data entered in the Pricing Factors form, as indicated in Table 38 on page 207.

To create a new pricing factor:

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model in which you want to create a new aggregate type pricing factor.

3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark in order to lock it.
4. Click the Pricing Factor Designer view tab.
5. In the Pricing Factor Designer list, click New.
6. In the new record in the Pricing Factor Designer list, enter the information described in Table 28 in Chapter 10:
   - a. In the Type field, select Script-Based.
   - b. In the Name field, specify the script name using the ScriptName::ScriptMethod format, where the ScriptName and ScriptMethod character strings are exactly as specified in the forms in the Business Service Administration view.
7. In the Pricing Factors form, enter the information described in Table 38.

Table 38. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Name	Replace the system-generated name with a unique, meaningful name.
Target Price	Choose the Target Price from the following: ■ Base = List ■ Base = Purchase ■ Base = Cost ■ Base = MSRP ■ Current Price Can be specified in either the applet or the script: a value passed from the script overrides the value specified in the applet.

Table 38. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Type of Calculation	Select a type of calculation, which is passed to the script as a string using the following codes: ■ "0" Unknown ■ "1" % Discount ■ "2" Markup Amount ■ "3" % Markup ■ "4" Price Override ■ "5" Multiplicative Amount ■ "6" Power (Target, Price, Power) ■ "7" Round (Current, Decimal Places) ■ "8" Discount Amount For example, if you select % Discount as the Type of Calculation, the script receives the string value "1". Can be specified in either the applet or the script: a value passed from the script overrides the value specified in the applet.
Adjustment Value	Specify the adjustment value. Can be specified in either the applet or the script: a value passed from the script overrides the value specified in the applet.
Active	Click the check box to activate or deactivate the pricing factor.
Next Factor When True	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as true. For more information, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."
Next Factor When False	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as false. For more information, see Chapter 15, "Using the Pricing Factor Flow Chart Designer."
Comments	Optionally, enter comments about this pricing factor.

Sample Script

The following sample script is presented as a learning tool. It may not function properly in your environment.

This example shows a very simple pricing script that demonstrates the process of passing information to and from the script.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
   //   User Properties:
   //   Use Variable Map      Y
   //   Field 1               Customizable Product Header.Account
   //   Field 2               Customizable Product.Name
   //   Parameter 3            Customizable Product XA.Color

   var account = Inputs.GetProperty ("Var 1");
   var Product = Inputs.GetProperty ("Var 2");
   var Color = Inputs.GetProperty ("Var 3");
   var CurrentPrice = Inputs.GetProperty ("Current Price");
   var curPrice   = parseFloat (CurrentPrice);
   var reason;

   if (account.indexOf("Siebel") > -1)
   {
      curPrice *=1.2;
      reason = "20% markup for Siebel.  ";
      Outputs.SetProperty ("Script Applied", "true");
   }

   if ((Product == "Shirt") && (Color == "Red"))
```

```
      {
         // note that this is 10 of whatever currency is being used
         curPrice += 10;
         reason += "Surcharge of 10.";
         Outputs.SetProperty ("Script Applied", "true");
      }

      CurrentPrice = curPrice.toString();
      Outputs.SetProperty ("Current Price", CurrentPrice);
      Outputs.SetProperty ("Reason", reason);
      return (CancelOperation);
   }
```

Using Mappings in Pricing Factors 14

This chapter describes how mappings are used in single type pricing factors that refer to values in other business components. It describes search specifications for mappings, and it describes the process of setting up a pricing factor that uses a mapping.

This chapter covers the following topics:

About Mappings in Pricing Factors

A typical single type pricing factor compares some field in the current quote, order, agreement, or component-based product with a value that you specify. For example, a single pricing factor might give the customer a 10% discount if the total price of a line item in a quote is greater than $1,000. These comparisons use arithmetic operators (such as =, >, or <).

Single type pricing factors can also use the operators EXISTS IN or DOES NOT EXIST IN to compare some field in the current quote, order, agreement or component-based product with a field in another business component. When you choose these operators, you must do the mapping to specify the other business component and field.

Because mappings allow you to base price adjustment decisions on data in most Siebel business components, you can create specialized pricing adjustments such as the following:

- Use the Account business component to set prices that vary by account.
- Use the Channel business component to set prices based on channel certification level.
- Use the Territory Assignment business component to set prices based on sales territories.
- Use the Opportunity business component to set prices based on customer relationship details.

NOTE: Only single type pricing factors can use mappings. For more information about single-type pricing factors, see Chapter 10, “Creating Pricing Models and Pricing Factors.”

About the Comparison Used in a Mapping

If you use the EXISTS IN or DOES NOT EXIST IN operator, Siebel ePricer decides whether to apply the price adjustment by seeing whether the value in a field of the quote, order, agreement, or component-based product exists in a field and business component that you specify. You must do the mapping to specify the field and its business component.

To determine which values in this mapped field will be used in the comparison, you specify:

- A business component and field in that business component. The business component and field can be written as <buscomp name>.<field name>. An example is Quote.Account for the Account field in the Quote business component.
- A search specification. The subset of records that results from the search is called the *mapping set*.

At runtime, the pricing factor determines whether the value in a field that you specify in the quote, order, agreement, or component-based product exists in the field that you specified in the mapping set to determine whether to apply a price adjustment.

To see whether the EXISTS IN or DOES NOT EXIST IN condition is satisfied, the pricing factor compares the values that you specify in these two fields of the Pricing Factors form:

- **Business Component Field.** Use the Business Component Field field of the Pricing Factors form to specify the field in the quote, order, agreement, or component-based product.
- **Mapping Field.** Use the Mapping Field field of the Pricing Factors form to specify the field in the mapping set.

The formats of these two comparison fields must be the same. If you compare the same field in two different business components (such as quote.account and partner.account), then the format will be the same. You can compare different fields, only if their values have the same format; for example, you can compare Requested Ship Date with Effective Date if the two use the same date format.

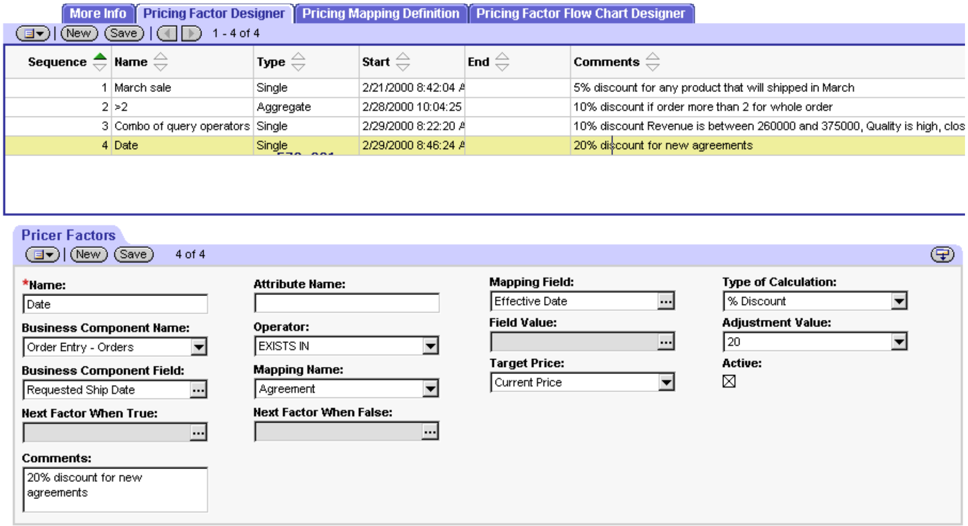
Example: A Mapping Based on the Opportunity Business Component

For example, say you want to give a 10% discount to all accounts in Palo Alto that have an existing opportunity.

The mapping specification creates a mapping set that includes the names of these accounts by searching the Account Name field and the Account Location field in the Opportunity business component of the Opportunity business object. All the records that have Palo Alto as the account location and that exist in the Opportunity business component are included in the mapping set.

Then the pricing factor is defined to compare the Account Name field of a quote with the Account Name field of this mapping set, using the EXISTS IN operator. If the quote has an account name that exists in this mapping set of accounts from Palo Alto with opportunities, the pricing factor gives it a 10% discount.

The Pricer Factor form used to create this mapping is shown in Figure 7.


More Info
Pricing Factor Designer
Pricing Mapping Definition
Pricing Factor Flow Chart Designer
New
Save
1 - 4 of 4

Sequence	Name	Type	Start	End	Comments
1	March sale	Single	2/21/2000 8:42:04 A		5% discount for any product that will shipped in March
2	>2	Aggregate	2/28/2000 10:04:25		10% discount if order more than 2 for whole order
3	Combo of query operators	Single	2/29/2000 8:22:20 A		10% discount Revenue is between 260000 and 375000, Quality is high, clos
4	Date	Single	2/29/2000 8:46:24 A		20% discount for new agreements

*Name: Date

Business Component Name: Order Entry - Orders

Business Component Field: Requested Ship Date

Next Factor When True:

Comments: 20% discount for new agreements

Attribute Name:

Operator: EXISTS IN

Mapping Name: Agreement

Next Factor When False:

Mapping Field: Effective Date

Field Value:

Target Price: Current Price

Type of Calculation: % Discount

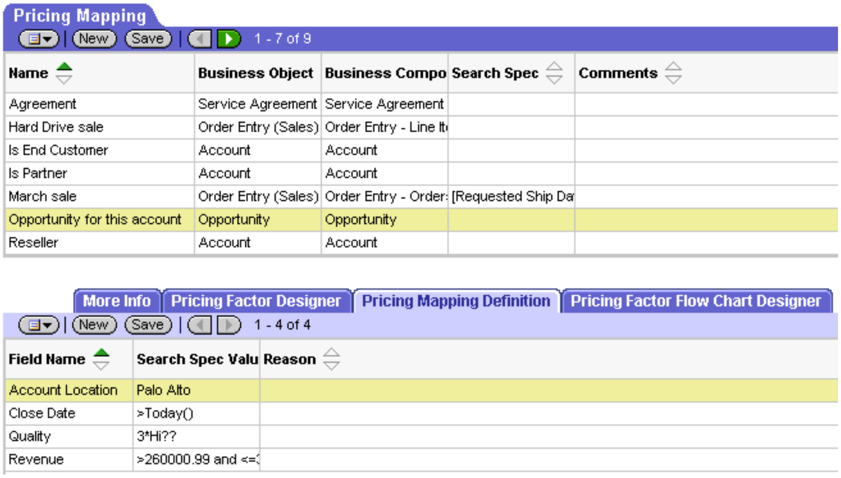
Adjustment Value: 20

Active: ☒

Figure 7. Pricing Factor with Links to Pricing Mapping Data

You can add more complex conditions by using the Pricing Mapping list and Pricing Mapping Definition list. Figure 8 shows a different example, where the mapping creates a mapping set based on records from the Opportunity business component. The search specifications set up in the Pricing Mapping Definition list will search for records with the following characteristics:

- The account is located in Palo Alto
- The close date is in the future
- The opportunity has a high probability of becoming a sale
- The amount of revenue expected from the sale is in the range of $260,000 to $375,000



Name	Business Object	Business Compo	Search Spec	Comments
Agreement	Service Agreement	Service Agreement		
Hard Drive sale	Order Entry (Sales)	Order Entry - Line It		
Is End Customer	Account	Account		
Is Partner	Account	Account		
March sale	Order Entry (Sales)	Order Entry - Order:	[Requested Ship Da	
Opportunity for this account	Opportunity	Opportunity		
Reseller	Account	Account		

More Info
Pricing Factor Designer
Pricing Mapping Definition
Pricing Factor Flow Chart Designer
New
Save
1 - 4 of 4

Field Name	Search Spec Valu	Reason
Account Location	Palo Alto	
Close Date	>Today()	
Quality	3*Hi??	
Revenue	>260000.99 and <=:	

Figure 8. Pricing Mapping Lists

About Search Specifications for Mappings

A mapping search specification filters the records in the business component that you specified to yield the mapping set.

You can define a mapping search specification in two places:

- **Pricing Mapping list.** When you add a mapping record to the Pricing Mapping list, you can enter the search specification using standard Siebel query syntax.
- **Pricing Mapping Definition list.** Alternatively, the Pricing Mapping Definition list gives you a simpler but less powerful way to define search specifications.

If you do not enter a search specification in either of these places, all of the records in the business component are included in the mapping set.

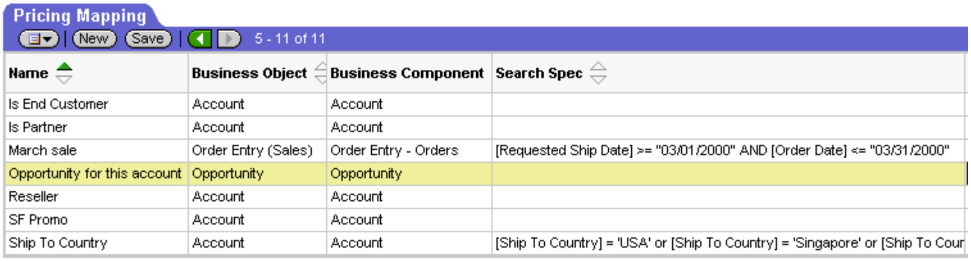
If you enter a search specification in both the Pricing Mapping form and in the Pricing Mapping Definition form, then the criteria in both search specifications are joined with a logical AND operator and executed. If the search specifications are contradictory, the result could be a null set.

You can preview the results of a mapping definition by creating a query in the business component view on which the search specification is based. Use the same logic in the query that you used in the search specification.

NOTE: When you set search specifications on fields containing phone numbers, you must use the Pricer Mapping list (instead of the Pricer Mapping Definition list) if you want the search to return numbers from countries besides the local country. The local country is defined in the Windows Control Panel regional setting. For more information about setting international phone formats, see *Applications Administration Guide*.

About Search Specifications in the Pricing Mapping List

In the Search Spec field in the Pricing Mapping list, you can enter any search specifications that use the standard Siebel Query syntax described in *Siebel Tools Reference*, as shown in Figure 9. You must keystroke the entire search specification; this field does not include drop-down lists.



Name	Business Object	Business Component	Search Spec
Is End Customer	Account	Account	
Is Partner	Account	Account	
March sale	Order Entry (Sales)	Order Entry - Orders	[Requested Ship Date] >= "03/01/2000" AND [Order Date] <= "03/31/2000"
Opportunity for this account	Opportunity	Opportunity	
Reseller	Account	Account	
SF Promo	Account	Account	
Ship To Country	Account	Account	[Ship To Country] = 'USA' or [Ship To Country] = 'Singapore' or [Ship To Cour

Figure 9. The Pricing Mapping List with Search Specifications

The standard syntax rules are summarized below:

- Enclose the field name in square brackets; for example:

  ```
  [Order Type] ='Internal Order'
  ```

- Enter the field name with the same capitalization and punctuation that it has in the repository.
- Enclose the search string in single or double quotation marks. (If it contains an apostrophe, enclose the search string in double quotation marks.)
- Join multiple search criteria with an AND or an OR.
- Repeat the field name when you are comparing a field to multiple values, for example:

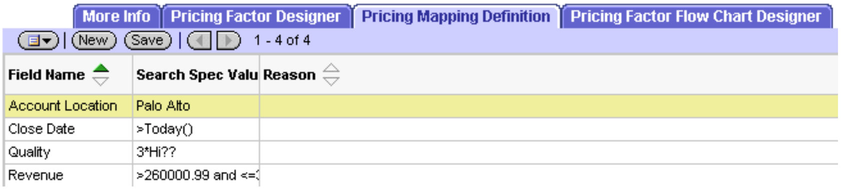
  ```
  [Order Type] ='Internal Order' OR [Order Type] ='Sales Order'
  ```

For more information about search specification syntax, see *Siebel Tools Reference*.

About Search Specifications in the Pricing Mapping Definition List

The Pricing Mapping Definition list, shown in Figure 10, allows you to create search specification by entering criteria on multiple rows:

- In each row, you use a drop-down list to choose a field name, and you enter the rest of the search specification in the Search Spec Value field.
- The rows are all joined with the logical AND operator.


More Info
Pricing Factor Designer
Pricing Mapping Definition
Pricing Factor Flow Chart Designer
New
Save
1 - 4 of 4

Field Name	Search Spec Valu	Reason
Account Location	Palo Alto	
Close Date	>Today()	
Quality	3*Hi??	
Revenue	>260000.99 and <=(	

Figure 10. Pricing Mapping Definition List

The drop-down list of field names includes only the fields in the business component you specified in the Pricing Mapping view, reducing the chance of error. However, if you need to join two search conditions with an OR, you cannot do it here; you must use the Search Spec field in the Pricing Mapping list.

The syntax follows the standard Siebel query syntax. Because the Pricing Mapping Definition list automatically applies some formatting, the following rules also apply:

- Do not enclose a character string in quotes, even if there is a space between two words.
- Use single quotation marks to enclose a value that contains a hyphen, such as an ID value equal to `'1-15VMV'`.

- Enclose a numeric value or a date and time in single or double quotation marks only if there is an embedded space in the value. If you do not use quotation marks, only the numbers to the left of the space will be considered in the search. In the following example, both the date and the time will be considered:

  ```
  ='01/01/00 5:30:00PM'
  ```

  - If you use only the date portion and enclose it in single quotation marks, Siebel ePricer assumes the time is midnight. For example, `='01/01/00'` is interpreted as `='01/01/00 00:00:00'`.
  - If you use only the date portion and do not enclose it in single quotation marks, the results returned will be for the entire day. For example, `01/01/00` returns results for `01/01/00 00:00:00` to `01/01/00 11:59:59 PM`.

The Process of Setting Up a Pricing Factor with a Mapping

To set up a pricing factor with a mapping, you go through the following process:

1. "Verifying That a Field Can Be Used in a Mapping"
2. "Creating a Mapping Record in the Pricing Mapping List" on page 222
3. "Defining the Mapping Search Specification in the Pricing Mapping Definition List" on page 223 (necessary only if you did not enter the search specification in the Pricing Mapping list)
4. "Creating a Pricing Factor That Uses a Mapping" on page 225

After you have defined the pricing factor that uses the pricing mapping, test it and test its interaction with the other pricing factors in the pricing model.

NOTE: The Pricing Mapping Definition list gives you an easier but less powerful way of entering the search specification than the Pricing Mapping list. For more information, see "About Search Specifications for Mappings" on page 216.

Verifying That a Field Can Be Used in a Mapping

A mapping's search specification, in either the Pricing Mapping list or the Pricing Mapping Definition list, refers to a business component field. The business component field must fit the following conditions for the pricing factor to execute successfully:

- The business component must have been defined explicitly in Siebel Tools.
- If the business component contains an entry in the Join field for the extension tables with the business component base table, then certain additional conditions, described in the following procedure, must be met. The business component must contain an entry in the Join field for the extension tables with the business component base table.

To verify that a field can be used in a mapping

1. Open Siebel Tools.
2. In the Object Explorer view, choose Siebel Objects > Business Component.
3. Select the business component for the field to be checked.

   For example, for the List Price field, select the PriceList business component.
4. In the Object Explorer list, choose Siebel Objects > Business Components > Fields.

   The Fields list appears, showing information about the selected business component.
5. Note the table named in the Join field:
   - If the table name in the Join column ends in _X or _XM (such as the Area Code field in the example described in "Example: A Mapping Based on the Opportunity Business Component" on page 214), or if it is not specified in the Join column at all, (such as Currency Code in the example), you can use the business component field in a mapping. In either case, you do not need to complete this procedure.
   - If the Join field contains a table name that does not end in _X or _XM (such as List Price and the S_PRI_LST_ITEM table or Payment Term and the S_PAYMENT_TERM table), complete this procedure before continuing with the mapping definition.
6. In the Object Explorer view, choose Siebel Object > Business Component > Join.
7. Examine the Tables field in the Joins list:
   - If the table noted in Step 5 on page 221 appears in the list, you can use the business component field in the mapping. In the example described in "Example: A Mapping Based on the Opportunity Business Component" on page 214, the table S_PAYMENT_TERM is listed in the Table column and can be used.
   - If the table noted in Step 5 on page 221 does not appear in the list, you cannot use the business component field in the mapping. In the example, the table S_PRI_LST_ITEM is not listed in the Table column and cannot be used.

Creating a Mapping Record in the Pricing Mapping List

After you have planned the mapping and verified that you can use the fields it requires, the first step in creating a pricing factor that uses a mapping is to create the appropriate mapping record in the Pricing Mapping list.

You use this list to specify the business object and business component that the mapped fields are in.

You can also enter the search specification here, or you can enter the search specification in the Pricing Mapping Definition list.

Later, when you create the pricing factor itself, the mapping name you enter here will appear in the Mapping Name drop-down list in the Pricing Factor Designer view. The business object and business component you select here will determine what is displayed in the Business Component field. The business component you select here will also determine which fields you can select in the Field Name field of the Pricing Mapping Definition list.

NOTE: Multiple pricing factors can use the same mapping. If you want to create a pricing factor that uses an existing mapping, you can skip this procedure.

CAUTION: Because many pricing factors can use the same mapping, you should not modify a Pricing Mapping record unless you are sure the modification will work for all the pricing factors that use the mapping. If there is any doubt, you can create a new pricing mapping by copying an existing one and modify the new record as needed.

To create a mapping record in the Pricing Mapping list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.
2. Click the Pricing Mapping Definition view tab.

   The Pricing Mapping Definition list appears, and the Pricing Mapping list appears above it.
3. In the Pricing Mapping list, click New.

4 In the new record in the Pricing Mapping list, enter the information described in Table 39.

Table 39. Fields in the Pricing Mapping List

Field	Description
Name	Required. Enter a unique, meaningful name for the mapping. This name will appear in the Mapping Name field drop-down list in the Pricing Factors Designer view.
Business Object	Required. Select the business object that the mapped fields are in.
Business Component	Required. Select the business component that the mapped fields are in.
Search Spec	Enter a search specification using standard Siebel Query Language described in *Siebel Tools Reference*. For more information about creating search specifications, see "About Search Specifications for Mappings" on page 216. You may enter the search specification either here or in the Pricing Mapping Definitions list. If there is no search specification in either of these places, then all of the records in the specified business component are in the mapping set.
Comments	Optional, but recommended. Enter comments describing the mapping.

Defining the Mapping Search Specification in the Pricing Mapping Definition List

Optionally, if you did not enter the search specification in the Pricing Mapping list, you can define the search specification in the Pricing Mapping Definition list.

The Pricing Mapping Definition list gives you a simpler way of defining complex search specifications.

You can enter each element in the search specification on a separate line. In each line, choose a field, and enter an operator (< , < = , = , > = , > , < >) followed by a string.

The entire search specification is true only if all of the elements are true. The search specification uses an implicit AND operator between the elements.

For more information about using the Pricing Mapping Definition list, see "About Search Specifications in the Pricing Mapping Definition List" on page 218.

NOTE: Before you can use the Pricing Mapping Definition list to create a search specification, you first define the fundamental mapping information—the Name, Business Object, and Business Component fields—in the Pricing Mapping list.

CAUTION: If you enter a search specification in both the Pricing Mapping form and in the Pricing Mapping Definition form, then the criteria in both search specifications are joined with a logical AND operator and executed. If the search specifications are contradictory, the result could be a null set.

To define mapping search specifications in the Pricing Mapping Definition list

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.
2. Click the Pricing Mapping Definition view tab.

   The Pricing Mapping Definition list appears, and the Pricing Mapping list appears above it.
3. In the Pricing Mapping list, select the pricing mapping to which you want to add a search specification criteria.
4. In the Pricing Mapping Definition list, click New.

5. In the new record of the Pricing Mapping Definition list, enter the information described in Table 40 on page 225.
6. If you want to add more rows, repeat Step 4 and Step 5 on page 225 until finished.

Table 40. Fields in the Pricing Mapping Definition List

Field	Description
Field Name	Required. Select a field name.
Search Spec Value	Required. Enter a operator followed by the search string value. Logical operators are: < , < = , = , > = , > , < > .
Reason	Optional. Enter your reason for using this piece of search logic.

Creating a Pricing Factor That Uses a Mapping

After you have created the pricing mapping, you can create a pricing factor that uses it.

To use a pricing factor with a mapping, you must select either EXISTS IN or DOES NOT EXIST IN in the Operator field in a Pricing Factors form. Unless you use one of these as an operator, the mapping fields are inaccessible.

This procedure assumes that you are familiar with the overall process for creating and applying a pricing model, described in "The Process of Creating and Applying a Pricing Model" on page 149.

To create a pricing factor that uses a mapping

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model in which you want to create a new single type pricing factor.
3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark in order to lock it.

4 Click the Pricing Factor Designer view tab.

5 In the Pricing Factor Designer list, click New.

6 In the new record in the Pricing Factor Designer list, enter the information described in Table 28 on page 152. In the Type field, select Single.

7 In the new Pricer Factors form, enter the information described in Table 41.

Table 41. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Name	Replace the system-generated name with a unique, meaningful name.
Business Component Name	Select a Siebel business component containing the data that the pricing factor will compare with the mapped field. The following business components are valid for single type pricing factors: Agreement Item XA, FS Agreement Item, Order Entry - Line Item, Order Entry - Orders, Order Entry XA, Price List, Price List Copy, Price List Item, Quote, Customizable Product Header, Quote Item, Customizable Product, Quote Item XA, Customizable Product XA, Service Agreement.
Business Component Field	Select the field in the Siebel business component containing the data that the pricing factor will compare with the Mapped Field.
Operator	Select an operator to be used in the comparison. To use a mapping, you must select either EXISTS IN or DOES NOT EXIST IN.
Field Value	Do not enter anything here. This field is only used with arithmetic operators.
Mapping Name	Select a mapping from the drop-down list. This list includes the names of mappings that you created in the Pricing Mapping list.
Mapping Field	Select a mapping field from the drop-down list. The pricing factor will compare this field with the Business Component Field.

Table 41. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Next When True	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as true. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”
Next When False	If you are using the Pricing Factor Flow Chart Designer, this read-only field displays the factor to be executed next if this factor evaluates as false. For more information, see Chapter 15, “Using the Pricing Factor Flow Chart Designer.”
Type of Calculation	Select a calculation that will be used to adjust the target price. The options are: ■ Discount Amount decreases the target price by the specified amount. ■ % Discount decreases the target price by the specified percentage value. ■ Markup Amount increases the target price by the specified amount. ■ % Markup increases the target price by the specified percentage value. ■ Price Override replaces the existing current price with the specified price. ■ Power (Target Price, Power) raises the target price to the specified exponential value. For example, if the adjustment value is 2, the target price will be raised to the power of 2, which would yield 4. A target price of $100 with an exponent of 1.176 results in a selling price of $224.9055. ■ Multiplicative Amount multiplies the target price by a specified amount. ■ Round (Current, Decimal Places) sets the number of decimal places to which the price will be rounded.

Table 41. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Adjustment Value	Enter the amount to be used in the calculation to adjust the target price. For example, if the type of calculation is a % Discount and the adjustment value is 10, a 10% discount is applied to the target price. If the type of calculation is Discount Amount and the adjustment value is 10, then a $10 discount is applied to the current price (assuming currency is USD). If the type of calculation is Price Override and the adjustment value is 10, then the price becomes $10 (assuming currency is USD).
Target Price	Select the price that will be used as the basis of the price adjustment calculation. The options are: ■ Base = MSRP. The MSRP in the price list line item detail. ■ Current Price. The price in its current state, which includes all adjustments that ePricer has completed when this pricing factor executes. ■ Base = Cost. The cost in the price list line item detail. ■ Base = List Price. The list price in the price list. If a promotional price is defined for an item, the promotional price is used instead. ■ Base = Purchase. The Purchase Price in the price list line item detail. The adjustment is applied to the target price only if it is a % Markup or a % Discount. If the adjustment is a Markup Amount or a Discount Amount, it applies to the current price. If the adjustment is a price override, both the target price and current price are not relevant.

Table 41. Pricer Factor Form Fields

Pricer Factor Form Fields	Entry
Active	Click the check box to activate or deactivate the pricing factor. A pricing factor will not execute when it has been deactivated. If it appears in a decision flow path, it will be interpreted as a factor that did not meet the conditions for execution.
Comments	Optionally, enter comments explaining this pricing factor to the end user. The comment entered here appears in the quote's or order's Pricing Comments field.

Using the Pricing Factor Flow Chart Designer 15

This chapter describes how to use the Pricing Factor Flow Chart Designer to control the decision flow logic for pricing models that use multiple pricing factors. It includes information about designing decision flow logic, instructions about how to use the Pricing Factor Flow Chart Designer, and hints for debugging decision flow logic.

This chapter covers the following topics:

- "About the Pricing Factor Flow Chart Designer" on page 232
- "Using the Pricing Factor Flow Chart Designer" on page 239
- "Debugging Decision Flow Logic" on page 242

About the Pricing Factor Flow Chart Designer

When you define a pricing model, each factor has a sequence number that defines the default order in which the factor is executed within the model.

The Pricing Factor Flow Chart Designer allows you to redefine the sequence in which pricing factors are executed using branching logic, also known as decision flow logic. This logic uses the outcome of a pricing factor to determines which pricing factor is executed next.

The Pricing Factor Flow Chart Designer allows you to define the decision flow logic using an interactive graphical user interface, where you drag and drop elements to create the flow chart.

The decision flow logic determines how pricing factors should be sequenced, when factors should be skipped, and when to exit a pricing model. It can also execute different pricing factors in different sequences, depending on the data in a quote, order or agreement.

If you do not use the Pricing Factor Flow Chart Designer, Siebel ePricer processes the pricing factors in the order specified by the numbers in their Sequence field. In many pricing models, this numerically ordered sequence is all you need to manage the processing order of the pricing factors. For more information about pricing factor sequence numbers, see “Creating Pricing Factors” on page 151.

NOTE: When discussing the Pricing Factor Flow Chart Designer, the term *single factor* is sometimes used to refer to any factor that applies to single line items; this includes single, matrix-based, and script-based type pricing factors. The term *aggregate factor* is sometimes used to refer to any factor that applies to multiple line items; this includes aggregate and bundling type pricing factors.

Strategies for Design of Pricing Factors and Decision Flow Logic

You can switch between the Pricing Factor Designer and the Pricing Factor Flow Chart Designer:

- When you create a pricing factor in the Pricing Factor Designer, it appears in the Pricing Factor Flow Chart Designer, so you can incorporate it into the decision flow.
- You can double-click on the icon for a pricing factor in the Pricing Factor Flow Chart Designer to display the Pricing Factor Designer and Pricing Factor form for that pricing factor, so you can modify it.

This connection between the Pricing Factor Designer and the Pricing Factor Flow Chart Designer allows you to use either a top-down or bottom-up approach to the developing pricing factors and pricing model decision flows.

- **The top-down approach.** Begin with a general view of the pricing factors and decision flow logic in a pricing model, and then work out the details of each pricing factor. To use this approach, begin by working in the Pricing Factor Flow Chart Designer, which allows you to create pricing factors and put them in the proper place in the logical path. Use these simple pricing factors as placeholders as you work on the decision flow logic model, and return to them later to design them in detail.
- **The bottom-up approach.** Begin by creating individual pricing factors, and then work out their decision flow logic. For example, if you are creating a complex pricing factors, you may want to work out all their details before you fit them into the decision flow logic. To use this approach, begin by working in the Pricing Factor Designer. After you have defined pricing factors, icons representing them appear in the Pricing Factor Flow Chart Designer. You can drag it into the proper position in the logical path of the decision flow, and integrate a new or modified pricing factor into the pricing model by rearranging the connectors to and from the factor.

As a general rule, a bottom-up approach is more suitable if you are new to Siebel ePricer.

Whichever approach you use, you should define single pricing factors first (Single, Matrix-based, or Script-based type factors), and then arrange them into the single factor decision flow path (the path that begins with the Start symbol). After testing and validating this decision flow path, develop Bundling and Aggregate type factors, and arrange them in the aggregate factor decision flow path (the path that begins with the Aggregate Start symbol).

Before you work on the aggregate factor decision flow, the single factor decision flow logic should be stabilized, because the results of the single factor decision flow determine the starting prices for the aggregate factor decision flow.

About the Design of Decision Flow Logic

The Pricing Factor Flow Chart Designer provides a graphical user interface (GUI) that allows you to:

- **Define the decision flow logic for a pricing model.** You can control the order in which pricing factors are executed by manipulating icons and arrows that represent pricing factors and decision flow path connections.
- **Create pricing factors by dragging and dropping a Factor icon into the decision flow design area.** You can double-click these icons to display a pricing factor in the Pricing Factor Designer and define it in detail.

The use of all the Pricing Factor Flow Chart Designer Palette icons is summarized in Table 42.

Table 42. Pricing Factor Flow Chart Designer Palette Icons

Palette Icon/Symbol	Function/Usage
Factor	To create a new, single type pricing factor with a system-generated default name, drag and drop the Factor icon from the palette to the design area. The factor symbol appears, displaying the default name. To open the Pricing Factor record associated with a factor symbol, double-click on the symbol. When you create a new factor symbol, use the Pricing Factor record to specify at least its Name and Type. To resize a factor symbol, click and drag the handles at its edges. To move a factor symbol, click and drag the symbol (avoiding the handles).
Start	To create a decision flow for pricing factors of the single, script-based, or matrix type, begin by dragging and dropping the Start icon from the palette to the design area.
Aggregate Start	To create a decision flow for pricing factors of the bundling or aggregate type, begin by dragging and dropping the Aggregate Start icon from the palette to the design area.
Exit	To create a complete decision flow, you must create a logic path linking either start symbol to the one exit symbol. The path may branch, but all branches must all ultimately lead to the exit symbol. If an exit symbol does not exist in the decision flow design area, drag and drop the Exit icon to the design area to create one.
Connector	To create a connector, drag and drop the Connector icon from the palette to the design area, making sure to connect the left end of the line to a connector point (x) on a factor or start symbol. The connector symbol appears, displaying the Y label if it is the first connector attached to a factor, or the N label if it is the second. If the conditions for applying the pricing factor are met, Siebel ePricer applies the pricing factor's pricing adjustment, and then follows the Y connector to the next factor. If the conditions for firing a pricing factor are not met, Siebel ePricer follows the N connector to the next factor.

A complete decision flow path must include:

- A Start or Aggregate Start symbol connected with a Y connector to a factor symbol
- Both a Y and an N connector linking each factor symbol in the decision flow logic path to the next factor, or to the exit symbol
- One decision flow logic path that is valid for any value that the user might enter at runtime

Both the Start and Aggregate Start decision flow path should lead to one Exit symbol.

The following rules govern the operation of decision flow logic, for both complete and incomplete decision flow paths:

- If there is a Start symbol in a pricing model, the application starts from there. If there is no Start or Aggregate start symbol, the application starts with the pricing factor with the lowest sequence number.
- If a decision flow that begins at the Start symbol ends at the exit symbol, then the decision flow logic restarts at the Aggregate Start symbol (if an aggregate start decision flow path has been defined for the model).
- When Siebel ePricer processes a pricing factor:
  - If the conditions for applying the pricing factor are met, Siebel ePricer applies the pricing factor’s pricing adjustment, and then goes to the pricing factor identified by the Y connector in the decision flow logic, known as the *next when true* factor.
  - If the conditions for firing a pricing factor are not met, Siebel ePricer goes to the pricing factor identified by the N connector, known as the *next when false* factor.
- Siebel ePricer continues to follow the decision flow sequence until its logic fails or it reaches an exit.
- If a needed connector is missing, then the model stops processing pricing factors. This occurs only if the connector is needed in the session: if one connector leading from a factor is missing, but the outcome of the factor does not require that connector, then Siebel ePricer continues by using the other path.

- If an error in the definition of the decision flow results in a logical infinite loop, ePricer performs a predefined number of iterations before ending the decision flow process. This number is defined the user property of the Pricing Manager business service that is named "Max number of execution allowed." For information about working with user properties, see *Siebel Tools Reference*.
- If the exit path is missing, then Siebel ePricer fires the pricing factor with the next higher sequence number, which may lead to unexpected results.

When you use the decision flow connectors, their logic is displayed in the Pricing Factor Designer form: the Next When True field contains the pricing factor that the Y connector points to, and the Next When False fields represents the pricing factor that the N connector points to.

Example: Decision Flow Logic for a Compound Pricing Factor

Suppose you want to set up a sequence of pricing factors that applies a line item discount to:

- Orders for certain accounts for a particular product (a "CPU Pentium 1000")
- Orders made within a special promotional period.

Customers can get one of these discounts, but not both.

You might set up four price factors with the following logical sequence:

(1) IF Order Entry - Orders.Account EXISTS IN Account.Account.Name, Search Spec: [Type] = 'Customer'

AND

(2) Order Entry - Line Items.Product = 'CPU Pentium 1000'

THEN Price Adjustment = 1.33333% Discount

OR

(3) Order Entry - Orders.Order Date > '12/02/03'

AND

(4) Order Entry - Orders.Order Date < '12/25/03'

THEN Price Adjustment = 1.33333% Discount

Figure 11 shows how you might diagram the decision flow logic for this example. If pricing factors 1 AND 2 are both true, than Siebel ePricer gives the discount and exits from the decision flow. If one of them is untrue, it goes to pricing factors 3 AND 4, and if they are both true, it gives the discount. Notice that the sequence has a start and an end, and that the end covers all possible cases.

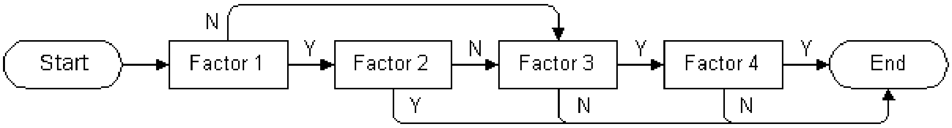


Figure 11. Pricing Factor Sequence Logic: (1 AND 2) OR (3 AND 4)

Using the Pricing Factor Flow Chart Designer

You can define one decision flow logic path for single pricing factors beginning at the Start symbol, or one for aggregate pricing factors beginning at the Aggregate Start symbol, or one of each:

- Beginning at the Start symbol, you may add single type, matrix-based type, and script-based type pricing factors.
- Beginning at the Aggregate Start symbol, you may include bundling type and aggregate type pricing factors.

These two must be defined separately. When the pricing model runs, logic beginning at the Start symbol is executed first, and the logic beginning at the Aggregate Start symbol is executed afterwards.

For more information about using the Pricing Factor Flow Chart Designer, see "About the Pricing Factor Flow Chart Designer" on page 232.

The following instructions describe how to create one decision flow. You may also create two decision flows, one beginning with Start and one beginning with Aggregate Start.

CAUTION: Connectors in the decision flow path beginning at the Start symbol must never lead to any factor in the flow path beginning at the Aggregate Start symbol, except the exit factor. Connectors in the decision flow path beginning at the Aggregate Start symbol must never lead to any factor in the flow path beginning at the Start symbol, except the exit factor.

The following general instructions about using the Pricing Factor Flow Chart Designer are valid for all pricing models.

To work with decision flow logic in the Pricing Factor Flow Chart Designer

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the Pricing Model for which you want to define a pricing factor decision flow.

3. If the Locked field of this pricing model record does not already have a check mark in it, click it to add the check mark to lock it.

4. Click the Pricing Factor Flow Chart Designer view tab.

   The Pricing Factor Flow Chart Designer appears, with a palette area and a decision flow design area.

   NOTE: Some pricing factors may be stacked on top of each other in the upper left corner of the design area. A pricing factor is located here by default if you create it with the Pricing Factor Designer.

5. Drag and drop a Start symbol or an Aggregate Start symbol into the upper left corner of the decision flow design area.

6. Drag and drop the Exit icon into the lower-right corner of the decision flow design area.

7. Create pricing factors, as necessary, from within the Pricing Factor Flow Chart Designer.

   a Drag and drop the factor icon into the design area to create an new pricing factor record. A system-generated default name appears on the factor.

   b Double-click on the new factor symbol to open the corresponding new record in the Pricing Factor Designer list.

   c In the new record in the Pricing Factor Designer list, enter at least the Name and Type of the Pricing Factor to identify it. You can complete its definition later.

   d Reopen the Pricing Factor Flow Chart Designer.

   The new name appears on the factor symbol.

8 Add a connector from the Start symbol to the first pricing factor to be executed.

- **a** From the palette area, drag and drop the connector icon so that its left edge touches a connection point on the right side of the start symbol.

  If a connector is properly attached to a symbol, a red dot appears at the connection point.

- **b** Click the connector to make a handle appear at the arrowhead of the connector, then drag this handle to a connector point on the first factor to be executed in the decision flow path.

9 Attach connector symbols to each factor to link it to one or two other factors or to the Exit symbol.

- **a** The first connector arrowhead that you attach to a factor is automatically labeled Y (Yes). Attach its arrowhead to the next factor to be fired if it fires.

- **b** The second connector arrowhead that you attach to a factor is automatically labeled N (No). Attach its arrowhead to the next factor to be fired if it does not fire.

NOTE: Both the Y and the N connector can point to the same next factor.

10 Adjust a connector if necessary by dragging it or by right-clicking it and using the menu to modify the connector line.

- **a** Make sure that each connector is linked properly to a factor connection point at both ends and that you can clearly distinguish each Y connector and N connector.

- **b** To divide a connector line into multiple segments and change its shape:

  - ❑ Right-click the connector and choose Edit > Add Point to add a handle to the connector line, which serves as a hinge or pivot point in the line.

  - ❑ Drag and drop the new handle to a new position in the design area.

  - ❑ You can repeat this process to divide a connector into multiple line segments.

Debugging Decision Flow Logic

There may be errors in the design of a decision flow for several reasons:

- Human error may result in a decision flow that does not accomplish what you intended.
- If all paths do not lead to the Exit symbol, then Siebel ePricer fires the pricing factor with the next higher sequence number, which may or may not be part of the decision flow.
- If the decision flow includes a logical loop, ePricer will perform a defined number of iterations and then automatically ending the decision flow.

You must test the decision flow to avoid these problems.

Perform incremental testing to identify problems in individual pricing factors before you test an entire decision flow path. Isolate specific sections of each start or aggregate start decision flow path, and test the behavior of short sequences of pricing factors within these decision flow paths.

Both the Active flag and the sequence numbers are useful for testing the factors without using the decision flow, so you can make sure the individual factors are working before you introduce the added complexity of the decision flow:

- In addition to the decision flow, assign sequence numbers that establish sensible default logic for the firing sequence. Then, you can use this simple default sequence for initial testing and debugging.
- You can use the Active flag to turn a pricing factor on and off, which allows you to test individual factors and selected combinations of factors. (If a pricing factor that appears in a decision flow is inactive, the decision flow logic will treat that factor as if the conditions for execution were not met, and the control will follow the path marked N.)

To test the decision flow, you can create new pricing factors that serve as tracers, which you insert into the decision flow at key decision points within the decision flow logic. You can also modify an existing pricing factor to serve as a tracer. Create tracers with exaggerated price adjustments, to make it obvious when these pricing factors are firing. By inserting a tracer in different places, you can identify where the decision flow stopped working as you anticipated.

NOTE: If you create tracers, be sure to remove them before releasing the model. A naming convention that identifies pricing factors as tracers will help in both testing and clean-up.

Testing and Validating Pricing Models 16

This chapter includes information about using the Pricing Model Validation views to test and validate pricing models. It also has hints about how to troubleshoot testing problems.

This chapter covers the following topics:

About Testing and Validating Pricing Models

To validate pricing models, you can test them as they will be used in actual runtime scenarios.

To validate that a pricing model functions as expected, you need to test each pricing factor and the sequences of pricing factors that may occur when the model is actually used.

It is best to test incrementally. Test the results of each pricing adjustment before testing the interactions of adjustments. This is especially important if you are applying many pricing factors in the pricing model or compounding multiple pricing factors in a model.

Siebel ePricer has built-in features for testing pricing factors.

- The Quote Pricing Model Validation view and the Order Pricing Model Validation view allow you to create quote and order data and to experiment with sample scenarios.
- The Siebel ePricer log file records the effects of each pricing factor when it is executed.

When you test pricing factors in the pricing model validation views, the pricing model must be locked. A locked model cannot accidentally execute in runtime.

If you test the pricing factors in a simulated runtime environment, the pricing model must be unlocked. An unlocked model can be used in runtime processing.

To test a model in any environment, you must use the Reload function to load the model cache.

It is particularly important to test models when pricing models use complex decision flows, with different pricing adjustments, such as volume discounts, component-based pricing, attribute-based pricing, and pricing factors.

The Process of Testing Pricing Models

To test pricing models, you go through the following process:

- **"Reloading the Pricing Model for Testing Purposes" on page 247.** Pricing models are cached, and you must be sure the latest version of the model is in the cache before you test it.
- **"Using the Pricing Model Validation Views" on page 249.** You can use the Quote Pricing Model Validation view or the Order Pricing Model Validation view to create quote and order data and to test pricing models and factors with sample scenarios.
- **"Examining the Pricing Engine Log File" on page 251.** When you use these views for model validation, Siebel ePricer generates a log file named Pricing_Engine_Log.txt, which allows you to determine exactly how ePricer processed pricing adjustments within a given pricing session with a quote or order.

NOTE: If you establish a test environment (for example, a separate server and database) for simulating the effects of pricing models in a live environment, then you may test pricing models using the standard quote and order views, rather than using these Model Validation views. The Model Validation views allow you to test different pricing factors and pricing factor combinations without affecting the production version of the pricing model.

Reloading the Pricing Model for Testing Purposes

To enhance performance, pricing models are cached when they are initially executed by a quote or order or initially unlocked.

After the cache has been created once, you must reload the model to cache any changes. You can do this at any time, as mentioned above, because the caching process can take place whether the pricing model is locked or unlocked.

To test a pricing model, you must save it and reload it into the cache, so you have the latest of that model in the cache.

Always make a backup copy of a model before updating it.

The safest practice is to use a test database for changing or updating the model and validating the model. If it is not possible to use a test database, use a dedicated application server for testing. This minimizes any chance of changing a cache that might be accessed by runtime users.

After you complete the validation of the model, you must unlock the model and reload it to make sure that the model is updated in the cache in the application server.

CAUTION: When you load a new version of a pricing model into the cache, that version is used by the current application server. If you load a testing model into the cache on the server used for live runtime users, you could overwrite the pricing model cache used by runtime users, replacing it with the test model. Consult with your Server Administrator to develop procedures that allow you to cache models safely during testing.

To reload a pricing model for testing

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager.

   The Pricing Model Manager list appears.

2. In the Pricing Model Manager list, select the pricing model you want to reload.
3. Lock the pricing model, if it is not already locked.

   The check box in the Locked column should contain a check mark.

4. Be sure the Pricing Model to reload is selected. It should be the current record.
5. In the Pricing Model Manager list, click the menu button, and then Reload.

Using the Pricing Model Validation Views

The ePricer application provides two pricing model validation views: the Quote Pricing Model Validation view and the Order Pricing Model Validation view.

These views work in a similar manner, although they use slightly different sets of Siebel business components. Which one you select may depend on the business components used in a model's pricing factors. For example, if you need to test a pricing factor that uses a Order business component (that is, any business component that appears only in an order), then you must use the Order Pricing Model Validation View.

When you use these views, you generally should:

- **Lock the pricing model.** You must lock a pricing model in order to test it in a pricing model validation view.
- **Activate and deactivate pricing factors.** Activate the pricing factors you want to test, and deactivate those you do not want to test.

To test this...	Do this
Single factor	Activate the factor to be tested and deactivate those not to be tested.
Multiple factors	Activate the factors whose interactions you are testing and deactivate all other factors.
All factors in the model	Activate all factors in the model.

NOTE: If you make a pricing factor inactive, ePricer decision flow logic interprets this pricing factor as if it did not fire because the test conditions were not met. The decision flow logic follows the connector marked N leading from the inactive factor to determine the next factor to test in the decision flow.

- **Reload the model into the cache.** You must reload the model in the cache, so the test runs only the factors that you activated.

CAUTION: To avoid overwriting the cache used to serve runtime users with a test model, follow the advice in "Reloading the Pricing Model for Testing Purposes" on page 247.

- **Create test quotes or orders.** In the Quote Pricing Model Validation view or the Order Pricing Model Validation view, use the Test Quotes or Test Orders forms to enter sample data to test a pricing factor.
- **Test the quote or order.** Use the Test Quotes Line Items or Test Orders Line Items forms to create data and execute the quote or order validation.
- **Check the pricing.** After the pricing factor executes, make sure the pricing is correct by checking the following fields in the Test Quotes Line Items or Test Orders Line Items list and in the associated Totals form:
  - Start Price field displays the list price or the promotional price from the price list. The promotional price appears if one has been defined for the item; otherwise, the list price appears.
  - Net Price field is the price after all adjustments are applied at the line item level. Volume discounts are reflected in the net price.
  - Pricing Comments field in the Line Items list displays the comment field from the pricing factor. If multiple pricing factors execute in a test, multiple comments are displayed in this field, separated by commas. This field is blank if a pricing model was not used to calculate the selling price or if the Comment field in the pricing factor was left blank.
  - Discount Comments/Reasons field in the Totals or SubTotals form displays the pricing factor Comment associated with the aggregate pricing factor. If no aggregate pricing factors executed, this field is blank.

To test pricing factors using the Pricing Model Validation views

1. From the application-level menu, choose View > Site Map > Pricing Administration > Pricing Model Manager > Pricing Model Manager.
2. Make sure the pricing model that contains the pricing factors to be tested is locked. If it is not, lock it by clicking its Locked check box.
3. Click the Pricing Factor Designer view tab and, in the Pricing Factor designer list, use the Active check box to activate the pricing factor or factors to be tested and deactivate those that are not to be tested.
4. Click the More Info view tab.

5. In the Pricing Model Manager list, click the menu button, and then click Reload.

   The current version of the selected Pricing Model is reloaded into the cache, overwriting any existing cache for the model. To avoid overwriting the cache used to serve runtime users with a test model, follow the advice in "Reloading the Pricing Model for Testing Purposes" on page 247.

6. From the Show drop-down menu, select either Quote Pricing Model Validation or Order Pricing Model Validation.

7. Enter test data for a quote or an order.

8. In the Test Quotes Lines Items view tab, click Reprice (or, if you are testing an aggregate type pricing factor, click Reprice All).

9. Review the pricing results to verify that the pricing factor was applied properly.

Examining the Pricing Engine Log File

If you want more details about how the pricing adjustments were applied in this test, check the Pricing Engine Log file.

When you run tests using the Quote Pricing Model Validation or Order Pricing Model Validation, Siebel ePricer will create a log file whenever you process new products or click the Reprice or Reprice All button.

This file is located in the Siebel server installation directory in a TEMP file titled Pricing_Engine_Log_<username>.txt, where the username is your login name.

This log file is a simple text file to which records are appended each time the Siebel ePricer engine attempts to execute a pricing adjustment. It provides information about how ePricer executed or attempted to execute pricing adjustments.

Make arrangements with your Siebel Systems Administrator so that you have immediate read and write access to this file, which is crucial for testing and validation purposes.

Save a copy of the log file under another name to preserve the information in it. Give the copy a name that identifies the session, and add text to the beginning of the copied file to define the testing parameters, if appropriate.

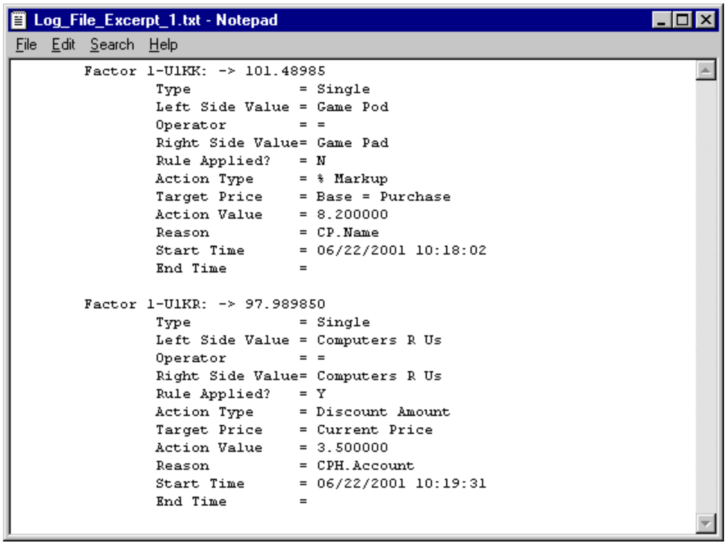
After you have saved a copy of it, delete the original log file.

If you continue the testing session without renaming the log file and deleting the original log file, Siebel ePricer will continue to append data to the original log file, which can make it too large to be useful for troubleshooting. If you rename it, Siebel ePricer will create a new Pricing Engine Log file the next time you attempt to execute a pricing factor. By renaming the log file, you can create a separate log file for each test.

If you exit the testing session, and then begin testing again, the log file for the new test data will overwrite the existing log file. You can preserve the log file by renaming it as soon as each defined test is completed.

Example: Sample Results in the Pricing Engine Log File

You can see an example of a Pricing Engine Log File in Figure 12. The current price at the beginning of each entry shows the result of a price adjustment, and the details below it describe that price adjustment.



```
        Factor 1-U1KK: -> 101.48985
                Type            = Single
                Left Side Value = Game Pod
                Operator        = =
                Right Side Value= Game Pad
                Rule Applied?   = N
                Action Type     = % Markup
                Target Price    = Base = Purchase
                Action Value    = 8.200000
                Reason          = CP.Name
                Start Time      = 06/22/2001 10:18:02
                End Time        =

        Factor 1-U1KR: -> 97.989850
                Type            = Single
                Left Side Value = Computers R Us
                Operator        = =
                Right Side Value= Computers R Us
                Rule Applied?   = Y
                Action Type     = Discount Amount
                Target Price    = Current Price
                Action Value    = 3.500000
                Reason          = CPH.Account
                Start Time      = 06/22/2001 10:19:31
                End Time        =
```

Figure 12. Using the Log File to Detect a Pricing Factor Error

In the first entry in this sample, there was an error. The name of a product, Game Pod, was incorrectly entered as Game Pad, so a match did not occur and the pricing adjustment was not applied. Notice that Game Pod appears in the Left Side Value field and Game Pad appears in the Right Side Value field. The row that says Rule Applied? = N indicates that the pricing adjustment was not applied.

In the second entry in this sample, there was not an error. The Left Side Value and the Right Side Value are the same: both are Computers R Us. The row that says Rule Applied? = Y tells us that the pricing adjustment was applied, and the pricing factor executes. The Action Type is Discount Amount, and the Action Value is 3.500000, so we know that a $3.50 discount was applied. The Reason field says CP.Name, which means that a customizable product name field was used to set up the condition for firing this factor.

All the fields in the Pricing Engine Log file are explained in "Fields in the Pricing Engine Log File" on page 254.

Typical Problems in the Pricing Engine Log File

Using the Pricing Engine Log file, you may be able to identify the typical problems described in Table 43.

Table 43. Identifying Problems Using the Pricing Engine Log

If you see this...	Check this
Only the Product ID lines are displayed for a log entry.	■ Verify that the date of the test is within the range specified in the pricing factor. ■ Verify that the pricing factor is active. ■ Verify that the pricing model has been associated with a price list. ■ Exit and reenter the application to verify that the price list has been refreshed.
Left Side Value is blank.	Verify that the specified business component field is active.

Table 43. Identifying Problems Using the Pricing Engine Log

If you see this...	Check this
Rule Applied equals N and Right Side Value is blank.	It is probable that fields with unlike data types are being compared. Check the data types of both business components fields in Siebel Tools. If the data types are different, change the pricing factor business component fields.
Numbers after the Factor ID are the same.	Check Rule Applied: ■ If it equals N, the pricing factor did not execute. ■ If it equals Y, the pricing factor is setting the selling price to the target price.

Fields in the Pricing Engine Log File

Table 44 describes the fields in the Pricing Engine Log.

Table 44. Fields in the Pricing Engine Log

Field	Description
Product ID	Displays the database ID of the product in the quote or order.
Price List ID	Displays the price list used by the pricing model.
List Price	Displays the list price of the product from the Price List Line Items view.
Promotional Price	Displays the promotional price of the item from the Price List Line Items view. If the product has a Promotional Price, it is the basis for pricing calculations, not List Price.
Current Price	Displays the current price of the item, which reflects all pricing adjustments that have been applied before to this point.
Quote ID and Quote Item ID (or Order ID and Order Item ID)	Identifies the quote or order that is being reported.
Volume Discount fields	Show if a volume discount was applied to the product.

Table 44. Fields in the Pricing Engine Log

Field	Description
Factor ID	Identifies the pricing factor that was executed. The number following this ID is the current price after this pricing adjustment was executed.
Type	Identifies the type of pricing factor - single, matrix-based, script-based, bundling, or aggregate. Use this field to verify that the pricing factor is being applied as intended.
Left Side Value	Identifies the value that comes from the Quote or Order. If this field is blank, it typically means that the business component field is inactive.
Operator	Identifies the comparison operator in the pricing factor.
Right Side Value	Identifies the value that comes from the field value or mapping. For some types of adjustments, individual values are not listed: ■ If this field contains [..], it means that the search specification in a mapping returned values. ■ If the pricing factor is script-based, the pricing factor name appears here. ■ If the pricing factor is bundling type, the Buy product requirements are referenced here. Use this value to spot character string errors in a field value or mapping, such as the one in Figure 12 on page 252.
Rule Applied?	Indicates whether the price was adjusted. If the value in this field is N, the pricing factor was not applied. If it is Y, then the factor was applied.
Action Type	Displays the type of calculation in the pricing factor.
Target Price	Identifies the price-basis in the pricing factor as one of the following: List price, Current price, MSRP, Purchase Price or Cost.
Action Value	Displays the value of the Adjustment Amount field in the pricing factor.
Reason	Displays the text of the Comments field in the pricing factor.

Table 44. Fields in the Pricing Engine Log

Field	Description
Start Time	Displays the date and time at which the factor will be activated.
End Time	Displays the date and time at which the factor will be deactivated.

Troubleshooting Testing Problems

Table 45 lists solutions to some common problems that may occur when you use the Model Validation views to test pricing adjustments.

Table 45. Troubleshooting Testing Problems

If this happens...	Try this
Cannot update a field in a factor.	Verify that the model is locked.
New factors are not executing	Reload the model.
The validation view results indicate that a pricing adjustment has not executed.	■ Exit and reenter the application, and test again. ■ Verify that the pricing factors that you are testing have been activated. ■ In the price list you are using for the test, verify that the volume pricing and attribute pricing fields contain the correct data. ■ Verify that the pricing model is linked to the correct price list or component-based product. If it is not, link it, and then exit and reenter the application before retesting. ■ Verify that the date of the test is within the range defined by the effective dates of the price list. ■ Verify that the date of the test is within the range defined by the Start Date and End Date of the pricing model. ■ Verify that the date of the test is within the range defined by the Start Date and End Date of pricing factors you are testing. ■ Check the Pricing Engine Log file to see if the pricing factors executed.

Table 45. Troubleshooting Testing Problems

If this happens...	Try this
Test result price does not match the expected result.	■ Verify that the pricing factors you are testing are activated and are executing in the intended sequence. ■ Verify that the pricing factor field data has been entered correctly, especially the Sequence, Type, Adjustment Amount, and Type of Calculation fields. ■ Verify that multiple factors are not conflicting with each other. ■ Verify that a manual discount is not overriding the price calculated by the Pricing engine. ■ Delete or rename the log file. The Pricer engine will create a new one when you execute a pricing factor. Then use the Active field to test each pricing factor in the pricing model individually. ■ Verify that the pricing model is linked to the correct price list or component-based product. If it is not, link it, and then exit and re-enter the application before retesting. ■ Check the Pricing Engine Log file to see if the pricing factors executed.
A factor that uses a Field Value is not executing.	When this happens it is often the result of a typographical error when entering the Field Value. Fix the error and retest. If necessary, check the Right Side and Left Side fields in the Pricing Engine Log file to make sure they are the same.
An amount calculated by an aggregate pricing factor does not match the expected results.	■ Verify that the business component fields used with an aggregate pricing factor are amount fields. ■ Click Reprice All to recalculate the aggregate amounts.

Table 45. Troubleshooting Testing Problems

If this happens...	Try this
A factor that uses a mapping is not executing	■ Check the Right Side and Left Side fields in the Pricing Engine Log file to determine what values, if any, are being returned. If the Right Side value is blank, it may mean that there is a data type mismatch, such as a mismatch between the mapping field and the business component. In necessary, use Siebel Tools to determine the data type. ■ Use Siebel Tools to verify that the fields are active. If they are not active, activate them by setting Force Active = TRUE and then retest.
Multiple sessions are running. The pricing model has been updated, but the updates are not executing in all sessions.	When multiple sessions are running, the Pricing Administrator using each session must click Reload to use the most current version of the model. For example, if Pricing Administrator A makes a change to a pricing model and saves the changes, then Pricing Administrator B, who has a different session active, can see the pricing model changes. However, the changes will not execute in Pricing Administrator B’s session until the pricing model has been reloaded. To determine how and when to use Reload to update caches, consult with your server administrator.

Siebel ePricer Deployment and Integration A

This chapter contains information and examples showing how to integrate Siebel ePricer with Siebel business components and with external applications.

This chapter covers the following topics:

Supported Integration Mechanisms

Siebel ePricer supports integration with external as well as internal applications. External integration allows ERP or Web storefront applications to use Siebel ePricer as a central pricing engine. Internal integration allows users to create rules based on any business component. Therefore, objects such as policies and opportunities can display prices based on pricing rules. The Siebel ePricer is essentially a business service called Pricing Manager.

Refer to *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II* to help you with the process of integrating Siebel applications, including Siebel ePricer.

Refer to *Siebel Tools Reference* for more information on creating and managing objects in Siebel Tools. Refer to *Siebel Object Interfaces Reference* for more information on defining and using property sets and business services with Siebel VB and Siebel eScript.

Any Siebel ePricer integration effort requires familiarity with Siebel Tools as a prerequisite. You should plan to work closely and continuously with a Siebel Tools administrator to complete all but the most elementary integration tasks.

Note that the publications mentioned above do not provide all of the specific names, methods and locations that you will need in order to complete a Siebel ePricer integration. This chapter will provide many of these specifics, while the publications listed above will put them into context. For information about third party products, refer to the documentation supplied with those products.

About Pricing Integration

Siebel ePricer can be called as a business service internally or externally, that is, from other Siebel application products or from other manufacturers' applications. Siebel ePricer functionality is used internally by eSales, eEvents, eTraining and several other Siebel applications, and can be used with popular external applications and services such as those provided by ERP applications. While pricing is clearly associated with sales, orders, and electronic shopping cart functionality, many different systems across the enterprise may need to interact with the runtime Siebel ePricer engine. Internal integration enhancements will allow a user to create rules on any Siebel business component.

Figure 13 shows how the Pricing Manager business service can be integrated to provide Siebel ePricer functionality to both internal and external system modules.

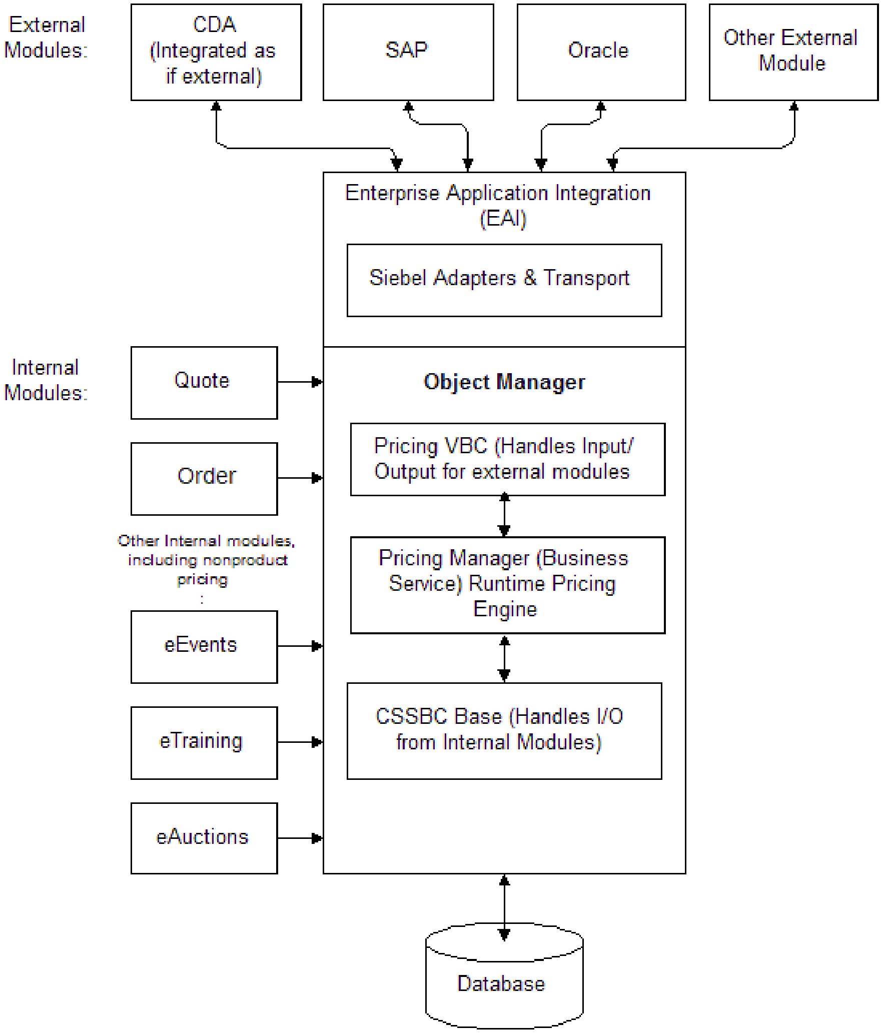

External Modules:
CDA (Integrated as if external)
SAP
Oracle
Other External Module
Enterprise Application Integration (EAI)
Siebel Adapters & Transport
Internal Modules:
Quote
Order
Other Internal modules, including nonproduct pricing
:
eEvents
eTraining
eAuctions
Object Manager
Pricing VBC (Handles Input/ Output for external modules
Pricing Manager (Business Service) Runtime Pricing Engine
CSSBC Base (Handles I/O from Internal Modules)
Database

Figure 13. The Pricing Manager Business Service (the Runtime Pricing Engine)

Terms and Definitions

If you are not familiar with Siebel ePricer, its business service, or the architectural context in which they exist, then the following definitions should provide some basic orientation:

- Pricing administration screens are used create and manage price lists, and to insert, edit and modify Pricing rules.
- Quote views and Order views or other end-user interfaces interact with the runtime Pricing engine, that is, the Pricing Manager business service.
- The Pricing Manager business service and its methods are made available in Siebel Tools.
- Siebel applications ("Internal applications") that work with CSSBCBase can invoke Pricing Manager services in runtime.

About Calls to Pricing Manager

To use Siebel ePricer as a business service, you must create calls to the Pricing Manager business service that satisfy the following three high-level requirements.

The calls must be able to:

- Invoke Pricing Manager, the Siebel ePricer runtime engine.
- Create pricing rules that effectively configure and populate Pricing Administration views.
- Test and validate pricing rules in the integrated suite of products.

  This last task involves different views, depending upon whether you are integrating internal or external modules.

  - If you are testing Internal modules, then you test pricing rules in the Quote Validation view and the Order Validation view.
  - If you are testing external modules, you test pricing rules in the Business Service Simulator or use a test build of your own application with the integrated Pricing Manager business service. The Business Service Simulator can be found in Business Service Administration.

For each kind of interface, this chapter reviews the procedures for completing the high-level requirements. At the end of this chapter is an API guide and an introduction to the Siebel business component user properties that are used to configure the pricing engine.

Transports and Adapters

Set up a COM interface so that the external system can call the Siebel business service through OLE.

About Using Siebel ePricer as a Business Service

The runtime Siebel ePricer engine code is embedded in a business service called Pricing Manager. This business service is based on the class CSSPricerService. Much of the CSSPricerService code and functionality is inherited from the class CSSBCBase. This is the base class that is used for providing basic business component functionality. As such, all the Pricing Manager routines can be invoked from applications that are based on a business component.

To use Pricing Manager, you will need basic API information with details about the methods, arguments, Siebel Tools business components, and user properties. This information is provided at the end of this chapter is the section titled "Siebel Pricing Manager API" on page 288.

Internal Integration, an Example

The following is an example of how the Opportunities view can display prices based on pricing rules. Customers that start with the Opportunities view in their business process sometimes want to see the effect of pricing rules on selected products within the Opportunities view itself.

Configuring the Opportunity Business Component

Configure the Opportunity business component to include the necessary fields to display pricing.

To configure the Opportunity business component

1 Add Business Component User Properties to the Opportunity business component.

This identifies pricing-related information on the Opportunity business component. For the definition of all pricing-related user properties, see Table 52 on page 289.

Name	Value
Additional BusComps for Siebel ePricer	Opportunity
Discount Amount Field	Cost
List Price Field	Product Price
Price List Id Field	Opportunity.Name
Original List Price Field	Revenue
Pricing Comments Field	Comment
Product Id Field	Product ID
Quantity Field	Product Quantity

2. Add a Reprice button to the Opportunity Product applet.

   When users press this button, the pricing engine is called. Use the following standard workflow for adding buttons. This workflow is described in fuller detail in *Siebel Tools Reference*.

   - Alter the Opportunity Product Applet
   - Add Controls "Reprice"
   - Display Web template
   - Add controls to template

Setting Up Products and Pricing information in the Siebel Application

You can perform the following in any base application (for example, Siebel Sales or Siebel Call Center). For a detailed description of how to create products, refer to *Product Administration Guide*.

To set up products and pricing information

1. Create the following new products in Product Administration:

   - How the Grinch Stole Christmas
   - If I Ran the Circus

2. Create a new price list called Internal Integration.
3. Add the products you created to the price list.

Product	List Price
How the Grinch Stole Christmas	$11.20
If I Ran the Circus	$11.96

Adding the Opportunity Buscomp to the PRICING_BUS_COMPS LOV

In order to create rules on prices in the Opportunity business component, you must add the Opportunity and Opportunity Product business components to PRICING_BUS_COMPS LOV.

To add the Opportunity business component to the pricing factor

1. Navigate to Application Administration > List of Values.
2. Create a new record for Opportunity Product.

Field Name	Value
Type	PRICING_BUS_COMPS
Display Value	Opportunity Product
Language Independent Code	Opportunity Product
Language Name	English – American

3. Create a new record for Opportunity.

Field Name	Value
Type	PRICING_BUS_COMPS
Display Value	Opportunity
Language Independent Code	Opportunity
Language Name	English – American

Creating a Pricing Model and Pricing Factors

The next step in the internal integration example is to create the pricing model and pricing factor on the Opportunity business component.

To create a pricing model and pricing factors

1. Navigate to Pricing Administration > Pricing Model Manager.

2 Create a new record with name Internal Integration.

3 Navigate to Pricing Factor Designer and create a pricing factor on the Opportunity business component.

Field Name	**Value**
Name	Opportunity's Description is software
Business Component Name	Opportunity
Business Component Field	Description
Operator	=
Field Value	Software
Type of Calculation	% Discount
Adjustment Value	5
Comments	Because your opportunity is software related, there will be a 5% discount!

4 Create another Pricing Factor based on a product on the Opportunity business component.

When finished, remember to unlock the model.

Field Name	**Value**
Name	Opportunity product is How the Grinch Stole Christmas
Business Component Name	Opportunity Product
Business Component Field	Product
Operator	=
Field Value	How the Grinch Stole Christmas
Type of Calculation	% Discount
Adjustment Value	10
Comments	Because your product is How the Grinch Stole Christmas, there will be a 10% discount!

Associating the Pricing Model with the Price List

The next step is to associate the pricing model with a price list.

To associate the pricing model with the price list

1. Navigate to Pricing Administration > Price List.
2. Query for the Internal Integration price List.
3. In the Pricing Model column, select Internal Integration.
4. Open the Help menu and choose About Record.
5. Write down the row ID (price list ID).

Testing Your Work

The last step is to test your work.

To test your work

1. Navigate to Opportunities and create a new record.

Field Name	Value
Name	<Price List ID> that you copied
Description	Software

2. Go to Products tab and create a new record.

Field Name	Value
Product	How the Grinch Stole Christmas

3 Click the Reprice button.

You should see the following:

Field Name	Value
Revenue (List)	$11.20
Cost (Discount Amount)	$1.62
Net Price (Adjusted List Price or Start Price on Quote)	$11.20
Comments	Because your opportunity is software related, there will be a 5% discount!, Because your product is How the Grinch Stole Christmas, there will be a 10% discount!

If you want to use a price list name instead of an ID, you can do so by adding the price list name to the business component user properties of the Opportunity business component.

External Integration

External applications such as ERP or Web store-front applications already have quote or order type structures. In the business process, a user will create a quote or order and pass that quote to a central pricing engine, Siebel ePricer, to be priced.

Siebel ePricer will execute methods based on the structures to come up with a price for each line and then the total. In order for the Pricing Manager business service to price an entity (product or service), the business service must receive the relevant information in a format the pricing engine can understand.

To provide the right information in the right format, you will need to create the following:

- Business components in the Siebel application that map to the external quote or order
- Property Set Object

Creating New Business Components

These new business components will typically mimic the Siebel Quote Object, which consists of Quote (header), Quote Item, and Quote Item XA. The Quote header contains information such as Account Name, Price List, Contact Name.

The Quote Item contains line level information such as the Product Name, Quantity Requested, Net Price. The Quote Item XA contains the product attribute and attribute value information.

Your specific pricing needs may only require the Quote header and Quote item business components. You must assign these business components to the either the CSSPricerVBC class or the CSS BCBase class.

The Property Set Object

The property set object is used to pass information to the Pricing Manager. A standard object that Siebel uses for Service Call arguments is CSSPropertySetEx. The property set object consists of the Siebel Property Pair and Siebel Property Set.

CSSPropertySetEx handles two things: (1) defining the hierarchy and relationships among various parts of the object and (2) defining the name/value combination of a particular attribute of the object.

For complete details about property sets, refer to *Siebel Object Interfaces Reference*. Specific details about property sets for Pricing Manager are covered in later sections of this chapter. Most of the following information is derived from *Siebel Object Interfaces Reference*.

In the CORBA Object Manager, the Property Set object is implemented as a structure (and potentially an array of structures) called SiebelPropertySet in the IDL, rather than as an interface. The client application constructs a SiebelPropertySet in C++ and then passes it to all methods that take property sets as arguments, for example, SiebelService InvokeMethod.

The advantage of this implementation is that the property set can be constructed completely on the client side instead of incurring network traffic associated with constructing it through an interface. The methods defined for the PropertySet differ greatly from the methods available through COM. However, the same functionality is available through either interface.

A SiebelPropertyPair represents a distinct name/value combination to describe an attribute of an object. The SiebelPropertyPair structure is defined in the IDL as follows:

```
struct SiebelPropertyPair

{

   string name; //Name of the attribute

   string value; //Description of the attribute

};
```

A SiebelPropertySet represents a collection of SiebelPropertyPairs (structures) that describe an Object and its attributes. The SiebelPropertySet structure is defined in the IDL as follows:

```
struct SiebelPropertySet

{

   string type;

   string value;

   sequenceSiebelPropertyPair properties;

   sequenceSiebelPropertySet  children;

};

typedef sequenceany anySeq;
```

The following C++ code fragment from a client application demonstrates the construction of a SiebelPropertySet and how to pass it to a SiebelService method.

```
// Declarations

SiebelPropertyPair  inputPP1, inputPP2, inputPP3;

SiebelPropertySet   inputPS;

SiebelPropertySet*  outputPS;

SiebelService pService;


// Setting up the Input Property Set

     inputPS.type = "root";

     inputPS.value = "rootValue";


// Specifying the number or SiebelPropertyPairs that will be

// added to the PropertySet

     inputPS.properties.length(3);
```

```
// Defining the first property
     inputPP1.name = "prop1";
     inputPP1.value = "value1";
     inputPS.properties[0] = inputPP1;

     inputPP2.name = "prop2";
     inputPP2.value = "value2";
     inputPS.properties[1] = inputPP2;

     inputPP3.name = "prop3";
     inputPP3.value = "value3";
     inputPS.properties[2] = inputPP3;

// Adding 2 Child Property Sets
     inputPS.children.length(2);

     inputPS.children[0].type = "child1";
     inputPS.children[0].properties.length(1);
     inputPS.children[0].properties[0].name  = "child1:prop1";
     inputPS.children[0].properties[0].value = "child1:value1";

     inputPS.children[1].type = "child2";
     inputPS.children[1].properties.length(2);
     inputPS.children[1].properties[0].name  = "child2:prop1";
```

```
         inputPS.children[1].properties[0].value  =  "child2:value1";
         inputPS.children[1].properties[1].name  = "child2:prop2";
         inputPS.children[1].properties[1].value  = "child2:value2";


    // Passing the Input Property Set and a pointer to the Output
    // Property Set to the Service Method
         pService->InvokeMethod ("test1", inputPS, outputPS);
```

The incoming call includes arguments containing, at the top level, the price list ID, the product ID, quantity, and other pricing-related business component data. Details of preparing data for CSSPropertySet for the Pricing Manager are discussed in the sections that follow.

Setting Up Property Set Data for CSSPropertySet

CSSPropertySet has a tree structure in which each node contains name and value pairs. The root node provides fundamental data pointing to the price list, product, and quantity.

You cannot send a pricing model ID in an external Siebel ePricer call.

The root node contains the following names and values:

Table 46. Product data passed with CSSPropertySet

Name	Value	Required
Price List Id	The unique ID of the price list to be used	Yes
Product Id	The unique ID of the product to be priced	Yes
Quantity	The quantity of the product (with the specified Product ID) for the quote or order	Yes
Original List Price	Optional: A list price that will override the list price for the specified price list and product ID	No

CSSPropertySet Contents for Pricing Model Data

If a customer's system sends a Price List ID or a Price Model ID, the system expects a price model to be applied to the item's price. In order for Pricing Manager to evaluate and process the model data, the external system must also provide the business component field value. The immediate child nodes of the root provide the header/parent business component record. Below this node is another child node, which provides the child records of the header/parent record.

Each node representing a record has the format shown in Table 47.

Table 47. Each Node Representing a Record Has the Following Format

Name	Value
BusComp_Name	The exact business component name
Row_Number	The row number of the record: 1, 2, 3, ...
Fieldname 1	Fieldname 1 value
Fieldname 2	Fieldname 2 value
Fieldname 3	Fieldname 3 value

CSSPropertySet Contents for Aggregate Pricing Model Data

For an aggregate pricing model, the root node should contain the following information:

Table 48. CSSPropertySet Root Node for Aggregate Pricing Models

Name	Value	Required
List Total	The total of the list prices for all items	Yes
Current Total	The total current price	Yes

Child nodes of the root node should include the information in the header/parent and multiple child records.

Return Values

The return object of the service call is also in a CSSPropertySet. For single product repricing, it contains the following information:

Table 49. CSSPropertySet Return Content for Single Product Repricing

Name	Value
Currency Code	The code specifying the currency used for the price
List Price	The list price
Current Price	The net price or final price
Upsell Message	The upsell message, for any volume discount
Reason	The reason for all the discounts in the model

For aggregate pricing, it contains the following information:

Table 50. CSSPropertySet Return Content for Aggregate Repricing

Name	Value
Final Discount Price	The final price for the whole quote

Pricing Nonproduct Items

Nonproduct items include such things as training, events, or literature on a business component. For a business component defining an entity that you wish to process with the Siebel ePricer business service, in the business component user property, there is a field called Product Item Flag Field. If you set this flag to True, this entity can be priced as a product, otherwise it is priced as a nonproduct. This field indicates whether the item has a product ID or not. When a nonproduct is passed, the engine will get the starting price from a source other than the price list. The method is called CalculatePriceExt.

When this method is called, and you pass in Price List Id, Product Id, Original List Price, Quantity, and request for a price, Siebel ePricer will price this as a product.

When this method is called, and you pass in Pricing Model Id, List Price, Volume Discount Id, Price Book Id, and request for a price, Siebel ePricer will price this as a nonproduct.

If the external system is pricing a nonproduct item, then the root node should provide the information shown in Table 51. Note that the default for user property on quote item is set to False. If you want to use something other than quote item to capture the product/service item to be priced, you will need to set up the user property for that quote item-like object to have a value of False for the Product Item Flag field.

Table 51. NonProduct Data Passed with CSSPropertySet

Name	Value	Required
List Price	The start price to be used for this nonproduct item	No
Price Model Id	The ID for the price model to be applied	No
Volume discount Id	The ID for the price model to be applied	No
Price Book Id	The ID for the price book (attribute pricing table) model to be applied	No

Using the Business Service with SAP, an Example

SAP can call Siebel ePricer as a business service. From SAP, a request is sent to GetProductListPrice along with parameter values including account name and product name. The Siebel ePricer business service will determine the price of the product and send the price to SAP. Three new business service methods with input and output arguments will be passed as name value pairs.

The supported methods include the following:

- GetProductListPrice (Input - Product Id, Price List Id, output)
- CalculatePriceExt
- CalculateFinalPriceExt

The object manager processes the flow between SAP and Siebel. The request from SAP is converted by EAI into Siebel Integration Object format. This is processed by Workflow Manager. The methods are incorporated in the workflow process. The input and output name value pairs will be passed between workflow processes. The process simulator displays the value being passed. The emulation screen in EAI emulates messages passing through.

Creating Business Components

The first step is to create the business components that you will use with the pricing factors.

To create business components

1. Create a new business component called External Quote that is based on Siebel Quote.

Field Name	Value
Name	External Quote
Project	External Project
Class	CSSPricerVBC or CSSBCBase

2 With the business component selected, go to the Fields applet and create the following records:

Field Name	Value	Calculated
Boolean	DTYPE_BOOL	Yes
Currency	DTYPE_CURRENCY	Yes
Date	DTYPE_DATE	Yes
Date Time	DTYPE_DATETIME	Yes
Some Id	DTYPE_ID	Yes
Integer	DTYPE_INTEGER	Yes
Note	DTYPE_NOTE	Yes
Number	DTYPE_NUMBER	Yes
Phone	DTYPE_PHONE	Yes
Text	DTYPE_TEXT	Yes
Time	DTYPE_TIME	Yes
UTC Date Time	DTTYPE_UTCDTATETIME	Yes

3 Create a new business component called External Quote Item by making copy of External Quote.

4 With the External Quote Item business component selected, go to the Fields applet and create the following records for quote-time specific data:

Field Name	Value	Calculated
Product Id	DTYPE_ID	Yes
Quantity Requested	DTYPE_INTEGER	Yes
Class	DTYPE_CURRENCY	Yes

5 For External Quote Item, create the following user properties:

Field Name	Value
Product Id Field	Product ID
Quantity Id Field	Quantity Requested
Net Price Field	Item Price

6 Create a new business component, External Quote ItemXA:

Field Name	Value
Name	External Quote ItemXA
Project	External Project
Class	CSSPricerVBC or CSSBCBase

Creating the External Business Object and Compiling the SRF

The next step is to create a business object and to compile the SRF file.

To create the business object and compile

1 Create new business object, External Business Object.

2 In the Business Object Components applet, create the following records:

- External Quote
- External Quote Item
- External Quote ItemXA

3 Go to Siebel Objects > Business Service and query for Pricing Manager.

4 Lock the project and go to Business Service > Business Service User Prop.

5 Add a new record:

Field Name	Value
External Siebel Pricer BO Name	External Business Object

6 Compile the application SRF file.

Setting Up Pricing and Rules

The next step is to set up product classes and attributes to support attribute-based pricing. You also need to create a price list and add the products to the price list.

To set up pricing and rules

1 Navigate to Application Administration > Class Administration.

2 Create a product class, Automobile.

3 Create dynamic attributes for the Automobile class.

4 Create attribute pricing.

5 Create product(s) and assign them to the Automobile class.

6 Create a price list.

7 Navigate to Application Administration > List of Values and add the following records:

Type	Display Value	Language Independent Code	Language Name
PRICING_BUS_COMPS	External Quote	External Quote	English-American
PRICING_BUS_COMPS	External Quote Item	External Quote Item	English-American
PRICING_BUS_COMPS	External Quote ItemXA	External Quote ItemXA	English-American

Creating the Business Service

The next step is to create the business service used to populate the new business components from the external application.

After you create the business service, you are ready to use Workflow and the Simulator to test your new business components. Once you have successfully verified all pricing, you are ready to hook up Siebel ePricer to an external application through the scripts.

To create the business service

1. Navigate to Business Service Administration > Business Service Methods.
2. In the Business Service applet, create a new record with the name Pricer Integration.
3. In the Methods applet, create a new record with the name Pricer Integration Method.
4. In the Business Service Scripts applet, create a new record called Service_PreInvokeMethod where the Program Language is eScript.
5. In the Business Service Scripts applet, enter your script.

Integration Risks, Constraints, and Best Practices

Depending on the call to the Siebel ePricer business service, the following may be used:

- Siebel Enterprise Application Integration (EAI)
- Com
- CORBA
- Java beans
- Http-based integration objects

Virtual Business Components and Matrix-Based Pricing

Virtual business components can be created on a flat text file that contains pricing data. The flat file must be created such that it defines records containing:

- Key data that will enable the corresponding matrix search specifications to identify specific records
- Matrix-based pricing rule price adjustment data

This virtual business component is referenced by a matrix-based pricing factor in a Siebel ePricer pricing model. When pricing requests are made in a quote, the virtual business component is used to get pricing data based on the flat text file.

Siebel Pricing Manager API

This section includes the following information:

- "Pricing Manager Business Component User Properties" on page 289
- "Pricing Manager Business Service Methods" on page 297

Pricing Manager Business Component User Properties

Pricing Manager business component user properties are described in Table 52. These are the user properties on the newly-created business components used to call the Pricing Manager.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Additional Buscomps for Siebel ePricer I/O Type: Input DTYPE: n/a	Any buscomp being executed for pricing within the current business object, for example in order entry: Order Entry > Order Entry" and Order Entry > Line Item.	"Quote, Quote Item XA	Used to identify the business components that will be used by the pricing engine. The sample data indicates that Quote and Quote ItemXA buscomp are used. This allows the user to define rules similar to this: If Quote.Account = A.K. Parker, then apply discount of 10%.	No
Calculate Price Post Method Invoke 0 I/O Type: n/a DTYPE: n/a	Any business component method supported by an active business component can be added here. Use the arguments supported by the method, for example, Quote.RefreshMVG BusComp (Quote Item).	UpdateServicePrice		No

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Calculate PriceAll Pre Final Method Invoke 0 I/O Type: n/a DTYPE: n/a	Any business component method supported by an active business component can be added here. Use the arguments supported by the method, for example Quote.RefreshMVG BusComp(Quote Item).	Quote.RefreshMVGBusComp (Quote Item)	CalculatePriceAll concerns 2 processes: (1) calculate pricing for each single item and (2) calculate aggregate pricing. The method in the example is invoked between these processes. The effect of this is to refresh the Quote Total such that it holds the correct sum of prices from items, since item prices change during Step 1. This causes Siebel ePricer to invoke the Quote.RefreshMVGBus-Comp method with the argument “Quote Item" immediately after single product pricing for each item, which is before the aggregate portion of CalculatePriceAll method is called.	Yes. (Unless Customer needs to change the fundamental way in which quote totals are calculated.)
Complex Object Instance Name I/O Type: n/a DTYPE: n/a		Quote	For customizable products: If the customizable product is supported for this buscomp, this indicates the name of the integration object for the customizable product.	Required only if customizable product functionality is to be supported.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Complex Object Instance Root Id I/O Type: n/a DTYPE: n/a		Quote.Id	This is the ID of the header of the customizable product instance. For example, it would be the row Id of the Quote Header. Every customizable product requires a Header ID, a Root Product ID, and an XA).	Required only if customizable product functionality is to be supported.
Current Volume Discount Field I/O Type: Output DTYPE: n/a	Name of the field that would receive the name of the current volume discount.	Volume Discount Item	The volume discount to which you are currently entitled, based on the quantity.	Only required if volume discount needs to be displayed.
Current Volume Discount Id Field I/O Type: Output DTYPE: (Siebel) Id	Name of the field that would receive the row ID of the current volume discount.	Volume Discount Item Id	The ID of the volume discount to which you are currently entitled, based on the quantity.	Only required if volume discount needs to be displayed.
Discount Amount Field I/O Type: Output DTYPE: Number	The name of the field that displays the total discount amount as calculated by the pricing engine.	Discount Amount	The field in which Discount Amount is to be stored after the pricing calculation. This is the main result field. After all volume discount and Pricing factor calculations are completed, the difference between start price and net price will be output to this field. The value includes volume discounts and all pricing factors except pricing attribute adjustments (because attribute adjustments are already included in the start price).	Yes

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Discount Comments I/O Type: Output DTYPE: Text	The name of the field that displays the pricing comments for the aggregate factors.	Quote.Discount Reason	The field in which pricing comments are to be stored. These are the reasons for firing each aggregate pricing factor that adjusted the quote total.	No, unless Aggregate factor is used.
Discount Source Field I/O Type: Input and Output DTYPE: Text	Field name to store the discount source code as outlined in the description. When Siebel ePricer changes the discount amount, this field is updated to value *Pricer*. When users override the discount fields (for example, discount%, discount price, discount amount), this field is updated to *Manual*. The default value is *Pricer*.	Discount Amount	This field stores the source of the discount, which must be one of two values, *Pricer* or *Manual*. *Pricer* indicates that this discount results from a Siebel ePricer engine calculation. *Manual* indicates that this discount results from a manual operation.	This is not required, but without this field, all discounts will be treated as *Pricer* discounts.
Exclude Pricing Flag Field I/O Type: Input DTYPE: Bool	The name of the field that displays the Keep Discount flag.	Keep Discount	A Boolean field to indicate whether the pricing engine should keep the current discount. If this field has a runtime value of TRUE, the pricing engine will not update the discount fields: (Disc%, Discount Amount, Discount Price). If not supplied, the pricing engine treats this value is FALSE.	No.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Is Complex Product Root I/O Type: Input DTYPE: Bool	The name of the field that is set to TRUE for line items that are the root product of a customizable product.	IsComplexProductRoot	Indicates that this product is the root product of a customizable product.	Not required if no customizable product is to be supported.
List Price Field I/O Type: Output DTYPE: Currency	The field name that contains the start price.	Adjusted List Price	The List Price field, includes attribute adjustments on they original list price or promotional price if they exist.	Yes
Need Refresh List Price Field I/O Type: Input DTYPE: Bool	The field name that contains the indicator for refreshing the list price.	Quote.Quote Expired	When the value of this field is TRUE, the pricing engine will use the latest price list price to calculate the pricing. Otherwise, the pricing engine will use the existing base price of the item to update the price. When the quote has expired, the pricing engine will refresh the start price from the price list.	No. Default Value is FALSE
Net Price Field I/O Type: Input DTYPE: Currency	The field name that contains a calculated value for net price.	Item Price	The net price field for the item. This field is likely to be a calculated field resulting from Start price and the discount Fields (Disc%, Discount Amount, Disc Price).	Required for bundle type of factor.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Next Volume Discount Field I/O Type: Output DTYPE: Text	Field name used to store the name of the next level volume discount.	Volume Upsell Item	Similar to current discount. This is the next higher level of volume discount for the Item. More specifically, this is the next higher level or tier of volume discounting, based on the current quantity. The buyer would be entitled to this volume discount if the buyer purchased a qualifying higher quantity of the Item.	Required if volume discount is used.
Next Volume Discount Id Field I/O Type: Output DTYPE: (Siebel) Id	Field name used to store the row Id of the next level volume discount.	Volume Upsell Item Id	Similar to current discount ID. This is the ID of the next higher level of volume discount for the Item. See description of NextVolumeDiscount Item.	Required if volume discount is used.
Original List Price Field I/O Type: Output DTYPE: n/a	The name of the field that contains the list price from the price list.	Base Price	The original list price field. The value is from the price list. It is set to the promotional price if there is one. If not, it is set to the list price. This price does not include attribute pricing adjustments.	Yes
Price List Id Field I/O Type: Input DTYPE: (Siebel) Id	The name of the field that contains the price list row Id.	Quote.Price List Id	This field contains the price list ID. (usually stated at the quote header).	Yes
Price Model Id Field I/O Type: Input DTYPE: (Siebel) Id	The name of the field that contains the pricing model row Id.	Quote.Price Model Id	This field contains the price model ID and is used only for nonproduct pricing.	Yes only if non-product pricing is involved.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Pricing Comments Field I/O Type: Output DTYPE: Text (250 char limit)	The name of the field that contains line item price model comments.	Pricing Comments	The field in which pricing comments are to be stored. Contains the reasons for firing each pricing factor that adjusted the quote Item.	Yes
Pricing Enabled I/O Type: Input DTYPE: Calculated	This is set to 'Y' to indicate the business component needs to invoke Siebel ePricer.	Y	This field is a user property that is set for a business component that is used to trigger the Siebel ePricer logic.	
Product Id Field I/O Type: Input DTYPE: (Siebel) Id	This is the name of the field that stores the product ID.	Product Id	Contains a unique product Id.	Yes
Product Item Flag Field I/O Type: Input DTYPE: Bool	This is the name of the field that indicates this as a product or nonproduct item.	ItemIsProduct	If this flag is TRUE, the record is treated as a product record. Otherwise, the record will be treated as nonproduct item such as an eTraining course or an event.	No. Default Value is TRUE
Quantity Field I/O Type: Input DTYPE: n/a	This is the name of the field that stores the item quantity.	Quantity Requested	Quantity of the product.	Yes

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Skip Pricing Field I/O Type: Input DTYPE: Bool	This is name of the field that contains the indicator of whether or not Siebel ePricer should fire.	Skip Pricing Flag	If this flag is set to TRUE, the entire pricing operation is skipped. This is different from the Exclude pricing flag, which only excludes the discounting functions but still calculates the start price and original list price. Currently this field is used to skip the pricing of customizable product components. Pricing for customizable product components should be priced only within the object instance.	No, Default is FALSE.
Total Current Price Field I/O Type: Input DTYPE: Currency	This is the name of the field that stores the total extended price of all items in the quote.	Quote.Total Current Price	This is the total of all Net Price values for items in the quote.	No, unless Aggregate factor is used.
Total Discount Amount I/O Type: Output DTYPE: Currency	This is the name of the field that stores the total extended amount from the bundle or aggregate factors.	Quote.Discount Amount	This is the discount adjustment for aggregate factors at the quote (parent) level.	No, unless aggregate or bundling-type factors are allowed.
Total List Price Field I/O Type: Input DTYPE: Currency	This is the name of the field that stores the total non-extended price.	Quote.Total List Price	This is the total of all start price values for items in the quote.	No, unless Aggregate factor is used.

Table 52. Pricing Manager Business Component User Properties

Name, I/O Type, Data Type	Valid Values *	Sample Values	Explanation	Required
Upsell Message Field I/O Type: Output DTYPE: Text	The name of the field that should receive and display the upsell message.	Volume Upsell Message	This is the volume discount upsell message.	Yes, unless no volume discount is allowed.
Volume Discount Id Field I/O Type: Output DTYPE: (Siebel) Id	Not applicable	Volume Discount Id	This is not necessarily needed. It is supplied for historic purposes. This field stores the ID of the Volume Discount object. This is not the volume discount item, but the parent of it. It defines discount type and effective dates.	No, for volume discount.

* All of the values for field names have a format of: BusCompName.BusCompFieldName. If no BusCompName is supplied, it implies that the field appears in the main buscomp where the pricing button is pressed.

Pricing Manager Business Service Methods

The Pricing Manager business service is based on the class CSSBCBase.

This section describes the following:

- Pricing Manager Methods external to the application (Table 53 on page 298)
- Pricing Manager Methods known to Tools (Table 54 on page 298)

- Parameters for CalculatePriceExternal Method (Table 55 on page 299)

Table 53. Pricing Manager Methods External to the Application

Method Name	External / Internal	Purpose	Input Arguments	Output Arguments
CalculatePrice	Internal	Used for calling Siebel ePricer from other business components.	None	None
CalculatePriceAll	Internal	CalculatePrice for all quote Item discount and calculate aggregate price for the whole quote.	None	None

Table 54. Pricing Manager Methods Known to Tools

Method Name	External / Internal	Purpose	Input Arguments	Output Arguments
CalculateFinalDiscountExternal	External	Calculates aggregate pricing for external application user. A multiple item record must be passed as the argument. Assumes the CalculatePrice functions 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 Siebel ePricer from other business components.		

Table 54. Pricing Manager Methods Known to Tools

Method Name	External / Internal	Purpose	Input Arguments	Output Arguments
CalculatePrice-External	External	Single Item repricing for external users.	See Table 55 on page 299.	
GetFinalDiscount	Obsolete - will be removed	Obsolete		
GetProductListPrice	External/ Internal	Obsolete		
Reload Cache	External/ Internal	For performance, data is cached in memory, this helps indicate to the Object Manager that the database has changed. It tells the Object Manager to re-read factors from the database.	None	None
SetPriceList	External/ Internal	To set the price list at the profile attribute level. This is currently used in eSales to override with the Price list id set by the session.	No input because this is set 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.	None	None

Table 55. Parameters for CalculatePriceExternal Method

Name	Optional	Type	Explanation
Currency Code	N	Output	The currency code for the price.
Current Discount	N	Output	Current volume discount.

Table 55. Parameters for CalculatePriceExternal Method

Name	Optional	Type	Explanation
Current Discount Id	N	Output	Current volume discount Id.
Current Price	N	Output	The final price.
List Price	N	Output	Start price.
Next Discount	N	Output	Next available item for volume discount.
Next Discount Id	N	Output	
Original List Price	Y	Input	
Price List Id	N	Input	
Pricing Comments	N	Output	
Product Id	N	Input	
Quantity	N	Input	
Service Price Percentage	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.
Volume Discount Id	N	Output	Volume discount ID (a copy of the ID in Quote).

Standard Configuration Settings

Configuration settings can be set at the server level and at the remote client level. The standard configuration file includes a section called Pricing Cache which should appear as follows:

```
[PricingCache]

PricerPriceListCacheSize = 50

PricerPriceItemCacheSize = 100

PricerVolDisCacheSize = 50

PricerPriceModelCacheSize = 50
```

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. The cache sizes above are samples. As a rule, you should not change the cache sizes specified for your system unless advised to do so by your server administration.

The range of acceptable cache sizes varies depending upon the machine, except that there is a logical minimum. The minimum is logically two, since a cache size of one would effectively mean that no caching could take place. The maximum depends on the practical maximum that is defined operationally. You can cache more records on a bigger machine that has more memory without creating problems.

The elements in the local Pricing Cache section function as follows:

- **PricerPriceListCacheSize**. This applies to the price list records at the header level.
- **PricerPriceItemCacheSize**. This applies to the price list line item records for each price list cache.
- **PricerVolDisCacheSize**. This applies to the volume discount records, including their children.
- **PricerPriceModelCacheSize**. This applies to the model cache.

If the preceding parameters are set at the local level, then they override settings for the same parameters at the server administration level.

The following parameters are set at the server administration level, and typically do not appear at the local level. If the following parameters do appear at your local level, check with your system administrator to determine whether they will override settings for the same parameters at the server administration level.

The following parameters do not affect the customizable product pricing data. This is the data you define in the Pricing Designer for a specified customizable product. The customizable product pricing caches are not guaranteed to be refreshed within any specific time period. Customizable product pricing caches refresh one port at a time. As you switch to another port in the customizable product, the customizable product pricing cache will refresh.

- **`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, there is a trade-off between refresh-time and loading speed. If you decrease a LifeTime value, the result is more frequent refresh of data and a long loading time for data. This is represented by the parameter `PARAM_SUBSYS_PRICE_LIFE-TIME`
- **`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`.
- **`PricerMappingCacheSize`**. This is the number of records to be cached for each mapping factor. The default is 100. If this value is set to zero, mapping information will not be cached, and the database will immediately be searched.

The first two parameters above are integers that specify a number of seconds. Their default value is -1, which means that the cache can remain unrefreshed indefinitely. Users can change this value to a number that is between 0 and 86400. 0 means turn off the cache; 86400 is the number of seconds in a full day.

Setting both numbers to low values results in more frequent refreshing and therefore slows down the performance, but it keeps the pricing information up to date.

Index