Siebel

Order Management Guide

September 2023

Siebel
Order Management Guide

September 2023

Part Number: F87464-01

Copyright © 1994, 2023, Oracle and/or its affiliates.

Authors: Siebel Information Development Team

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

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

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

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

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

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

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

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

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

Contents

Preface i

1 What's New in This Release 1

What's New in Siebel Order Managment Guide, Siebel CRM 23.9 Update 1
What's New in Siebel Order Management Guide, Siebel CRM 22.12 Update 1
What's New in Siebel Order Management Guide, Siebel CRM 22.11 Update 1
What's New in Siebel Order Management Guide, Siebel CRM 21.6 Update 2

2 Overview of Order Management 3

Overview of Order Management 3
About Order Management 3
The Order Management Life Cycle 4
Scenario for Order Management 4
Roadmap for Working with Order Management 5

3 Setting Up Order Management 7

Setting Up Order Management 7
Roadmap for Setting Up Order Management 7
Setting Up Inventory Locations 18
Setting Up Super Users for Product Administration 19
Setting Up Integration for Order Management 20
Lists of Values to Update for Order Management 29

4 Integrating Order Management with Back-Office Applications 33

Integrating Order Management with Back-Office Applications 33
Integration Strategy for Order Management 33
Smart Part Numbers and Order Management Integration 34
About ASIs for Order Management 34
Setting Up Quote and Order ASIs 36
Process of Integrating Quotes and Orders with Back-Office Applications 37

ASI Workflows for Quotes and Orders 39

5 Integrating Order to Cash and Distributed Order Orchestration 41

Integrating Order to Cash and Distributed Order Orchestration 41
Process of Integrating Order to Cash 41
Integrating Distributed Order Orchestration 52
Setting Up Data Validation Rules for Distributed Order Orchestration 52

6 Integrating Order Management with Third-Party Product Availability Applications 55

Integrating Order Management with Third-Party Product Availability Applications 55
About Third-Party Product Availability Applications 55
Process of Setting Up Third-Party Product Availability Applications 55

7 Integrating Order Management with Third-Party Credit-Check Applications 65

Integrating Order Management with Third-Party Credit-Check Applications 65
About Third-Party Purchase Order Credit Checking Applications 65
Process of Setting Up Purchase Order Credit Checking Applications 66

8 Integrating Order Management with Third-Party Taxation Applications 73

Integrating Order Management with Third-Party Taxation Applications 73
About Third-Party Taxation Applications 73
Installing Taxware Applications on Windows 74
Setting Up Taxware Applications on Windows 75
Installing and Setting Up Taxware Applications on UNIX 75
Enabling Tax Calculation 75
Taxing Customizable Products 77
Taxware Integration with Siebel eSales 77
Customizing Taxware Integration 84
Handling Unicode Conversion Errors 86

9 Integrating Order Management with Oracle ATG Web Commerce 89

Integrating Order Management with Oracle ATG Web Commerce 89
About Integrating Siebel Business Applications with Oracle ATG Web Commerce 89

Setup Tasks for Integrating Siebel Business Applications with Oracle ATG Web Commerce 92
Process of Exporting Records to Oracle ATG Web Commerce 95
Maintaining Product Data for Integration with Oracle ATG Web Commerce 102

10 Integrating with Oracle Intelligent Advisor 105
Integrating with Oracle Intelligent Advisor 105
About Creating Product Advisors 105
Process of Setting Up Oracle Intelligent Advisor to Create Product Advisors 105
Using Oracle Policy Automation Product Advisors 107

11 Distributed Order Orchestration 109
Distributed Order Orchestration 109
About Oracle Fusion Distributed Order Orchestration 109
Scenarios for Distributed Order Orchestration 109
Tasks for Using Distributed Order Orchestration 111

12 Creating and Managing Catalogs 115
Creating and Managing Catalogs 115
About Siebel Catalog 115
Roadmap for Creating Catalogs 116
Process of Setting Up Catalogs 116
Process of Setting Up Categories 119
Building Catalog Hierarchies 124
Example Process of Creating Display Templates for Categories 125
Process of Setting Up Attribute Search 128
About Dynamic Product Catalogs 130
About the Siebel Sales Catalog in Open UI 131

13 Setting Up Product Recommendations 135
Setting Up Product Recommendations 135
About Product Recommendations 135
Scenario for Product Recommendations 136
Creating Product Recommendations 137
Turning Off Product Recommendations 138
Guidelines for Implementing Product Recommendations 138
Product Recommendation Technical Reference 139

Configuring Recommendation and Scoring Logic 143
Unified Messaging Service and Product Recommendations 143

14 Creating a Quote or Sales Order 145
Creating a Quote or Sales Order 145
Process of Creating a Quote 145
Process of Creating a Sales Order That Is Not Based on a Quote 150
Common Tasks for Creating Quotes and Sales Orders 152
Changing the Default Price List and Rate List 183
How End Users Work with Product Promotions 184
Process of Enabling the Automatic Configuration of Customizable Products 186
Configuring Multiple Customizable Products 198
Changing Save All button behavior 199
200
Invoking GetProdInstInfo for Telecommunications Use case 203

15 Working with Quotes 209
Working with Quotes 209
About Quotes 209
Updating Opportunities from Quotes 209
Including Quote Items in a Forecast 210
Accessing An Existing Quote 210
Revising a Quote 211
Viewing Change History for Quote Line Items 212
Adding Line Item Fields to the Change History 212
Converting a Quote to an Order 213
Converting a Quote to an Agreement 213
Displaying Charts to Analyze Quotes 214

16 Working with Sales Orders 215
Working with Sales Orders 215
About Orders 215
Accessing an Existing Order 216
Submitting an Order 216
Associating an Activity Plan or Activity with an Order 217
Approving an Order 217

Checking the Status of Order Shipments 218
Viewing Shipping Details for Orders and Order Lines 218
Placing an Order on Hold 219
Canceling an Order 219
Revising an Order 220
Viewing Change History for Sales Order Line Items 220

17 Using Asset-Based Ordering 223
Using Asset-Based Ordering 223
About Asset-Based Ordering 223
Basic Concepts of Asset-Based Ordering 224
Scenario for Asset-Based Ordering 229
Creating a Quote Using Asset-Based Ordering 230
Creating a Quote That Modifies an Asset 232
Ungrouping Quote or Order Line Items 233
Editing Quote Line Items Using Asset-Based Ordering 234
Converting Quotes to Orders Using Asset-Based Ordering 235
Editing Order Line Items Using Asset-Based Ordering 235
Submitting an Order Using Asset-Based Ordering 236
Revising a Submitted Order Using Asset-Based Ordering 237
Creating a Follow-On Order 239
Modifying the Service Profile 239
Deactivating an Installed Asset 240
Suspending and Resuming Service 241

18 Workflow and Method Reference for Order Management Interface 243
Workflow and Method Reference for Order Management Interface 243
Order Management Interface Workflow Reference 243
Order Management Interface Methods Reference 256

19 Workflow Reference for Employee Asset-Based Ordering 261
Workflow Reference for Employee Asset-Based Ordering 261
About Workflows for Employee Asset-Based Ordering 261
Workflow to Ungroup Quote Line Items 262
Workflow to Ungroup Order Line Items 263
Workflow to Edit Quote Line Items 264

Workflow to Convert a Quote to an Order 266
Workflow to Edit Order Line Items 268
Workflow to Submit an Order 271
Workflow to Apply an Order to the Service Profile 272
Workflows to Modify the Service Profile 275
Workflows to Redisplay the Service Profile 276
Workflows that Allow Users to Modify a Service Item 278
Workflows to Disconnect a Service Item 284
Workflows That Allow Users to Suspend or Resume Service Items 287
Workflows for Contact-Based Asset-Based Ordering 291
Workflows for Quotes and Orders Assets Catalog Views 291
Workflows that Support the Automatic Configuration of Customizable Products 292

20 **Product Data Service and Import Export API Reference 299**

Product Data Service and Import Export API Reference 299
About the Product Data Service API 299
Product Data Services API Architecture 299
Product Data Services API Methods Reference 300
PDS PublishCatalog Test Sample Workflow 349
ISS Authoring Import Export Business Service Reference 350

Preface

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

Using Oracle Applications

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

Documentation Accessibility

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

Contacting Oracle

Access to Oracle Support

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

Comments and Suggestions

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

1 What's New in This Release

What's New in Siebel Order Managment Guide, Siebel CRM 23.9 Update

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

Topic	Description
Get Product Instance Information API *Invoking GetProdInstInfo for Telecommunications Use case*	New topic. GetProdInstInfo API is introduced to enforce inter-customizable products rules, while using configuring multiple customizable products view.
Changing Save All button behavior	New topic. This topic describes the steps to follow to change **Save All** button behavior.

What's New in Siebel Order Management Guide, Siebel CRM 22.12 Update

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

Topic	Description
Configuring Multiple Customizable Products *New View: Configuring Multiple CPs*	New topics. The new feature provides views for quotes and orders, which shows multiple *Customizable Products(CPs)* and allows users to switch from one CP to other quickly.

What's New in Siebel Order Management Guide, Siebel CRM 22.11 Update

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

Topic	Description
Setting Up Super Users for Product Administration *Configuring Super Users* *Unlocking Objects*	New topics. Product administrator *super users* can now unlock versioned objects that have been locked by other users.

What's New in Siebel Order Management Guide, Siebel CRM 21.6 Update

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

Topic	Description
ExecuteSearch Method	Modified topic. Added information on supported output for the ExecuteSearch method, including additional input and output argument properties.

2 Overview of Order Management

Overview of Order Management

This chapter provides an overview of the general order management process, and introduces the specific concepts and capabilities available through Siebel Order Management.

This chapter covers the following topics:

- *About Order Management*
- *The Order Management Life Cycle*
- *Scenario for Order Management*
- *Roadmap for Working with Order Management*

About Order Management

Oracle's Siebel Order Management allows employees such as salespeople and call center agents to create and manage quotes and orders through their entire life cycle. Siebel Order Management can be tightly integrated with back-office applications, allowing users to perform tasks such as checking credit, confirming availability, and monitoring the fulfillment process.

To administer order management, you must understand product and pricing administration. For more information, see *Siebel Product Administration Guide* and *Siebel Pricing Administration Guide* .

> **Note:** The Siebel Bookshelf is available on Oracle Technology Network (http://www.oracle.com/technetwork/indexes/documentation/index.html) and Oracle Software Delivery Cloud. It might also be installed locally on your intranet or on a network location.

Many order management features are available in Siebel eSales and Siebel Partner Portal applications. This book focuses on order management features in employee applications such as Siebel Sales and Siebel Call Center. For details about Siebel eSales and Siebel Partner Portal, see *Siebel eSales Administration Guide* and *Siebel Partner Relationship Management Administration Guide* .

Asset-based ordering allows quotes and orders to be created according to a customer's existing assets. Asset-based ordering is particularly useful in supporting companies whose product offerings include complex service products, such as phone services and equipment.

Siebel Order Management allows employees to:

- Create quotes and orders for new products and services
- Create quotes and orders to modify existing products and services
- Modify in-process orders that have been submitted for fulfillment
- Generate order information for submission to back office billing and fulfillment applications

The Order Management Life Cycle

The order management life cycle includes tasks that fall into several groups:

- **Preorder tasks.** Can include creating accounts, contacts, and opportunities or helping anonymous customers through the needs-analysis process.
- **Order entry tasks.** Can include selecting products and services, capturing quotes and orders, verifying products and pricing, checking availability to promise, entering shipping, tax, and payment information, and performing a credit check.
- **Post-order tasks.** Can include fulfilling the order, providing the customer with order status, notifying the customer that the order has shipped, creating supplemental orders, and monitoring or modifying activities that support the order, such as installation.

Scenario for Order Management

This topic gives one example of how order management might be used. You might use order management differently, depending on your business model.

- When your customer expresses interest in your products and services, you keep track of that information and help the customer identify the appropriate solution. A sales representative creates an opportunity and records the best solution to meet the customer's needs.
- After the best solution is identified, you provide a quote that details the products and their prices. A sales representative converts an opportunity to a quote, or creates a quote manually. Products and services are customized so that your customers specify exactly what options they are interested in and see the associated prices for each option.
- Your customer accepts a quote, and it is converted to an order. (Or, a sales representative creates an order directly, without creating a quote.)
- The sales representative enters shipment information and checks on the availability of the items. If an order contains a number of items, then the sales representative checks the availability of each line item. The sales representative also performs tasks such as calculating tax and shipping costs, verifying payment information, checking the customer's credit, and authorizing the use of a credit card.
- The sales representative performs final tasks, such as attaching electronic documents such as a purchase order or a letter of credit to the order, and generating service activities related to the order such as installation.
- The sales representative submits the order. If appropriate, the order is routed for approval by a supervisor. An acknowledgement of the order is automatically sent to the customer by email.
- If you are using asset-based ordering, then when the order has been filled, the appropriate product line items become assets. Assets are associated with your customer's account and are a central part of your customer's service profile.

When using asset-based ordering, not all products in an order convert to assets. For example, if you create an order for telephone service, then the installation is a line item of the phone service product, but it does not become an asset. You specify what products will become trackable assets through the Product Administration screen.

- If your customer wants to revise an order before it is fulfilled, then you handle the change by modifying unsubmitted orders or by creating supplemental orders that revise submitted orders.

- As the needs evolve over time, the customer may request additions and changes to the products and services. To make these changes, the new quotes and orders are based on the current items in the customer's profile. Requests for changes to existing services are called *delta quotes* or *delta orders*.

Note: Although end users may start the order management process at a number of different screens and views, the underlying order management cycle is essentially the same.

Roadmap for Working with Order Management

To set up and work with Siebel Order Management, perform the following processes and tasks:

Setting up order management.
For more information, see *Setting Up Order Management*

Integrating order management with third-party applications.
For more information, see *Integrating Order Management with Back-Office Applications*

Working with quotes and orders.
For more information how end users can work with quotes and orders, see *Creating a Quote or Sales Order Working with Quotes* and *Working with Sales Orders*

Working with asset-based ordering.
For more information about asset-based ordering, see *Using Asset-Based Ordering* and *Workflow Reference for Employee Asset-Based Ordering*

3 Setting Up Order Management

Setting Up Order Management

This chapter describes how to set up Siebel Order Management features. It includes the following topics:

- *Roadmap for Setting Up Order Management*
- *Setting Up Inventory Locations*
- *Setting Up Super Users for Product Administration*
- *Setting Up Integration for Order Management*
- *Lists of Values to Update for Order Management*

Note: Tasks required to integrate your Siebel Business Application with back-office applications are not included in this chapter. For information about integration tasks, see *Integrating Order Management with Back-Office Applications*.

Roadmap for Setting Up Order Management

There are some tasks that all users perform to set up order management, and there are some additional tasks that you might need to perform depending on your business model.

Required Tasks

To set up order management, all users perform the following tasks:

1. *Activating Workflows for Order Management*. Some order management functionality is based on Siebel Workflows. You must activate these workflows.
2. *Setting Up Accounts and Contacts for Order Management*. Accounts are used in many quote and order entry tasks, such as selecting the billing and shipping address. When you create accounts, enter the contacts associated with each account, which will be the contacts that are available to select in a quote or order for that account.
3. *Setting Up Salespeople*. Set up sales people as employees and use Sales Order views to associate them with teams.
4. *Setting Up Products, Pricing, and Catalogs*. To make products available in quotes and orders, you must define products, define price lists that assign prices to products, create product catalogs, and make the catalogs available to users.
5. *Setting Up Shipping Information*. You can define shipping carriers, shipping methods, shipping terms, and set up shipment freight calculation.
6. *Setting Up Tax Calculations*. You can integrate your Siebel Business Application with an external application to calculate taxes due on a quote or an order.
7. *Setting Up Payment Terms and Other Payment Factors*. You can customize payment terms, payment methods, accepted credit card types, and other payment-related factors. Credit card processing is handled through implementation with a third-party payment processing application.

8. *Setting Up Integration for Order Management*. You must set up integration with third-party applications used for checking credit. You must activate workflows used for integration of quotes and orders with back office applications. For information about setting up integration and activating workflows, see *Integrating Order Management with Back-Office Applications*.

Optional Tasks

In addition, depending on your business model, you might have to perform some of the following tasks to complete setup of order management.

1. *Setting Up Inventory Locations*. If your business model allows orders to specify that products are shipped from a particular inventory location, then you must set up inventory locations.
2. *Setting Up Order Types*. If you want to use different order types than the default types, then you must set up order types.
3. *Defining Order Statuses*. If you want to use different statuses types than the default types, then you must set up order statuses and use the state model to set up rules for modifying orders with each status.
4. *Creating Product Selection Favorites*. You can create groupings of products that your users will frequently add to orders.
5. *Creating Activity Plan Templates*. You can create templates of activities that must be carried out when particular products are ordered. For example, you can indicate all the installation activities required for a particular order.
6. *Letting Users Create Quotes and Orders from Accounts and Contacts*. There are additional workflows you can activate to allow users to create quotes and orders from the Accounts and Contacts screens.
7. *Setting Up Availability to Promise*. You can integrate your Siebel Business Application with a back-office application to check availability of certain products.
8. *Customizing the Sales, Quote, and Order Processes*. You can customize the processes that employee end users will follow to order products and services. You can use Siebel Business Processes to automate many of the steps in your business processes, and you can customize the predefined business processes.
9. *Setting Up Asset-Based Ordering*. If you want to use asset-based ordering, then you must perform additional setup tasks.
10. *Setting Up Order Cancellation for Non-English Implementations*. If your implementation is in a language other than English, then you must modify the rule for order cancellation.
11. **Setting up data validation.** If regulatory agencies or internal processes and procedures require you to verify the quality and accuracy of your transactions, then you can set up data validation. For more information, see *Siebel Order Management Infrastructure Guide* .
12. **Setting up recommendations.** You can display up-sell recommendations, cross-sell recommendations, and other recommendations to customers when they select products. For more information, see *Setting Up Product Recommendations*.

Activating Workflows for Order Management

Some order management functionality is based on Siebel Workflows. You must activate these workflows before using the product. For information about activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

Activate the workflows for the other order management functionality that you are using, as listed in the following table. For example, if you are using Siebel eSales, then activate the workflows for eSales.

Order Management Functionality	Where to Find Workflows
Order Management Interface	*Workflow and Method Reference for Order Management Interface*.
Pricing	Technical reference section of *Siebel Pricing Administration Guide* .

Order Management Functionality	Where to Find Workflows
Product Promotions	Product promotions section of *Siebel Pricing Administration Guide* .
Integration of Quotes and Orders with Back-Office Applications	*Activating Workflows for Integration of Quotes and Orders*.
Configurator	Configurator Workflow reference in *Siebel Product Administration Guide* .
Asset-Based Ordering	*Workflow Reference for Employee Asset-Based Ordering*.
Eligibility/Compatibility	Eligibility and Compatibility technical reference section of *Siebel Product Administration Guide* .
Recommendations	Product recommendations technical reference section of *Siebel Product Administration Guide* .
Siebel eSales	Workflow reference section of *Siebel eSales Administration Guide* .
Compound Product Validation	Compound product validation section of *Siebel Product Administration Guide* .
Third-Party Purchase Order Credit Checking Applications	*Activating the Workflows for Purchase Order Credit Checking Applications*.
Application Deployment Manager	See *Siebel Application Deployment Manager Guide* .

Setting Up Accounts and Contacts for Order Management

The way you set up accounts will affect how users enter shipping and payment information in quotes and orders.

When accounts have multiple locations, they can be set up in one of two ways:

- **Single account for all addresses.** You can create one account and enter multiple addresses entered on the Bill To/Ship To view tab.
- **Separate accounts, one for each address.** When you set up separate accounts for each address, you must create an account hierarchy, where one account is the parent account for the other locations.

Consider which account setup is better for your business, so that you can ship to and bill customers correctly. Consider how the account is implemented in your back office, so that you can keep information updated between the two applications.

The account setup you choose will impact the implementation of your organization's order management processes, as follows:

- **Single account for all addresses.** The account is the same in all places on the Payment and Fulfillment screens. You can select the appropriate addresses for billing and shipping. The address fields on quote and order default to the address specified as "Primary" in the Addresses view tab for the account.

- **Separate accounts, one for each address.** You can select different accounts as the bill-to and ship-to entities on the Payment and Fulfillment screens in order to ship to those addresses.

The contacts that you associate with accounts determine the contacts that you can select in quotes and orders. Make sure that you attach the contacts to accounts that you want to have available to users when they create orders.

Use the Account screen to enter accounts and attach contacts to them. For more information about entering accounts and their contacts, see *Siebel Applications Administration Guide* .

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Salespeople

Before salespeople can create quotes or orders, you must enter the salespeople as employees. For more information about entering employees, see *Siebel Applications Administration Guide* .

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Products, Pricing, and Catalogs

Before employees can create quotes and orders, you must:

- **Set up products.** Define the products that are being sold. If the products have attributes, then you can set up a class system to manage the attributes before defining the products. If you have customizable products that have other products as their components, then you must specify the components and design the selection pages that users will see.

For information about setting up products, see *Siebel Product Administration Guide* .

- **Set up price lists.** Set up price lists and assign prices to products. You can create multiple price lists if you sell the same products to different customers at different prices. You can create discounts, such as volume discounts and discounts for bundles of products. If you have Siebel Pricer, then you can create more advanced pricing rules using pricing models and pricing factors.

  For information about setting up price lists, see *Siebel Pricing Administration Guide* .

- **Set up product catalogs.** End users can select products for customers from product catalogs and add them to a quote or an order. When you create a catalog, you specify what products are included in it and arrange them in a hierarchy of categories. Once catalogs are set up, users can browse through categories to find products, and they can search for products using full-text search or parametric search.

  For information about setting up and administering catalogs, see *Creating and Managing Catalogs* .

- **Assign catalogs to employees.** To assign a catalog to an employee, you associate the catalog with an access group that the user is a member of. An employee is a member of an access group, if the employee is associated with a position, organization, division, account, or user list that is a member of the access group. When displaying a list of products, for example, in a parametric search, the employee will see only those products that are in catalogs that have been assigned to that employee.

  For information about defining access groups, see *Siebel Security Guide* .

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Shipping Information

You can set up shipping information in the following ways:

- Modify the lists of values used to select shipping information.
- Set up shipping charges using a number of factors, including the weight of the product, the shipping method, the shipping carrier, and the ship to and ship from location.
- Set up preferred carrier and shipping method for an account.

When end users create quotes or orders:

- They are able to select the options that you set up in the lists of values.
- The shipping charges that you set up are displayed for each line item and for the quote or order as a whole.
- The preferred carrier and shipping method for the account appear as the default carrier and shipping method.

This task is a step in *Roadmap for Setting Up Order Management*.

Modifying Lists of Values for Shipping

You can modify the following lists of values that are used to select shipping information:

- **Carriers.** This field specifies what carrier is used to deliver items in an order. Without configuration, the available carriers include Airborne, DHL, Emery, Federal Express, UPS, US Mail. Make sure the list of values contains the carriers you want to use. In the Administration - Data screen, List of Values view, carriers are listed in records where the Type field has the value FS_CARRIER.
- **Shipping methods.** The Shipping Method field is used primarily to identify how quickly the shipment will occur, and how long delivery will take. The shipping method impacts the shipment charges. Without configuration, the available methods are Next Day, 2 Day Service, and Ground. You can add other methods by adding them in the list of values. For example, you might want Air, Train, or Ship as a method. In the Administration - Data screen, List of Values view, shipping methods are listed in records where the Type field has the value CARRIER_PRIORITY.
- **Shipping terms.** The Shipping Terms field is used to specify the relationship between payment and shipping. Without configuration, the available shipping terms are Due, FOB, TBD, No charge. You can add other terms by adding them in the list of values list. In the Administration - Data screen, List of Values view, shipping terms are listed in records where the Type field has the value FREIGHT_TERMS.
- **Shipment freight calculation.** The result of the shipment freight calculation shows in an order in the Shipping Charges field, in Line Items Totals form. Shipping is calculated at the line level, and is then rolled up for all line items and displayed in the Totals form.

For more information about changing lists of values, see *Siebel Applications Administration Guide* .

Setting Up Shipping Charges

To set up shipping charges, first you define shipping zones, and then you define the charges for each zone. A shipping zone includes both a ship-from location and a ship-to location.

These shipping charges apply to employee applications. For information about setting up shipping charges for Siebel eSales, see *Siebel eSales Administration Guide* .

To set up shipping charges

1. Navigate to the Administration - Order Management screen, and then the Shipping view.

2. In the Shipping Zone list, add a new record and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Ship Carrier	Select a carrier.
Ship Method	Select a shipping method.
Ship From Minimum	Enter the smallest number of the range of postal codes where the ship-from addresses of this zone are located.
Ship From Maximum	Enter the largest number of the range of postal codes where the ship-from addresses of this zone are located.
Ship From Country	Select the country where the ship-from addresses of this zone are located.
Ship To Minimum	Enter the smallest number of the range of postal codes where the ship-to addresses of this zone are located.
Ship To Maximum	Enter the largest number of the range of postal codes where the ship-to addresses of this zone are located.
Ship To Country	Select the country where the ship-to addresses of this zone are located.
Shipping Zone	Enter the code for this shipping zone.

3. Add a new record to Shipping Rate list for each shipping charge for this zone, and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Ship Carrier	Select a carrier.
Ship Method	Select a shipping method.
Shipping Zone	Enter the code for this shipping zone.
Minimum Weight	Enter the minimum weight for this charge.

Field	Comments
Maximum Weight	Enter the maximum weight for this charge.
Cost	Enter the cost for shipping products of this weight range in this zone.

Setting Up Default Carriers for Accounts

If you set up the preferred carrier for an account, then this carrier appears as a default when users create quotes and orders for this account.

To set up the default carrier for an account

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the desired account.
3. Click the More Info view tab.
4. In the More Info form, complete the necessary field, described in the following table.

Field	Comments
Preferred Carrier	Select the default carrier for this account.

Setting Up Tax Calculations

Tax calculations can be done manually, or through the Tax Calculator business service, which makes a call to Taxware, provided by a third-party vendor.

This task is a step in *Roadmap for Setting Up Order Management*.

Entering Taxes Manually

Users can enter tax information directly in the tax fields, in the Payments view in an order.

In the Payments form, the user either enters the tax rate or selects a tax exemption code and tax exemption ID for customers who are exempt from taxes.

If users enter tax information manually, then the only required setup is to define the appropriate tax exempt codes in the list of values.

In the Administration - Data screen, List of Values view, tax exempt codes are listed in records where the Type field has the value GLOBAL_TAX_EXEMPTION.

For information about changing lists of values, see *Siebel Applications Administration Guide* .

Using a Third-Party Application to Calculate Taxes

You can use Taxware, an external third-party application, to calculate taxes on quotes and orders. Clicking the Calculate Taxes step button the Tax Calculator business service, which makes a call to the third-party application. The Calculate Tax business service sends information about each line information, such as the product and the ship-to address, which allows Taxware to calculate the amount to be charged for each line items. The total of the tax on all the line items is displayed as the tax for the entire order.

When using a third-party application, the end user must make sure that the ship-to addresses are correct and then click the Calculate Taxes button on the Line Items Totals form. The amount of the tax will appear on the Total form, and the tax rate will appear in the Payments form. Because ship-to addresses can vary for different line items in an order, the tax rate can also vary for different line items. (For details on the user's experience of this feature, see *Calculating Taxes*.)

> **Note:** For products with components, taxes may be calculated either on the individual components, or on the extended price for the product with components. The method used is determined by the Tax Subcomponent check box in the Product Administration screen. For more information, see *Siebel Product Administration Guide* .

Tax-exempt information can be entered in your Siebel Business Application on the Payments view tab in a quote or order.

However, customers can also file their tax-exempt information with a tax provider for validation. When the third-party tax application contains tax-exempt information, that information is updated when the user clicks Calculate Taxes.

If you want to hide the tax-exempt fields from users, then you must customize the Payments form in Siebel Tools. There are three tax-exempt fields: Tax Exempt Flag, Tax Exempt Reason, and Tax Exempt Certification ID.

For details about setting up and using a third-party taxation application, see *Siebel eSales Administration Guide* .

Setting Up Payment Terms and Other Payment Factors

When you set up order management, consider the following payment factors:

- *Defining Payment Terms*
- *Credit Card Processing*
- *Credit Card Authorization and Verifications*
- *Adding New Credit Card Types*
- *Defining New Methods of Payment*
- *Define a Mailing Address for Check Payment*
- *Creating Payment Profiles*

This task is a step in *Roadmap for Setting Up Order Management*.

Defining Payment Terms

Payment terms are used to specify when payment is due, in relation to the order date or ship date. For example, payment terms might include terms such as Net 10, Net 20, Net 30, Net 60. Users select them on the Payments view tab of a quote or order.

To define a new payment term

1. Navigate to the Administration - Data screen, Orders, and then the Payment Terms view.
2. In the Payment Terms list, add a new record.
3. Enter a name for this payment term and complete appropriate fields to define the payment terms.

To change settings for current payment terms

1. Navigate to the Administration - Data screen, Orders, and then the Payment Terms view.
2. In the Payment Terms list, select the appropriate record and make the necessary changes.

Credit Card Processing

Credit card processing is done through integration with a third-party processing application. For details about how to set up this feature, see *Siebel eSales Administration Guide* .

> **Note:** Integration for credit card processing is provided with Siebel Order Management. If you want to integrate with a back-office application to provide processing of other payment methods, then you can use Siebel Tools to set up new business services to provide this functionality. For more information, see *Using Siebel Tools* .

Credit Card Authorization and Verifications

Your Siebel Business Application comes preconfigured with business services for some third-party services. If you use a different third-party service, then you might create special Business Services for card authorization and verification.

You can change the use of the following fields:

- **Authorization Code.** The Authorization Code field does not populate automatically. If your back-office application sends an authorization code, then you may want to create a business service to interface with Siebel Order Management.
- **Card Verification #.** The credit card verification number is a four-digit unique identifier that is often printed on the back of a credit card. It is used to reduce fraud because only the person holding the card must know it.

> **Note:** To avoid exposing sensitive data, when implementing credit card verification, consider how to clear the Card Verification # from your database.

Adding New Credit Card Types

Siebel Order Management provides the following credit card types without configuration: Visa, Mastercard, Discover, and American Express.

You can add more credit card types by entering them as values in the List of Values list. Add a record with the Type PAYMENT_TYPE CODE, and add a value in the Order field to associate it with a payment method.

In the Order field, the number range for credit cards is 10 to 20. Values 11 through 14 are preconfigured for Visa, Mastercard, Discover, and American Express. You can use a value between 15 and 19 to enter in the Order field for the new credit card, so all the credit cards appear together.

> **Note:** If you need to add a payment type for a payment method other than credit card, then use the following value ranges in the Order field: Check method: 20-29 Stored value method: 40-59 If you create another payment method which requires payment types, then use the range 60-80 for the new method.

To add new credit card types

1. Navigate to the Administration - Data screen, and then the List of Values view.
2. In the List of Values list, add a new record, and complete the necessary fields, described in the following table.

Field	Comments
Type	Enter PAYMENT_TYPE_CODE.
Display Value	Enter the name of the credit card that is displayed to the user.
Order	Enter a value between 15 and 19 to associate this record with a credit card payment method.

3. Navigate to the Administration - Data screen, and then the List of Values Explorer view.
4. In the List Of Values - Type list, search for the type, PAYMENT_METHOD_CODE.
5. In the Explorer pane, expand the Values folder under PAYMENT_METHOD_CODE.
6. In the Explorer pane, expand the Credit Card folder, and expand the Child List of Values folder under the Credit Card folder.
7. In the Child List of Values list, add a new record, and complete the necessary fields, described in the following table.

Field	Comments
Type	Enter PAYMENT_TYPE_CODE.
Name	Enter the same name that you entered in the Display Value field in Step 2.

Defining New Methods of Payment

Siebel Order Management comes preconfigured with the following payment methods: cash, credit card, purchase order, stored value, wire transfer, and check. Each method has a predefined set of fields to support the information needed for this method of payment.

> **Note:** The Orders screen allows more payment method information to be captured and displayed than the Quotes screen. For example, the stored value payment method is not available in a quote, and the fields available for the other methods are limited.

You can add more payment methods. To do so, you need to perform the following steps using Siebel Tools and the Administration - Data screen, List of Values view:

1. In the Business Component *Payments*:
   a. Add a Business Component User Property, such as

      Name: Payment Method Type 4

Value: MY Method

b. Change the Calculated Value of the Field Payment Type Read Only Flag. For example, change the value to:

   If ([Payment Method LIC] = "Credit Card" OR [Payment Method LIC] = "Check" OR [Payment Method LIC] = "Stored Value" OR [Payment Method LIC] = "Purchase Order" OR [Payment Method LIC] = "MY Method", "N", "Y")

2. In the Administration - Data screen, List of Values view:

   a. Add the new payment method value to LOV Type PAYMENT_METHOD_CODE. For example, add:

      Display Value: LIC: MY Method

      Order: 17

   b. Add the corresponding payment type values to LOV Type. For example:

      Display Value: LIC: MY Method Type 1

      Order: 81 *(= number of the Business Component User Property * 20 + 1)*

      Display Value: LIC: MY Method Type 2

      Order: 82 *(= number of the Business Component User Property * 20 + 2)*

After you have done this configuration, when the user create a new Payments record, the user can:

- Select the new payment method.
- Find the corresponding payment type values in the pick list on field Payment Type.

You can also require that the user complete particular fields for each payment method. You can add required fields to the Payments business component in Siebel Tools. You must also add an additional toggle applet to Applet: Payment Detail - Credit Card, to allow the user to enter these values.

Define a Mailing Address for Check Payment

Customers paying by check often need to know where to send the payment. A field in the Payment Detail form for check payments provides a convenient place to display this type of information, so the sales representative can communicate it to the customer. This field is labeled Check Mailing Address in the Payment Detail - Check form in the Order Entry - Payment view.

You can enter the information that must display in this field in Siebel Tools, in the Payments business component, in the user property Check Mailing Address.

Creating Payment Profiles

- Credit Card
- Purchase Order
- Check
- Wire Transfer

After you have selected a payment method, you enter the appropriate details for that method. For example, if you selected Credit Card as the method, then you enter details such as the name of the card holder, the credit card number, and the expiration date.

When salespeople create quotes or orders, they can use the Payment Details view of the Quote or Order screen to select the payment method for the customer. For example, if they select a credit card, then all the information for that credit card is automatically entered in the quote or order, so the salesperson does not have to enter it manually.

Note: You may have to use the site map to navigate to the Payment Profile views.

To create payment profiles

1. Navigate to the Account screen, Accounts List, and then the Payment Profiles view or the Contact screen, Contacts List, and then the Payment Profiles view.
2. In the Account or Contact list, select the desired account or contact.
3. Add records to the Payment Profiles list and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Primary	Select this check box if this profile is the customer's primary method of payment.
Profile Name	Enter a name for the payment profile.
Payment Method	Select the payment method. Options are Credit Card, Purchase Order, Check, Wire Transfer.
Expiration Date	Enter the date when this payment method expires. Salespeople cannot use it after this date.
Status	Select a status for this payment method. Options are: o **Inactive.** The payment method has not yet been activated. o **Active.** The payment method is active. This status must be selected to make the payment method available to salespeople. o **Suspended.** The payment method is no longer active.

4. In the details form under the list, enter the appropriate details for this payment method.

   The details form changes depending on the value you selected in the Payment Method field. For information about the fields in each detail form, see *Payment Fields for Payment Methods*.

Setting Up Inventory Locations

In some businesses, when users inquire about availability of a product, they can request a particular inventory location, and when products are reserved, they are promised from a particular source.

Inventory locations are used to identify where products are stored and the source from which the product is fulfilled.

An inventory location may be a warehouse, a trunk, or a field office, or it may be virtual.

For more information about setting up inventory locations and working with inventory, see *Siebel Field Service Guide* .

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Super Users for Product Administration

A product administrator, who is defined as a super user, can do the following:

- Unlock or edit objects that have been locked by other product administrators or users from the UI.
- Import product models with object locked by others.
- Use REST APIs to import product models.

Examples of versioned objects include a product, class, attribute definition, signal or variable map. The main advantage to being able to unlock versioned objects is so that someone else can work on them, thereby minimizing downtime and delays in time-to-market.

Configuring Super Users

Defining a product administrator as a super user involves assigning the "Product Authoring Super User" responsibility to the user, as shown in the following procedure.

Once a user is assigned this responsibility, then that user's ISSSuperUser profile attribute is set to Y to indicate that they are a super user. If a user's ISSSuperUser profile attribute is set to N, then this indicates that the user is *not* a super user.

A super user can unlock objects, modify objects and edit Workspaces, and release objects locked by other users – for more information, see *Unlocking Objects*.

To configure super users for product administration

1. Navigate to the Administration - Application screen, then the Responsibilities view.
2. Query for and select the *Product Authoring Super User* responsibility.

   Note: If this responsibility doesn't exist, create one.

3. Select the Responsibilities tab, and then do the following:
   a. In the User applet, add (product administrator) users to the responsibility as required.
      For example, add the following users: SADMIN or any other users you want to be part of the responsibility.
   b. This responsibility does not require any views.

Unlocking Objects

An object is locked if the Locked Flag check box is selected for the object. The Locked By and Locked Date fields indicate who locked the object and the date/time that the object was locked.

Super user can unlock an object, as shown in the following procedure. Alternatively, super user can Release New Version from Workspace Project view, Contents applet.

Once an object is unlocked, the current user or product administrators can lock the object. Once the object is locked, the user can modify the child entities. After completing the testing, the user can unlock the object by clicking the Release button.

To unlock an object

1. Login as super user.
2. From Site map, navigate to the object you want to unlock:
    - Administration - Product -> Product Definitions for product
    - Administration - Product -> Product Classes for class
    - Administration - Product -> Attribute Definitions for attribute
    - Administration – Order Management -> Signals for signal
    - Administration – Order Management -> Variable Maps for variable maps
3. Query and select the record you want to unlock.
4. Deselect the Locked Flag check box on the object.

   Doing this clears the Locked By field.
5. (Optional) Select the Locked Flag check box to lock the object.

   The super user's name appears in the Locked By field – indicating that they are now the administrator of the object.
6. Modify the object as required, and then click Release to release the changed object.

Setting Up Integration for Order Management

You can integrate order management with third-party applications. For information about setting up integration, see:

- *Integrating Order Management with Back-Office Applications*
- *Integrating Order Management with Third-Party Product Availability Applications*
- *Integrating Order Management with Third-Party Credit-Check Applications*
- *Integrating Order Management with Third-Party Taxation Applications*

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Order Types

Siebel Orders and Siebel Order Management initially define nine types of orders, as listed in the following table.

You can define additional order types to meet your needs, or you can edit the existing list of values.

This task is a step in *Roadmap for Setting Up Order Management*.

Type of Order	Typical Use
Internal order	An order used to replenish stock or move inventory among inventory locations.

Type of Order	Typical Use
Purchase order	An order used to buy parts from external vendors.
RMA (Return Material Authorization) Advance Exchange	An order used to handle customer returns that require receiving and immediate shipping of exchange parts, before receiving the returned part.
RMA Repair Return	An order used to handle customer returns that require receiving, repairing, and shipping back to customers.
RMA Return	An order used for receiving inbound returns from customers.
Repair order	An order used for ordering, shipping, and receiving parts to be repaired by a third party.
Sales order	A customer order that is owned by, processed by, and credited to a sales business. Typically, selling new finished goods to customers, normally from manufacturing inventory.
Service order	A customer order that is owned by, processed by, and credited to a service business. Typically, a request from a customer for service on existing products, including replacement and repair of parts.
Web Order	A sales order created at a Web site or requested directly over the Internet.

Each order type is either a sales order or a service order. The order type is determined by the Type Code associated with it in the Data Administration screen, Order Type view.

Each order type has action types associated with it. Action types are used primarily in Siebel Field Service. For more information about action types, see *Siebel Field Service Guide* .

Defining a New Order Type

Use this procedure to define a new order type.

To define a new order type

1. Add the necessary order types to the List of Values list.

   In the Administration - Data screen, List of Values view, records for order types have FS_ORDER_TYPE in the Type field.

   For more information about managing lists of values, see *Siebel Applications Administration Guide* .
2. Navigate to the Administration - Data screen, and then the Order Types view.
3. In the Order Types list, add a new record.
4. In the Order Type field, select a value from the list.

Adding or Changing the Display Value of Order Types

To configure the product using Siebel Tools to add new order types or to change the display value of Order Types, you must perform additional configuration steps for the homepage to work correctly. The following procedures show you how to configure the homepage.

To add new order types

1. In Siebel Tools, query for *Order Entry - Orders Home Add Virtual Form Applet*.
2. Under Drilldown Object, add an entry for the new Order Type.
3. Under the Dynamic Drilldown Destination of the original Drilldown Object, add an entry to point to the newly created Drilldown Object.
4. Repeat Step 1 to Step 3 for *Order Entry - Orders Home Search Virtual Form Applet*.
5. Repeat Step 1 to Step 3 for *Recent Record Order Entry - Orders List Applet*.

To change the display value of an order type

- In the Order Types view of the Administration - Data screen, update the order type to point to the new LOV display value.

Defining Order Statuses

Siebel Orders and Siebel Order Management come with a number of preconfigured order statuses. You can create additional statuses, if you need them, or edit the existing list of values.

See the table in *Lists of Values to Update for Order Management* for the Type field in the Administration - Data screen, List of Values view that you use to work with this list of values.

For more information about managing lists of values, see *Siebel Applications Administration Guide* .

When you create a new order status, you can determine what fields can be changed when an order has that order status. Often these rules mirror back-office constraints. For example, the Ship to Address cannot be changed after the order has been picked up or shipped. Similarly, order quantities and details cannot be changed after an order has been submitted.

To specify which records can be changed depending on order status, use the state model. For more information about working with the state models, see *Siebel Business Process Framework: Workflow Guide* .

This task is a step in *Roadmap for Setting Up Order Management*.

Creating Product Selection Favorites

Product selection favorites are combinations of products that have been saved for easy reuse. The entire favorite can then be added to a quote or order, instead of adding each product individually.

An administrator can set up product favorites in advance and make them publicly available. Also, individual users can create private favorites of frequently used products.

This task is a step in *Roadmap for Setting Up Order Management*.

To create a product selection favorite

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

2. In the Product Favorite list, add a new record, and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Name	Enter a name for the favorite.
Type	Select Private to make this favorite available only to yourself, or public to make it available to all end users.

3. In the Product Favorite Items list, add a record.

   The Pick Product dialog box appears.
4. In the dialog box, select one or more products and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Order Qty	Enter the quantity of this product in the favorite.
Product	Displays the name of the product. If you enter a value for Order Qty, then the product is included in the favorite.

5. In the dialog box, click OK.

   The products you selected are added to the Product Favorite Items list.

Creating Activity Plan Templates

In Siebel Order Management, activity templates are useful to itemize the activities that must occur after an order for a particular product is placed or filled. For example, when a product requires installation, you want to make arrangements for the installation when the order is placed.

Activity templates are typically created by an administrator, so that they are available when a sales administrator or call center agent creates an order.

The following steps describe how to create an activity template. For information on how to associate a template with an order, see *Associating an Activity Plan or Activity with an Order*.

This task is a step in *Roadmap for Setting Up Order Management*.

To create an activity plan template

1. Navigate to the Administration - Data screen, and then the Activity Templates view.

2. In the Activity Plan Templates list, add a new record and complete the necessary fields. Some fields are described in the following table.

Field	Comments
Name	Enter a name for the template.
Type	Choose the type Order Entry - Line Items.
Sales Stage	Select the sales stage when this template is available.
Auto Trigger	Select this check box to provides the activities from this template to a sales stage, when the sales stage is set for an opportunity
Public	Select this check box to allow others not on your team to use this activity plan.

3. Click the Activity Template Details view tab.
4. In the Activity Template Details list, add a new record for each required activity, choose an activity type, and then complete the other necessary fields.

Letting Users Create Quotes and Orders from Accounts and Contacts

There are workflows you may activate if you want users to be able to create quotes and orders from the Accounts screen and the Contacts screen.

For more information, see *Setting Up Integration for Order Management*.

This task is a step in *Roadmap for Setting Up Order Management*.

Setting Up Availability to Promise

You can integrate your Siebel Business Application with a back-office application to check availability of certain products.

For more information, see *Integrating Order Management with Third-Party Product Availability Applications*.

This task is a step in *Roadmap for Setting Up Order Management*.

Customizing the Sales, Quote, and Order Processes

You can customize the sales, quote and order process by using:

- *New or Modified Workflows*
- *Configuration with Siebel Tools*

This task is a step in *Roadmap for Setting Up Order Management*.

New or Modified Workflows

Some actions in the Quote and Order screens are set up using Siebel Workflows, such as the following:

- To allow users to create quotes and orders from the Account screen and the Contact screen, you enable a set of workflows.
- To use asset-based ordering, you enable a set of workflows.
- Some of the ASI-based functions that allow you to integrate with back-office applications also use workflows. For example, submitting an order activates a workflow, as does performing a credit check.

You can customize these workflows to carry out the specific processes required for your business. You can also create additional workflows as needed to support your business process.

Here are some changes you might make to customize the order management process using new Siebel Workflows:

- **Quote verification.** By default, users can verify a quote manually by choosing the Verify menu option from the menu. You can create a workflow that verifies quotes automatically.
- **Approval requirements.** Siebel Quote and Order screens include fields for required approvals which are used manually by default. For example, a sales representative might be required to get a supervisor's approval for orders greater than $500. Without configuration, the sales representative would need to put a manager's name in the Approved By field. You can create a Workflow that would automatically route an order to the supervisor for approval when the amount exceeded $500. You would use a combination of Siebel Business Process Designer and Assignment Manager to route the approval to the appropriate supervisor.
- **Customer acknowledgements.** You can create Workflows to automatically send order confirmations to customers. When the status of an order changes, the Workflows send the appropriate messages to the customer. Siebel eSales includes preconfigured workflows that send acknowledgements to customers, which you can use models to create similar workflows for use in order management. The eSales workflows are:
    - Send Order Accepted Email (eSales). Sent to all users who place an order successfully.
    - Send Order Awaiting Approval Email (eSales). Sent to the purchasing manager for an account when an order with a status of Awaiting Approval needs to be reviewed.
    - Send Order Failed Email (eSales). Sent to a user when an auction is closed and turned into an order, but the credit card used cannot be authorized.
    - Send Order Rejected Email (eSales). Sent to a corporate user when the purchasing manager or delegated customer administrator has rejected that corporate user's order.
    - Send Order Shipped Email (eSales). Sent to a user when that user's order has been shipped.

You can also customize workflows that are shipped with the product, in order to add or remove steps or change the order in which steps are performed. For a reference to these workflows, which will help you understand them so you can modify them, see *Workflow Reference for Employee Asset-Based Ordering*.

For more information about creating and customizing Workflows, see *Siebel Business Process Framework: Workflow Guide* .

Configuration with Siebel Tools

You can use Siebel Tools to configure Siebel screens and views to match the processes used in your company. For example, you can define the default view that users see when they log in, so that users start on the screen that they use the most frequently. You can also add, remove, or change the names of fields.

You also can use Siebel Tools to change user properties to customize some order management capabilities.

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

Component Quantity Edits

Component quantity edits are an example of one possible user property change you might want to make in Siebel Order Management.

Typically, a user changes the quantity of a component in the configuration session that starts after clicking the Customize button in the Line Items list. In this case, Siebel Configurator checks all configuration rules to make sure that the change in quantity is appropriate.

Your company can decide to allow changes to component quantities directly in the Line Items list in the Quote or Order by setting the user property Configuration ReadOnly Fields in Siebel Tools.

When this user property has been set, the Qty field in line item for components of products with components is editable. If the Qty field is read-only, then this user property has not been set.

Other User Properties in Order Management

Some other user properties that you might want to change to customize order management functionality are listed in the following table.

User Property	Business Component or Applet	Function
Check Minimum and Maximum Price	Quote Item, Order Entry - Line Items, FS Agreement Item	Checks minimum and maximum price when making manual adjustments to line item.
Skip Loading Default Cfg Instance	Quote Item, Order Entry - Line Items, FS Agreement Item	Sets to *Y* to skip the loading if the default configurable product instance when adding a new configurable product to a quote.
Spread Discount - Use Root Product Type	Quote Item, Order Entry - Line Items, FS Agreement Item	If set to *Y* and spreading a discount by type, spreads on components if the root type is the type selected. Otherwise, spreads on the type of component only if the type of component itself matches the selected spread type.
Add Items Limit To Price List	Internal Product Popup	Limits the products that display in the Add Items dialog box to records contained in the price list.
Configuration ReadOnly Fields	Quote Item List Applet, Order Entry - Line Item List Applet, FS Agreement Item List Applet	Removes the entries for Quantity and Quantity Requested to make the quantity of configurable product components editable in the list applet.

User Property	Business Component or Applet	Function
Minimum CC Txn Amount	Payments business component	Sets the minimum amount payable by credit card when entering a payment line.
Minimum Cash Txn Amount	Payments business component	Sets the minimum amount payable by cash when entering a payment line.
Minimum Check Txn Amount	Payments business component	Sets the minimum amount payable by check when entering a payment line.
Minimum Purchase Txn Amount	Payments business component	Sets the minimum amount payable by purchase order when entering a payment line.
Minimum Stored Txn Amount	Payments business component	Sets the minimum amount payable by stored value when entering a payment line.
Minimum Wire Txn Amount	Payments business component	Sets the minimum amount payable by wire transfer when entering a payment line.
Check Mailing Address	Payments business component	Mailing address for check.

Setting Up Asset-Based Ordering

Asset-based ordering is available with the Siebel Order Management module. Asset-based ordering adds the following features to the application:

- The Account Summary and Installed Asset views in the Accounts screen.
- The following buttons that are specific to asset-based ordering:
  - In the Account Summary view, New, Modify, and Disconnect
  - In the Installed Assets view, Suspend and Resume
  - In the Order screen, in the Quote and Order Line Items view, Submit, Auto Asset, and Customize

Note: Note that the Submit button on the Order screen and the Customize button on the Quote and Order Line Items view also work when asset-based ordering is turned off. From the user's perspective, the action triggered by clicking these buttons is the same. However, when asset-based ordering is turned on, the buttons perform these actions by running a workflow.

To set up asset-based ordering, perform the following tasks:

1. *Setting the Server Component Parameter*. Set the server component parameter *Order Management - Enable Asset Based Ordering* to True.
2. *Making Changes to the Asset-Based Ordering Workflows*. If necessary, change the asset-based order workflows to reflect your business needs.

3. *Activating the Workflows*. Activate the asset-based ordering workflows in the Business Process Designer screen.
4. *Configuring Asset-Based Ordering*. You can configure asset-based ordering using Siebel Tools.
5. *Setting the Default Value for Auto-Assets*. You can set the default value used when users create auto-assets.

This task is a step in *Roadmap for Setting Up Order Management*.

Setting the Server Component Parameter

You must change set a server component parameter to True to use the asset-based order workflows associated with the Configure and Submit buttons.

Use the following procedures to set the parameter for the Web Client (used when you are connected to the Web) and for the Mobile Web Client (used when you are not connected to the Web, for example, to work remotely on a laptop).

To set the server component parameter for the Web Client

1. Navigate to the Administration - Server Configuration screen, Enterprises, and then the Component Definitions view.
2. Use a query to find the object manager for the application in which you want to use asset-based ordering. For example, you might query for Call Center Object Manager (ENU).
3. Click the Parameters view tab.
4. Query for the parameter *Order Management - Enable Asset Based Ordering*.
5. Set the Current Value to True and save by stepping off the record.
6. Perform Step 2 through Step 5 for the object manager for each application in which you want to use asset-based ordering.
7. Restart the object managers for which you made the change.

To set the server component parameter for the Mobile Web Client

1. Use any text editor to add the .cfg file for the application.
2. In the .cfg file, add the following section:

```
[InfraObjMgr]

AssetBasedOrderingEnabled = TRUE
```

Making Changes to the Asset-Based Ordering Workflows

You can adjust the behavior of asset-based ordering by making changing workflows.

All workflows for asset-based ordering begin with the characters SIS OM. You can access all of these workflows by querying for SIS OM* in the Business Process Administration screen, Workflow Processes view.

For more information about the asset-based ordering workflows, see:

- *Using Asset-Based Ordering* describes the user's view of how asset-based ordering functionality works by default.
- *Workflow Reference for Employee Asset-Based Ordering* describes the asset-based ordering workflows.

For information about customizing workflows, see *Siebel Business Process Framework: Workflow Guide* .

Activating the Workflows

You must activate the workflows that provide asset-based ordering features to make those features available:

- All workflows that begin with *SIS OM*

- In addition, activate the following workflow: CUT Quote To Order PMT Enhancement - Sub Process

For information about activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

Configuring Asset-Based Ordering

You can use Siebel Tools to configure Siebel screens and views to match the processes used in your company. For more information about Siebel Tools, see *Using Siebel Tools* .

Setting the Default Value for Auto-Assets

When the user clicks the Auto-Asset button, an asset is created based on order line items. You can configure the Siebel business application to define default values for the fields of the auto-assets.

The SIS OM Asset integration object (IO) is used to create the asset. The following two integration components in this IO are based on S_ASSET table

- **Header.** Based on Asset Mgmt - Asset - Header business component.
- **Line items.** Based on Asset Mgmt - Asset business component.

If you define default values for these business components, then the default values are also used in the corresponding fields of the auto-asset.

Setting Up Order Cancellation for Non-English Implementations

If the implementation is in a language other than American English (ENU), then you must change some order validation rules so they use the non-English language. If you do not do this setup, then users who try to cancel orders get error messages saying that the cancellation failed and that payment is invalid because a ground for cancellation is required.

To set up order cancellation for Non-English Implementations

1. Navigate to Admin - Data Validation screen, and then the Rule Sets view.
2. In the Validation Rule Set list, query for the O2C Order Validation rule set.
3. Drill down on the O2C Order Validation rule set.
4. In the Rules list, query for the rule named Payment or Cancel reason not empty.
5. In the Payment and Cancel Reason not empty rule, change the Apply To field, change the value All Records to the corresponding value in the non-English list of values.
6. Click Activate.
7. Restart the services.

Lists of Values to Update for Order Management

Many of the changes you make while setting up order management are implemented by adding to or changing the items available in drop-down lists throughout the order management screens.

The lists of values that relate to order management are shown in the following table. The third column shows the value of the Type Field in the Administration - Data screen, List of Values view that determines what values are included in each of these lists of values.

For more information about maintaining lists of values, see *Siebel Applications Administration Guide* .

View	Field	Type Field in LOV Admin
More Info	Order Type	FS_ORDER_TYPE
More Info	Order Status	FS_ORDER_STATUS
More Info	Hold Reason	SAP_SO_HEADER_DELIV_BLOCK
More Info	Return Reason	RETURN_REASON
More Info	Priority	FS_INVLOC_ORDPRI
More Info	Default Discount %	EST_HW_DISC
Line Items/Totals	Shipping Method	CARRIER_PRIORITY
Line Items/Totals	Shipping Terms	FREIGHT_TERMS
Line Items/Totals	Carrier	FS_CARRIER
Line Items/Line Detail	Shipping Method	CARRIER_PRIORITY
Line Items/Line Detail	Unit of Measure	UNIT_OF_MEASURE
Line Items/Line Detail	Hold Reason	SAP_SO_HEADER_DELIV_BLOCK
Line Items/Line Detail	Product Status	FS_PRODINVCAT_STATUS
Fulfillment	Available Status	ATP_STATUS
Fulfillment	Return Reason	RETURN_REASON
Fulfillment	Priority	FS_INVLOC_ORDPRI
Fulfillment	Time to Reserve Unit of Measure	PERIOD_UOM
Payments	Tax Exempt Reason	GLOBAL_TAX_EXEMPTION
Payment Detail	Payment Method	MMOP_METHOD_TYPE
Payment Detail	Payment Type	PAYMENT_TYPE_CODE
Payment Detail	Payment Status	MMOP_PAYMENT_STATUS

4 Integrating Order Management with Back-Office Applications

Integrating Order Management with Back-Office Applications

This chapter describes the setup required to integrate Siebel's order management applications with back-office applications. This setup is required to send and receive information when end users submit quotes and orders, check the status of an order, request information about availability to promise order items, and check the credit status for a customer. All of these functions use the Application Services Interfaces (ASIs).

This chapter covers the following topics:

- *Integration Strategy for Order Management*
- *Smart Part Numbers and Order Management Integration*
- *About ASIs for Order Management*
- *Setting Up Quote and Order ASIs*
- *Process of Integrating Quotes and Orders with Back-Office Applications*
- *ASI Workflows for Quotes and Orders*

For more information about Application Services Interfaces, see *Siebel Application Services Interface Reference* .

For more information about how to plan to integrate order management into your other applications and processes, see:

- *Overview: Siebel Enterprise Application Integration*
- *Integration Platform Technologies: Siebel Enterprise Application Integration*

Integration Strategy for Order Management

If you integrate Siebel Business Applications with back-office, accounting, and other applications, then you will need to develop an integration strategy. The integration tasks you must consider depend on the business needs of your company. Some tasks you may want to consider are in the following list.

> **Note:** You can integrate some functionality using Oracle Application Integration Architecture (AIA), which does not require using the strategy outlined here. For more information, see *Process of Integrating Order to Cash* and *Integrating Distributed Order Orchestration*.

- **Determine the application of record for each of the key data elements.** Some of the data elements you must consider are:
  - Item masters (product)

    - Configuration and pricing rules
    - Customer master

- **Decide how to replicate data.** You can either replicate data between your Siebel Business Application and your back office application using batch replication at regular intervals (typically daily), or use real-time replication. If you use real-time replication, then you must define a triggering event that causes data to be replicated on both applications. For example, you might define the triggering event to be a transaction such as entering a new customer or a new product. Then the replication would occur when the transaction is submitted. Alternatively, you might add a button to an applet that runs a workflow that triggers replication. Because Siebel Business Applications save data when the user steps off the record, the event you define is not necessarily tied to when data is written to the database.
- **Define the points in your business process when you want to update information across applications.** You identify points in the business process when you must use ASIs to update information between your Siebel Business Application and your back-office or a third-party application. For example, when the status of an order is updated in a back-office application, you might want the order status in your Siebel Business Application to be updated automatically.
  If an end user of the Siebel Business Application performs a query while accessing a new order, then you could have the back-office application update the Siebel order in real-time. If your back-office application has the capability to do this update, then you can implement it using the Update Siebel Order ASI. However, if it is not feasible for your back-office application to provide this update in real-time, then the end user can click a Get Status button on the user interface to request the updated order information.

Smart Part Numbers and Order Management Integration

When interfacing with back-office fulfillment applications, it is very important that each product can be identified. Smart part numbers are unique keys used to identify each combination of product ID and product attributes. The smart part number becomes the Stock Keeping Unit (SKU) used in the back office application.

You can create smart part numbers manually, or you can interface with an auto-generation application.

The smart part number for each line item appears in the Order screen, in the Line Detail subview associated with the line item.

For more information about smart part numbers, see *Siebel Product Administration Guide* .

About ASIs for Order Management

Siebel's Application Services Interfaces (ASIs) are predefined interfaces that allow the exchange of information between Siebel Business Applications and external applications.

Several features of Siebel Order Management use ASIs. They include:

- **Availability to promise.** Availability to promise (ATP) allows end users to inquire whether an item is available. End users can specify a requested date, a specific source such as an inventory location, and preferred shipment

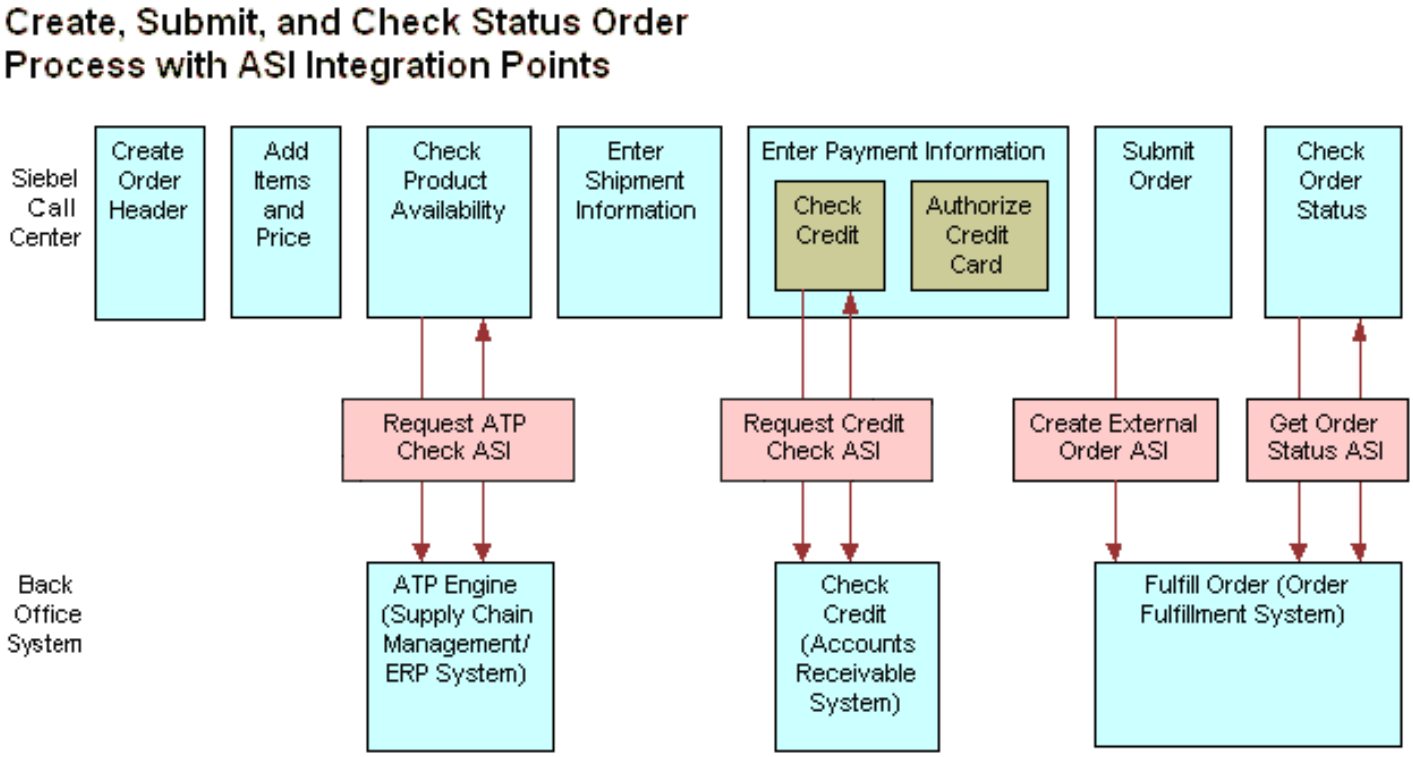
methods that could impact the delivery time. End users can also reserve available products for a customer to make sure they are available when the order is processed, and they can unreserve products if customers change their minds.

When the user clicks the Inquire, Reserve, or Unreserve button in a quote or order, the business service sends a request to a back office fulfillment engine to determine whether the specified line items are available. For details about how to set up the Availability to Promise ASI, see *Integrating Order Management with Third-Party Product Availability Applications*. For details about the user's experience of using the ATP functionality, see *Checking Availability to Promise (ATP)*.

- **Credit check.** When a customer uses the purchase order payment option for a quote or an order, the end user can perform a credit check to make sure that the customer can make the purchase on credit. In a quote, the user can perform the credit check by clicking the Credit Check button, which available in the Payment Detail - Purchase Order view when the payment method is Purchase Order. This action sends a request to an external application to get the credit decision for that transaction. Typically, the external application that performs the credit check is the back-office order or credit management application. For details about how to set up the Credit Check ASI, see *Integrating Order Management with Third-Party Credit-Check Applications* For details about the users' experience of using the credit check functionality, see *Checking Credit for a Purchase Order*.
- **Submit order.** When users have finished defining an order for a customer and the customer has agreed to the price and provided payment information, the order is submitted to the back office. When the user clicks the Submit button, the Submit Order ASI workflow sends information about the order to the back office application. For details about setting up the ASIs used by this workflow, see *Setting Up Quote and Order ASIs*. For details about the user's experience of using the submit order functionality, see *Submitting an Order*.
- **Submit quote.** After a quote has been verified, the user can submit the quote to the back-office or any external application. When the user chooses the Submit menu option for a quote record from the menu on the Quote list, the Submit Quote ASI workflow sends the quote information to the external application and updates Siebel with the response from the external application.

  The external application might use the quote information for converting a quote to an order or for production planning. Make any additional required configurations in the Siebel Business Application as required to meet your company's business needs. For details about setting up the ASIs used by this workflow, see *Setting Up Quote and Order ASIs*.
- **Query order status.** Typically, a company will receive information about the status of orders at regular intervals from the fulfillment application. However, when salespeople talk with customers about the status of an order, they want to make sure that they present the most current information. In order to do so, they can click the Get Status button on an order. This button calls the Get Order Status ASI workflow, which retrieves information from the back-office application. For details about setting up the ASIs used by this workflow, see *Setting Up Quote and Order ASIs*. For details about the user's experience of using the submit order functionality, see *Submitting an Order*.
- **Updating order status.** When the status of an order is updated in a back-office application, it can replicate that information in the Siebel Business Application to keep the two applications synchronized. This synchronization is done by using the Siebel Order ASI with a method of Update or Sync, and can be set up as a Web service. For details about using the Siebel Order ASI, see *Setting Up Quote and Order ASIs*.

The following figure shows the points in the order process where ASI integration occurs.



You must set up the ASIs shipped with Siebel Order Management before they can be used. This chapter provides the procedures that you must perform to set up these ASIs. It also describes the ways that Siebel Order Management calls ASIs, and documents the workflows that you may want to modify to change the ways that ASIs are called.

For general information about ASIs, see Siebel Application Services Interface Reference.

Setting Up Quote and Order ASIs

Two pairs of ASIs are used during the following order management processes:

- Submitting orders
- Importing quotes and orders
- Updating quote and order status
- Querying orders

These ASI pairs are:

- Siebel Quote (inbound) and External Quote (outbound)
- Siebel Order (inbound) and External Order (outbound)

Process of Integrating Quotes and Orders with Back-Office Applications

To set up real-time submission of quotes and orders with your back-office application, perform the following tasks:

- *Setting Up Web Services for Quote and Order ASIs*
- *Activating Workflows for Integration of Quotes and Orders*

Note: If you want to add to or change the fields that are sent to the back-office application, then you can customize the integration objects for each ASI. For more information, see *Siebel Application Services Interface Reference* .

Setting Up Web Services for Quote and Order ASIs

To set up Web services, you must set up both outbound and inbound Web services for the Quote and Order ASIs. The following procedures show you the steps required for each of these.

This task is a step in *Process of Integrating Quotes and Orders with Back-Office Applications*.

To set up the outbound Order Web service

1. Navigate to Administration - Web Services screen, and then the Outbound Web Services view.
2. In the Outbound Web Services list, create a query to display the record with *External Order* in the Name field.
3. Verify that the fields in the Outbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	External Order	Active

4. In the Service Ports list, click New, and enter the following values in the fields of the new record.

Field	Comments
Name	Enter a name for the port.
Port Type	Enter External Order.
Transport	Select from the list. If you want to set up testing, for example when you are customizing applets or extending the ASI, then select Local Business Service.

Field	Comments
Address	When you select a transport, the field is populated with a template address. Edit the address according to where you are sending the data.
Binding	If a SOAP header is expected by the receiving application, then select SOAP_RPC.

To set up the outbound Quote Web service

1. In the Outbound Web Services list, create a query to display the record with *External Quote* in the Name field.
2. Verify that the fields in the Outbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	External Quote	Active

3. In the Service Ports list, review the information in the fields for the External Quote service port and make any changes required for your application.

To set up the inbound Order Web service

1. Navigate to the Administration - Web Services screen, and then the Inbound Web Service view.
2. In the Inbound Web Service list, create a query to display the record *Siebel Order* in the Name field.
3. Verify that the fields in the Inbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	Siebel Order	Active

4. In the Service Ports list, review the information in the fields for the Siebel Order service port and make any changes required for your application.
5. In the Operations list, review the information shown and make any changes required for your application.

To set up the inbound Quote Web service

1. Navigate to the Administration - Web Services screen, and then the Inbound Web Services view.
2. In the Inbound Web Service list, create a query to display the record *Siebel Quote* in the Name field.
3. Verify that the fields in the Inbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	Siebel Quote	Active

4. In the Service Ports list, review the information in the fields for the Siebel Quote service port and make any changes required for your application.
5. In the Operations list, review the information shown and make any changes required for your application.

Activating Workflows for Integration of Quotes and Orders

In Siebel Tools, you must activate the following workflow processes used for real-time integration of quotes and orders:

- Get Order Status ASI
- Submit Order ASI
- Submit Quote ASI

For more information about activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

This task is a step in *Process of Integrating Quotes and Orders with Back-Office Applications*.

ASI Workflows for Quotes and Orders

This topic documents the workflow processes that are used to call the ASIs for quote and order processes. It is not necessary to modify these workflows, but you can modify them to suit your business model.

Get Order Status ASI

This workflow queries the outside application to retrieve the latest information on the order. It returns the new information and synchronizes the order on the current application.

Workflow Description

This workflow performs the following actions:

1. **Query Order By Id.** Queries for the order on the current application using the object id. Returns an order property set containing all the order information.
2. **Isolate Integration ID.** In the next step, the workflow will use a query by example. Query by example will take the property set and look for an order with the same exact values for the fields in the property set. If the order has changed on the outside application, then those values will no longer match. The workflow needs to query by a value that will not change, the integration ID. Isolate Integration Id takes in a property set and removes all the fields, except the integration ID
3. **External Order Proxy.** Using the property set, which only has an integration ID, this step queries by example on the outside applications. After it finds the order with that integration ID, it returns an order property set containing all the order information.
4. **Synchronize Order.** Takes the order property set returned by the previous step and synchronizes it with the order in the current application.

Submit Order ASI

This workflow submits the order to the outside application to synchronize the information between the two applications.

Workflow Description

This workflow performs the following actions:

1. **Query Order By Id.** Queries for the order on the current application using the object id. Returns an order property set containing all the order information.
2. **External Order Proxy.** Calls insert on the outside application, passing in the order property set. This step returns an order property set, which will contain any changes the outside application made to the order.
3. **Synchronize Order.** Takes the order property set returned by the previous step and synchronizes it with the order in the current application.

Submit Quote ASI

Workflow Description

This workflow submits the quote to the outside application to synchronize the information between the two applications.

This workflow performs the following actions:

1. **Query For Quote By Id.** Queries for the quote on the current application using the object id. Returns a quote property set containing all the quote information.
2. **External Quote Proxy.** Calls insert on the outside application, passing in the quote property set. This step returns a quote property set, which will contain any changes the outside application made to the quote.
3. **Synchronize Quote.** Takes the quote property set returned by the previous step and synchronizes it with the quote in the current application.

5 Integrating Order to Cash and Distributed Order Orchestration

Integrating Order to Cash and Distributed Order Orchestration

This chapter describes how to integrate order-to-cash and distributed-order-orchestration functionality using Oracle's Application Integration Architecture (AIA). It includes the following topics:

- *Process of Integrating Order to Cash*
- *Integrating Distributed Order Orchestration*
- *Setting Up Data Validation Rules for Distributed Order Orchestration*

Process of Integrating Order to Cash

You integrate order to cash functionality using Oracle Application Integration Architecture (AIA) rather then using the integration strategies described in *Integrating Order Management with Back-Office Applications*.

Order to cash features are described in the following topics:

- *Canceling an Order Line Item*. Users can cancel order line items.
- *Placing an Order on Hold*. Users can stop orders temporarily by using the Hold and Hold Reason fields in the order header.
- *Canceling an Order*. Users can cancel orders with a status of Booked or Awaiting Shipping by clicking the Cancel Order button.
- *Adding Shipping Information*. Users can enter special instructions on shipping and packing in the Ship Instructions and Pack Instructions fields.
- *Checking the Status of Order Shipments*. A read-only Back Office Error Text field in the order header shows feedback from the back office about order errors.

To integrate order to cash using AIA, perform the following tasks:

1. *Setting System Preferences for Order to Cash*
2. *Setting Up Integration Users for Order to Cash*
3. *Configuring Web Services for Order to Cash*
4. *Enabling Component Groups for Order to Cash*
5. *Configuring Workflow Policies for Order to Cash*
6. *Activating Workflows for Order to Cash*
7. *Setting Up Data Validation Rules for Order to Cash*

Setting System Preferences for Order to Cash

Administrators must set system preference values. The system preferences listed in this topic control aspects of Oracle Application Integration Architecture integration functionality.

This task is a step in *Process of Integrating Order to Cash*.

To set the system preferences for order to cash

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. Query for the Oracle Application Integration Architecture integration system preferences, and set the appropriate value for each.

   Oracle Application Integration Architecture integration system preferences are listed in the following table.

System Preference	Comments
Cancel Error Mode	Controls the behavior when an order that has future-dated line items is cancelled. Those future-dated line items will be affected by the cancellation of this order. Values are as follows: o **1** Do not display warning messages. o **2** Display warning messages. o **3** (Default) Display warning and error messages, and stop processing.
Validate Due Date	Controls the behavior of the pop-up applet that prompts for the Due Date for change orders such as Modify, Disconnect, Suspend, and Resume. Values are as follows: o **1** No warning messages are displayed. Modify, Suspend, Resume, and Disconnect (MACD) operations are continued for assets and orders. o **2** (Default) Display warning messages. Enable the Continue button on the Due Date window. o **3** Display warning messages. Disable the Continue button on the Due Date window.
Enable AIA Comms	Controls whether or not to enable the Oracle Application Integration Architecture Communications integration features. Values are as follows: o **TRUE** Enable Oracle Application Integration Architecture Communications integration features. o **FALSE** Disable Oracle Application Integration Architecture Communications integration features.
Enable AIA Testing	Controls whether or not to enable the Oracle Application Integration Architecture Communications integration testing features. Values are as follows: o **TRUE** Enable Oracle Application Integration Architecture Communications integration testing. o **FALSE** Disable Oracle Application Integration Architecture Communications integration testing.

System Preference	Comments
AIA Order Backup Path	XML files are available on the specified path, with file names in the form OrderNumber>_ RevisionNumber>.xml. The files can be sorted by Date Modified. Enabled *only* if the Enable AIA Testing system preference is set to TRUE. Value is as follows: /export/home/qa1/orderbackup/
Workspace Release Flag	A product that is synchronized from Oracle Product Information Management Data Hub is a Workspace version, that is, a test version, in Siebel CRM. This system preference holds or releases all the products associated with a Workspace. This system preference appears in the SWIProductImportIO, SWIAttributeImportIO, and SWIProductClassImport Web services. Additionally, a Product Release Flag in each product definition can be used to release the individual product. The Workspace Release Flag system preference has the following values: o **N**. Default value. Siebel CRM does not automatically release the products. In this case, a user can set the Product Release Flag of an individual product to Y to release that specific product, and all the other products in the same payload will not be released. o **Y**. Siebel CRM automatically releases the products. In this case, all the products belonging to the Workspace are automatically released. This setting overrides the Product Release Flag settings of the individual products in the Workspace. o **D**. Signifies a default or empty value. In this case, the default value of this system preference, which is N, determines whether to automatically release the product.
Enable AIA MDM	Enables or disables the Master Data Management (MDM) PIP. Default value is set to FALSE. To enable the MDM PIP, you must change this system preference value to TRUE and restart the Siebel server.
Enable AIA OrderToCash	Enables or disables the Distributed-Order-Orchestration integration and Order-to-Cash PIP. Default value is set to FALSE. To enable the Distributed-Order-Orchestration integration, you must change this system preference value to TRUE and restart the Siebel server.

CAUTION: It is recommended that you *not* set the Enable AIA Comms and Enable AIA OrderToCash system preferences to TRUE in the same Siebel CRM instance. Interoperability of Siebel Communications PIPs (Order-to-Bill and Agent Assisted Billing Care) with the Order-to-Cash PIP is not tested or supported by Oracle.

3. Set other system preferences, if necessary.

For more information about system preferences, see the chapter on ongoing tasks and the system preferences appendix in *Siebel Applications Administration Guide* .

Setting Up Integration Users for Order to Cash

Administrators must designate integration users. These integration user credentials are used by the Siebel application to accept inbound Web service calls from the Oracle Application Integration Architecture product. Integration users are used by the inbound Web services to log in and to create, update, or query records in the Siebel application. You must also set up integration user credentials in the Oracle Application Integration Architecture product that you are using.

This task is a step in *Process of Integrating Order to Cash*.

To set up an integration user

1. Navigate to the Administration - User screen, then the Employees view.
2. In the employees list, select the record for the employee who you want to set up as an integration user.
3. Select the check box in the Integration User Flag field for this employee.

Configuring Web Services for Order to Cash

The Process Integration Packs (PIPs) in the Oracle Application Integration Architecture product can use Web services. To configure a Web service, administrators change the URI address for its service ports. For descriptions of Web services used by order to cash, and information about Web services, see *Siebel CRM Web Services Reference* .

This task is a step in *Process of Integrating Order to Cash*.

Web Services Used by Order to Cash

This section lists Siebel Web Services that are available for customer management integration, which you can configure using the instructions in subsequent sections.

Integration uses the following inbound Siebel Web Services:

- SWICustomerParty
  - Operation Name: SWICustomerPartyInsertOrUpdate
  - Request Schema: SWICustomerPartyInsertOrUpdate_Input
  - Response Schema: SWICustomerPartyInsertOrUpdate_Output
- SWICustomerParty
- SWICustomerPartyQueryByExample
  - Request Schema: SWICustomerPartyQueryByExample_Input
  - Response Schema: SWICustomerPartyQueryByExample_Output
- SWICustomerParty
  - Operation Name: SWICustomerPartySynchronize
  - Request Schema: SWICustomerPartySynchronize_Input
  - Response Schema: SWICustomerPartySynchronize_Output
- SWICustomerParty

- Operation Name: SWIMergeServices
- Request: SchemaSWIMergeServicesMerge_Input
- Response Schema: SWIMergeServicesMerge_Output

The following events trigger Siebel Web Services:

- Account Updated

  The Siebel application calls the SyncCustomerSiebelEventAggregator service with an AccountUpdatedABM.
- Contact Updated

  The Siebel application calls the SyncCustomerSiebelEventAggregator service with a ContactUpdatedABM.
- Address Updated

  The Siebel application calls the SyncCustomerSiebelEventAggregator service with an AddressUpdatedABM.

Integration uses the following outbound Siebel Web Services:

- CalculateShippingChargeSalesOrderSiebelReqABCSImpl
- CheckATPSalesOrderSiebelReqABCSImpl
- ConfiguratorUserLangSiebelAdapter
- CreditCheckSalesOrderSiebelReqABCSImpl
- FetchAccountSiebelReqABCSImpl
- FetchContactSiebelReqABCSImpl
- MatchAccountSiebelReqABCSImpl
- MatchContactSiebelReqABCSImpl
- PaymentAuthorizationSalesOrderSiebelReqABCSImpl
- ProcessQuoteSiebelJMSProducer
- ProcessSalesOrderSiebelJMSProducerV2
- RequestProductStructureSiebelJMSProducer
- SyncCustomerSiebelEventAggregator
- ESB_ConfiguratorCopyConfigEbizAdapter_Service

Configuring Inbound Web Services

A call into the Siebel business application uses an inbound Web service. To configure the inbound Web services, complete the steps in the following procedure.

To configure the inbound Web services

1. Navigate to the Administration - Web Services screen, then the Inbound Web Services view.
2. Query for the Web service name in the Inbound Web Services list.

   For the names of the inbound Web services, see the Web service summary in *Siebel CRM Web Services Reference* .
3. If the Web service is not active, then change the status from Inactive to Active, and click Clear Cache.
4. In the Service Ports list for the Web service, change the following text strings in the Address field to the appropriate data for your server:

- <webserver>
- <lang>
- <UserName>
- <Password>

Configuring Outbound Web Services

A call out of the Siebel business application uses an outbound Web service. To configure the outbound Web services, complete the steps in the following procedure.

To configure the outbound Web services

1. Navigate to the Administration - Web Services screen, then the Outbound Web Services view.
2. Query for the Web service name in the Outbound Web Services list.

   For the names of the outbound Web services, see the Web service summary in *Siebel CRM Web Services Reference* .
3. In the Service Ports list for the Web service, change the web server name and port in the Address field to the appropriate data for your server.

Enabling Component Groups for Order to Cash

Administrators must verify that the component groups for order creation are enabled on the Siebel Server. When you enable component groups, all of the required components in those groups are started on the server for transaction logging. For information about enabling component groups, see *Siebel System Administration Guide* .

> **Note:** In addition to the component groups listed in this procedure, administrators must activate the component groups for any Siebel business application that they are using. For example, if you are using Siebel High Technology/ Industrial Manufacturing, then you must activate the component groups needed for that application.

You must also turn asset-based ordering off for the components that you enable for order to cash or for distributed order orchestration.

This task is a step in *Process of Integrating Order to Cash*.

To enable component groups for order to cash

1. Navigate to Administration - Server Configuration screen, then Enterprises and Component Groups view.
2. Enable Component Groups for order to cash.

   a. Query for each of the following Component Groups

      - . SIA
      - . Siebel CME [Alias- Communications]
      - . Siebel ISS [Alias- ISS]
      - . Workflow Management [Alias- Workflow]
      - . HTIM
      - . EAI

b. For each component group, if the value of Enable State is Disabled, then click Enable.
   - Click the Parameters view tab.
   - Query for the parameter Order Management - Enable Asset Based Ordering.
   - Set the Current Value to False and save by stepping off the record.

3. Disable asset-based ordering for these components.
   a. Navigate to Administration - Server Configuration screen, then Servers view.
   b. Query for Order Management - Enable Asset Based Ordering.
   c. Set the value to false.
4. Synchronize the components:
   - Navigate to Administration - Server Configuration screen, then Enterprises, and Synchronize view.
   - Select all the records.
   - Click Synchronize.
5. Restart the server.

For more information about enabling components, see *Siebel CTI Administration Guide*

Configuring Workflow Policies for Order to Cash

After administrators enable the component groups for order creation, they must configure the workflow policies. When information about an account or contact changes in the Siebel application, workflow policies invoke workflows and call Web services to synchronize data with the third-party billing application.

Workflow policies are used to support the following functions in Order to Cash.

- Account synchronization and update
- Contact synchronization and update

Though these two functions are not supported in distributed order orchestration, these workflow policies can be useful to track the changes in Siebel accounts and contacts in AIA.

This task is a step in *Process of Integrating Order to Cash*.

To configure the workflow policies

1. Create a job to run the Generate Triggers component in the Workflow Management component group.
   For more information, see the topic about running Generate Trigger in *Siebel Business Process Framework: Workflow Guide* .
2. Expose the Default Tasks parameter so that you can change this parameter value when you copy the Workflow Manager Agent component in the next step.
   For more information, see *Exposing the Default Tasks Parameter*.
3. Create a copy of the Workflow Monitor Agent component.
   For more information, see *Copying the Workflow Monitor Agent Component*.
4. Start the new Workflow Monitor Agent task for customer synchronization.
   For more information, see the topic about using the Workflow Monitor Agent in *Siebel Business Process Framework: Workflow Guide* .

5. (Optional) Review the log files for the new Workflow Monitor Agent task.

   For more information, see the topic about workflow policies and Siebel Server task trace files in *Siebel Business Process Framework: Workflow Guide* .

Exposing the Default Tasks Parameter

The Workflow Monitor Agent component in the Workflow Management component group contains the Default Tasks parameter, which is hidden from view by default. Administrators must reveal this parameter to advanced users so that these users can change its value in a copy of the Workflow Manager Agent component that they create.

To expose the Default Tasks Parameter

1. Navigate to the Administration - Server Configuration screen, then the Servers view.
2. Select the appropriate server in the Siebel Servers list.
3. Click the Components view tab, and select the Server Manager component in the Components list.
4. Page down, click the Parameters view tab, and select the Show Advanced Objects parameter in the Component Parameters list.
5. Change the Value field for this parameter to TRUE.
6. Log out of the application and then log in again so that the parameter setting takes effect.

Copying the Workflow Monitor Agent Component

For information about creating a copy of the Workflow Monitor Agent component, see the topic about using Workflow Monitor Agent in *Siebel Business Process Framework: Workflow Guide* .

> **Note:** Before you restart the Siebel Server so that the component definition takes effect, rename the dictionary cache file (diccache.dat) at *<Installation Directory>* `\ses\siebsrvr\bin` so that a new dictionary cache file is created when you restart.

The following table shows the values to enter for some fields for the copied component. To see this view, click the Component Definitions view in the Administration - Server Configuration screen, then the Enterprises view.

Field	Value
Component	Workflow Monitor Agent SWI
Alias	WorkflowMonSWI
Component Group	Workflow Management

The following table shows the values to enter for some parameters for the copied component in the Component Definitions view in the Administration - Server Configuration screen, then the Enterprises view. To see the advanced parameters, click Advanced in the Component Parameters list.

Parameter	Type	Value
Action interval	Not applicable.	5

Parameter	Type	Value
Default Tasks	Advanced	1
Sleep Time	Advanced	15
Group Name	Not applicable.	SWI Customer Update Policy Grp

Note: Notice that, in the first table in this topic, the Component Group must be named *Workflow Management* but in the second table in this topic, the Group Name of the new component group must be SWI Customer Update Policy Grp.

Activating Workflows for Order to Cash

The administrator must activate the workflows used by order to cash.

This task is a step in *Process of Integrating Order to Cash*.

To activate workflows for order to cash

1. Revise and activate workflows.
   a. Revise the following workflows, changing the Process Property UTCCanonical from its default value of Y to N:
      - CMU Credit Alert Status Sync
      - CMU SIA Submit Adjustment Request to External
      - SISOMBillingSubmitOrderWebService
      - SWI Account Update Workflow
      - SWI Billing Profile Update Workflow
      - SWI Contact Update Workflow
      - SWI Address Update Workflow
      - SWI Cancel Sales Order_O2C
      - SWI External Account Request Sync Process
      - SWI External Contact Request Sync Process
      - SWI Special Rating - Synchronize Process
      - SWIAdjustmentStatusUpdate
      - SWIOrderUpsert
      - SWIOrderUpsert_O2C
      - SWIOrderUpsertSubProcess
      - SWIQuoteUpsert
      - SWISendATPCheck
      - SWISendATPCheckLine
      - SWISendCalculateShippingCharge
      - SWISendCreditCheck
      - SWISendOrder
      - SWISendPaymentAuthorization
      - SWISendQuote
   b. Publish and then activate these workflows.
   c. Check the status of these workflows, and if it is In Progress, then make it complete by deploying the Workflow.

2. Activate the following workflows:
    - Submit Quote ASI
    - Submit Order ASI
    - SWI Initial Load Non CP Order Release
    - SWI External Product Sync Workflow
    - SWI Configurator Load
    - SWI Account Update Workflow
    - Oracle Verify Item (Quote)
    - Oracle Verify Item (Order)
    - Oracle Verify Header (Quote)
    - Oracle Verify Header (Order)
    - Oracle Verify Complex Product All (Quote)
    - Oracle Verify Complex Product All (Order)
    - Oracle Configurator Load
    - CZ Return
    - CZ PSP Interface
    - Account - New Order
    - SWIReviseOrderValidation

Setting Up Data Validation Rules for Order to Cash

Validation rules validate an order based on the rules configured in the Data Validation expression. By default, the data validation rules are set to False. When a workflow evaluates the data validation rules and finds a rule set to False, it continues to the next expression in turn. When it finds a rule set to True, it stops searching and no other rules are evaluated.

To place an order, all conditions must pass all the validation rules. If the order passes, then the order is submitted to the back office. If the order is evaluated as False, then the corresponding error message is shown.

If you are using order to cash functionality, then you can set up data validation rules to verify the quality and accuracy of your transactions, you can set up data validation.

This task is a step in *Process of Integrating Order to Cash*.

To set up data validation rules for order to cash

1. Navigate to the Administration - Data Validation screen, Rules Sets view.

    In the relevant Expression field, modify the expression.Transferring the Shipping Method From Quote to Asset to Order

Integrating Distributed Order Orchestration

To set up an Oracle Application Integration Architecture integration between your Siebel business application and Oracle Distributed Order Orchestration, you perform the same tasks needed to integrate order to cash, except that you define different data validation rules.

Distributed order orchestration uses the same workflows and Web services as order to cash.

To integrate distributed order orchestration

1. Perform the tasks for integrating order to cash

   For more information, see *Process of Integrating Order to Cash*.
2. Instead of setting the data validation rules in this process, set the data validation rules described in *Setting Up Data Validation Rules for Distributed Order Orchestration*.
3. Instead of setting up Siebel eBusiness Configurator, set up and use Siebel Configurator for DOO integration.

Setting Up Data Validation Rules for Distributed Order Orchestration

For information about why you must set up data validation rules, see *Setting Up Data Validation Rules for Order to Cash*.

To set up data validation rules for distributed order orchestration

1. Navigate to the Administration - Data Validation tab, then the Rules Sets view.
2. Create a new rule set based on the existing rule set named SWI Revise Order Validation.
   a. In the Rule Sets list, select SWI Revise Order Validation.
   b. From the menu, select Revise Rule Set.

      A new copy of this rule set is created.
   c. Click the name of the newly created rule set to display the list of rules in this rule set.
   d. Modify the expression for two rules, as shown in the following table.

Rule	Expression
SWI Revise Order	(SystemPreference("Enable AIA OrderToCash")="FALSE") OR ((([Status] = LookupValue("FS_ORDER_STATUS","Booked"))) AND (([Status] <> LookupValue("FS_ORDER_STATUS","Shipped")) AND ([Status] <> LookupValue("FS_ORDER_STATUS", "Closed"))))
SWI Revise Order Line	(SystemPreference("Enable AIA OrderToCash")="FALSE") OR((([Status] = LookupValue("FS_ORDER_STATUS","Booked")) OR ([Status] = LookupValue("FS_

Rule	Expression
	ORDER_STATUS","Picked")) OR ([Status] = LookupValue("FS_ORDER_STATUS", "Cancelled")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Pending")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Shipped")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Closed")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Supply Eligible")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Billed")) OR ([Status] = LookupValue("FS_ORDER_STATUS","Partially Shipped"))))

e. From the menu, select Activate Rule Set.

   The status changes from In Process to Active.

3. Create a new rule set based on the existing rule set named O2C Cancel Order Validation.

   a. In the Rule Sets list, select O2C Cancel Order Validation.
   b. From the menu, select Revise Rule Set.

      A new copy of this rule set is created.
   c. Click the name of the newly created rule set to display the list of rules in this rule set.
   d. Modify the expression for one rule, as shown in the following table.

Rule	Expression
PONR Not Reached	NOT(SystemPreference("Enable AIA OrderToCash")="TRUE" AND ([Status] = LookupValue ('FS_ORDER_STATUS', 'Shipped')) or ([Status] = LookupValue ('FS_ORDER_STATUS', 'Closed')) or ([Status] = LookupValue ('FS_ORDER_STATUS', 'Fulfilled')) or ([Status] = LookupValue ('FS_ORDER_STATUS', 'Supply Eligible')) or ([Status] = LookupValue ('FS_ORDER_STATUS', 'Billed')) or ([Status] = LookupValue ('FS_ORDER_STATUS', 'Partially Shipped')))

   e. From the menu, select Activate Rule Set.

      The status changes from In Process to Active.

6 Integrating Order Management with Third-Party Product Availability Applications

Integrating Order Management with Third-Party Product Availability Applications

This chapter tells you how to integrate the Siebel Business Application with an external Availability to Promise (ATP) engine to provide customers with information about availability of products they wish to order. It includes the following topics:

- *About Third-Party Product Availability Applications*
- *Process of Setting Up Third-Party Product Availability Applications*

About Third-Party Product Availability Applications

Product availability information is stored in third-party ATP applications. When a salesperson clicks the Inquire button or when a customer clicks a Check Availability button, product availability information is passed to the Siebel Business Application by a third-party product availability fulfillment engine, also known as an availability-to-promise (ATP) engine. Siebel Order Management displays whatever data the ATP engine returns.

Integration with the product availability fulfillment engine uses the business service named External ATP Check. In addition it uses a specific, prebuilt Application Services Interface (ASI) named ATP Check. Some configuration is required to set up this integration.

Siebel Application Service Interface (ASI) is used to integrate order management with third-party product-availability applications. For an overview of how ASIs are used for integrating order management, see *About ASIs for Order Management*. For general information about how to set up, use, and modify integrations according to ASIs, see Integration Platform Technologies: Siebel Enterprise Application Integration.

Process of Setting Up Third-Party Product Availability Applications

To set up the use of third-party product availability-to-promise applications, you must perform the following tasks:

1. *Checking Availability Fulfillment Methods*
2. *Setting the ATP Action Parameter in Quote and Order Business Components Properties*
3. *Verifying the External ATP Check Business Service User Properties*
4. *Configuring Web Services for Availability to Promise*

   When setting up the use of third-party product availability-to-promise applications, you can perform the following optional tasks:

5. *Preventing Rollup of Availability Summary Data*
6. *Extending the ASI for Availability to Promise*
7. *Extending Returned Data for Availability to Promise*
8. *Customizing UI Terms for Availability to Promise*
   When setting up the use of third-party product availability-to-promise applications, you can perform the following optional task that applies only to Siebel eSales:
9. *Requiring User Registration for Availability to Promise*
   For more information about how to set up, use, and modify integrations according ASIs, see *Siebel Application Services Interface Reference* .
   For information about the employee end user's experience in using the availability-to-promise features, see *Checking Availability to Promise (ATP)*.

Checking Availability Fulfillment Methods

For the integration to work, you must check that your ATP engine supports the Inquire, Reserve, and Unreserve methods. If your ATP engine refers to these methods with different terms, then you may need to do some configuration to map the method names.

For customers to check availability using Siebel eSales, you must find a method that is supported by your ATP engine and that checks availability without reserving it. The Inquire method is one possible example. You may want to check other supported methods, such as Reserve and Unreserve, to see how you can extend your availability checking function.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

Setting the ATP Action Parameter in Quote and Order Business Components Properties

Employees start product availability from a Quote or Order business component at either the header level or the line level. The Quote or Order business component defines the method to be used to call the ATP Check ASI. This method is one of those listed in the following table, depending on whether the user clicked a button to Inquire, Reserve, or Unreserve.

Customers using Siebel eSales start product availability from the Quote business component. The Quote business component defines the method ATPInquireAll, which calls the ATP Check ASI.

You must make sure that the value of the ATP Action Parameter in the user property defining the ATPInquireAll method in both the Quote and Order business components matches the commands expected by the ATP engine.

However, if your ATP engine expects different commands, then you can change them in Siebel Tools.

User Interface Element	business component	Method Name	ATP Action
Inquire All button, Quote or Order header	Quote or Order Entry - Orders	ATPInquireAll	Inquire
Inquire button, Quote or Order Line Items list	Quote Item or Order Entry - Line Items	ATPInquire	Inquire

User Interface Element	business component	Method Name	ATP Action
Reserve All button, Quote or Order header	Quote or Order Entry - Orders	ATPReserveAll	Reserve
Reserve button, Quote or Order Line Items list	Quote Item or Order Entry - Line Items	ATPReserve	Reserve
Unreserve All button, Quote or Order header	Quote or Order Entry - Orders	ATPUnReserveAll	Unreserve
Unreserve button, Quote or Order Line Items list	Quote Item or Order Entry - Line Items	ATPUnReserve	Unreserve

The following procedure shows you how to make changes to the ATP Action parameter for the method ATPInquireAll. You can use it as a model to make changes to any of the other ATP methods, as appropriate.

Note: If the middleware handles this mapping, then you do not need to change this ATP Action Parameter. For example, if the middleware knows that ATP Action is Inquire in your Siebel Business Application, and that ATP Action is Check in your fulfillment engine, then the middleware provides the mapping.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To set the business component user properties for the Quote business component

1. In Siebel Tools, select the Quote business component.
2. Select Business Component User Prop.
3. In the Value field, query for *ATP*.

   This query returns all of the Quote business component user properties that are related to the ATP functionality.
4. For the user property that includes the ATPInquireAll method, edit the ATP Action parameter so that it exactly matches the command expected by the ATP engine. The default value is Inquire.

To set the business component user properties for the Order business component

1. In Siebel Tools, select the Order business component.
2. Select Business Component User Prop.
3. In the Value field, query for *ATP*.

   This query returns all of the Order business component user properties that are related to the ATP functionality.
4. For the user property that includes the ATPInquireAll method, edit the ATP Action parameter so that it exactly matches the command expected by the ATP engine. The default value is Inquire.

Verifying the External ATP Check Business Service User Properties

You must verify the External ATP Check business service user properties.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To verify the External ATP Check business service user properties

1. In Siebel Tools, select the External ATP Check business service.
2. Select Business Service User Prop.
3. Verify that the Names and Values of the User Properties match those in the following table.

Name	Value
siebel_port_name	Default
siebel_web_service_name	External ATP Check
siebel_web_service_namespace	http://siebel.com/asi/

Configuring Web Services for Availability to Promise

Web services are used to direct the inbound and outbound messages to and from:

- The third-party ATP engine
- A demonstration script, which can be used to test your integration setup
- A middleware or integration server that will integrate with your fulfillment engine

You must make sure that the Web services are correctly configured.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To configure Web services for availability to promise

1. In your Siebel Business Application, navigate to the Administration - Web Services screen, and then the Outbound Web Services view.
2. Select the row with External ATP Check in the Name field.
3. Verify that the fields in the Outbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	External ATP Check	Active

4. In the Service Ports list, click New, and enter the following values in the fields of the new record.

Field	Comments
Name	Enter *Default*.
Service Display Name	Enter *External ATP Check*.
Transport	Select from the list. If you want to set up testing, for example when you are customizing applets or extending the ASI, then select Local Business Service.
Address	When you select a transport, the field is populated with a template address. Edit the address according to whether you are sending the data to middleware or the ATP engine. If you want to set up testing, for example when you are customizing applets or extending the ASI, then enter ATP ASI Test. For more information, see *Using the ATP ASI Testing Script*.
Binding	If a SOAP header is expected by the receiving application, then select SOAP_RPC. If you are using ATP ASI Test, then select Property Set.

For more information about configuring Web Services and setting up demonstration scripts, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

Using the ATP ASI Testing Script

The ATP ASI test is a testing script that simulates a simple fulfillment engine. It writes the output and response to an XML file on the server. By default the script writes the output and the response to ${SIEBEL_HOME}/bin, for example, D:\Siebel\bin. It writes an output file named ATPOutput.xml. This file represents what you would be sending to your middleware or fulfillment engine. Then the script generates a real-time response and sends the XML back to the Web service. The XML response that is generated is written into a file named ATPFinished.xml, which you can see in the file structure.

This testing infrastructure allows you to analyze what is sent to the middleware and what the Siebel Business Application is expecting back, and allows you to test your user interface.

Preventing Rollup of Availability Summary Data

Product availability information is displayed for:

- The entire quote or order
- The quote or order line items

- The scheduled line items

The following information is preconfigured to roll up:

- Available status, date, and quantity are rolled up from the promised lines to the quote or order lines.
- Available status and date are rolled up from the quote or order lines to the quote or order header.

If your availability fulfillment engine only returns promised schedule lines and does not roll up this data to the line or the header, then the preconfigured behavior will provide you with a rollup of the data.

However, if your fulfillment engine already calculates this rollup, or if you do not want to roll up these fields, then you may want to turn off this behavior.

Following are some examples of the preconfigured rollup of availability information:

- **Dates.** The latest date is rolled up at the parent level. For example, if line 1 has an available date of 7/15/04 and line 2 has an available date of 7/23/04, then the available date for the order is set to 7/23/04.
- **Status.** If all lines have an ATP status of Available, then the order also has a status of Available. However, if one of the lines has a different status, then the order status is kept blank.
- **Quantity.** If Schedule line 1 has an availability quantity of 10, Schedule line 2 has an available quantity of 20, and both lines have the same ATP status of Available, then the quantity rolled up is the sum of both quantities (30).

Note: Rollup only occurs on quantity and date when the status of the promised line or the status of the line item is Available or Reserved.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To prevent rollup availability summary date

1. In Siebel Tools, select the ATP business service.
2. Select Business Service User Prop.
3. Select the Rollup Availability user property, and change the value from Y to N.

Extending the ASI for Availability to Promise

If you want to extend the ASI, then you must add fields to the following objects:

- The External Integration object named ATP Check Interface
- The following Internal Integration objects:
  - ATP Check Interface Request - Quotes
  - ATP Check Interface Request - Orders
  - ATP Check Interface Response - Quotes
  - ATP Check Interface Response - Orders

This extension will create the necessary mapping between the ATP interface structure and the Siebel Quote and Order business components.

Note: If you make changes to these objects, then the names of the fields in the Internal and External integration objects must match.

If you add a field, then make sure that the names of the fields in the Internal and External integration objects match. For example, if you want to add the field Account DUNS# that maps on quote to Account DUNS#-quote and on order to Account DUNS#-order, then you would do so as shown in the following table.

Object Name	Integration Component Field Name	Integration Component External Name
ATP Check Interface	Account DUNS#	None
ATP Check Interface Request - Quotes	Account DUNS#	*your mapping of Account DUNS # to the Quotes Business Component*
ATP Check Interface Request - Orders	Account DUNS#	*your mapping of Account DUNS # to the Orders Business Component*

If there are fields that you do not want the fulfillment engine to be able to modify, such as Ship To Address Id, then add the NoUpdate Field user property to the internal response integration objects, as follows:

- ATP Check Interface Response - Quotes

  Integration Component Field:

  Name: Ship To Address Id

  External Name: *your mapping of Ship To Address Id to the Quotes Business Component*

  Integration Component Field User Property:

  Name: NoUpdate

  Value: Y
- ATP Check Interface Response - Orders

  Integration Component Field:

  Name: Ship To Address Id

  External Name: *your mapping of Ship To Address Id to the Orders Business Component*

  Integration Component Field User Property:

  Name: NoUpdate

  Value: Y

For information about how to extend the integration object, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To prevent an added field from being modified by the fulfillment engine

1. In Siebel Tools, select Integration Object in the Object Explorer.

2. Select the integration object ATP Check Interface Response - Quotes.
3. Expand Integration Object in the Object Explorer, and select Integration Component.
4. Select the Quote integration component.
5. Expand Integration Component in the Object Explorer and select Integration Component Field.
6. Select the name of the added field.
7. Expand Integration Component Field in the Object Explorer and select Integration Component Field User Prop.
8. Set the Name of the Integration Component Field User Property to NoUpdate.
9. Set the Value to Y.

Extending Returned Data for Availability to Promise

For information about how to extend the data that is received from the third-party inventory management application, including data about products that are expected to be manufactured, see *Siebel Application Services Interface Reference* .

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

Customizing UI Terms for Availability to Promise

You can change the following terms in the user interface to more precisely reflect the type of data expected by, and returned by, your particular product availability fulfillment engine.

- **Requested Date.** This date could be either the date on which the customer wants to receive the product or the date on which the customer wants the seller to ship the product.
- **Available Date.** This date could be either the date that the customer will receive the product or the date that the product will be shipped.

These two fields are designed to work together, reflecting parallel information.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

Requiring User Registration for Availability to Promise

Siebel eSales does not require the customer to register on the Web site before checking product availability. However, you can configure the logic behind the Check Availability button to require registration by:

- Hiding the Check Availability button. In this case, anonymous users will not see the button unless they explicitly log in.
- Adding the Explicit Login parameter to the view.

This task applies to customers checking product availability using Siebel eSales. It does not apply to employees checking product availability.

This task is a step in *Process of Setting Up Third-Party Product Availability Applications*.

To hide the Check Availability button

1. In Siebel Tools, lock the eSales - Quote project.

2. In the Object Explorer, click Applet and select Quote Form Applet - Current (eSales).
3. In the Object Explorer, click Applet User Prop.
4. Create a new user property named Hide Control 4 with the following value: 'ButtonInquireAll','GetProfileAttr("Is Anonymous") = "TRUE"'

To add the Explicit Login parameter to the ATP Fulfillment View (e Sales)

1. In Siebel Tools, lock the ATP project.
2. In the Object Explorer, click View and select ATP Fulfillment View (eSales).
3. Right-click the Views bar and select Columns Displayed.
4. In the Available Columns box, select Explicit Login, move it to the Displayed Columns box, and click OK.
5. Navigate horizontally across the view and check the Explicit Login column for ATP Fulfillment View (eSales).
6. Check in and compile the projects.

7 Integrating Order Management with Third-Party Credit-Check Applications

Integrating Order Management with Third-Party Credit-Check Applications

This chapter discusses how to integrate order management with third-party credit-check applications. It includes the following topics:

- *About Third-Party Purchase Order Credit Checking Applications*
- *Process of Setting Up Purchase Order Credit Checking Applications*

About Third-Party Purchase Order Credit Checking Applications

Before completing an order for products or servicesSiebel eSales, your company may want to assess whether the customer has a credit rating that is good enough for you to sell to them. Different applications are used to perform credit checks depending on the method of payment.

- For business customers paying with purchase orders, a back-office, third-party accounts receivable or credit management application does credit checks on the customer account. This chapter describes how to set up these applications.
- For customers paying with a credit card, a third-party payment application does credit checks. This feature is available in customer applications. For more information about third-party payment applications, see Siebel eSales Administration Guide.

Third-party accounts receivable or credit management applications use logic to make a credit determination using your company's previous history with the customer account and other credit data.

Integration with third-party credit checking applications uses a specific, prebuilt Application Services Interface (ASI) named External Credit Check. Some configuration is required to set up this integration, as described in this chapter.

For information about how to extend these integration objects, see Integration Platform Technologies: Siebel Enterprise Application Integration.

Sales representatives or call center agents can make purchase order credit checks for a quote or order. In the order process, this credit check is done at the payment line level, when the user clicks the Credit Check button in an order. In the quote process, it is done at the header level, when the user chooses the Verify menu option in a quote. The following two workflows directly control the process of checking the credit status of business accounts:

- Credit Check - Quotes
- Credit Check - Orders

In Siebel eSales, credit checks are performed as part of the check-out process after the customer clicks the Check Out button. The following two workflows directly control the process of checking the credit status of business accounts in Siebel eSales:

- eSales - Authorize Order Process
- Credit Check - Quotes

For more information about the customer check-out process, including the details of these workflows, see Siebel eSales Administration Guide.

Siebel Application Service Interface (ASI) is used to integrate order management with third-party credit-check applications. For an overview of how ASIs are used for integrating order management, see *About ASIs for Order Management*. For more information about to how to set up, use, and modify integrations according to ASIs, see Integration Platform Technologies: Siebel Enterprise Application Integration.

Process of Setting Up Purchase Order Credit Checking Applications

To set up Purchase Order Credit Checking Applications, you must perform the following tasks:

1. *Verifying the External Credit Check ASI User Properties*
2. *Configuring Web Services for Purchase Order Credit Checking Applications*
3. *Activating the Workflows for Purchase Order Credit Checking Applications*

To set up Purchase Order Credit Checking Applications, you can perform the following optional tasks:

4. *Modifying User Properties for Purchase Order Credit Checking Integration*
5. *Turning Off Credit Checking*
6. *Customizing When Credit Checks Are Performed*
7. *Changing the Credit Auto-Approval Limit for Accounts*
8. *Skipping Credit Checks for an Account*
9. *Restricting Which Employees Can Administer Credit Checking*
10. *Extending the ASI for Credit Check Interface Request*

> **Note:** If you modify any lists of values such as order types or purchase order payment methods, then be sure to review the workflows processes in Credit Check - Quotes and Credit Check - Orders to verify that the application logic to determine whether to call credit check is still correct.

For more information about to how to set up, use, and modify integrations according to ASIs, see *Siebel Application Services Interface Reference* .

Verifying the External Credit Check ASI User Properties

You must verify the External Credit Check ASI user properties.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

To verify the External Credit Check ASI user properties

1. In Siebel Tools, select the External Credit Check business service.

2. Select Business Service User Prop.
3. Verify that the Names and Values of the User Properties match those in the following table.

Name	Value
siebel_port_name	Default
siebel_web_service_name	External Credit Check
siebel_web_service_namespace	http://siebel.com/asi/

Configuring Web Services for Purchase Order Credit Checking Applications

Web services are used to direct the inbound and outbound messages to and from:

- The third-party credit checking application
- A demonstration script that can be used to test your integration setup
- A middleware or integration server that will integrate with your fulfillment engine and accounts receivable application

You must make sure that the Web services are correctly configured.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

To configure Web services for purchase order credit checking applications

1. Navigate to the Administration - Web Services screen, and then the Outbound Web Services view.
2. Select the row with External Credit Check in the Name field.
3. Verify that the fields in the Outbound Web Services list match those in the following table.

Namespace	Name	Status
http://siebel.com/asi/	External Credit Check	Active

4. In the Service Ports list, click New, and enter the following values in the fields of the new record.

Field	Comments
Name	Enter a name for the port.
Port Type	Enter *External Credit Check*.

Field	Comments
Transport	Select from the list. If you want to set up testing, for example when you are customizing applets or extending the ASI, then select Local Business Service.
Address	When you select a transport, the field is populated with a template address. Edit the address according to whether you are sending the data to middleware, the Credit Check application, and so on. If you want to set up testing, then enter Credit Check ASI Test.
Binding	If a SOAP header is expected by the receiving application, then select SOAP_RPC. If you are using Credit Check ASI Test, then select Property Set.

Activating the Workflows for Purchase Order Credit Checking Applications

For employee credit checking, you must activate the following workflow processes:

- Credit Check - Quotes
- Credit Check - Orders

For customer credit checking in Siebel eSales, you must activate the following workflows:

- eSales - Authorize Order Process
- Credit Check - Quotes

For information about activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

For more information about configuring Web Services and setting up demonstration scripts, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Modifying User Properties for Purchase Order Credit Checking Integration

Some of the user properties of the Credit Check Service business service are used to map integration object fields to their related business component fields. For example, the BC_CREDIT_DATE user property has a value of Credit Check Status As Of Date, which is the name of a business component field on the Quotes and Payments business components.

The PROP_SET_CREDIT_DATE user property has a value of Credit Status As Of, which is the name of an integration object field. Together, these values show that the data in the Credit Status As Of integration object field is moved to the Credit Check Status As Of Date business component field. You do not need to modify these values.

You will need to modify the value of the ERROR_STATUS_CODE user property if it does not match the value of the error code returned from the middleware.

To modify the error status code

1. In Siebel Tools, select Business Service in the Object Explorer.
2. Query for the Credit Check Service business service.
3. Select Business Service User Prop in the Object Explorer.
4. Change the value of the ERROR_STATUS_CODE user property to match the middleware value.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Turning Off Credit Checking

For Siebel eSales, you can turn off credit checking by removing the Credit Check subprocess step from the eSales - Authorize Order Process workflow.

For employee applications, in a quote, credit checking is part of the verify functionality. You can turn it off by changing the Credit Check user property in the Quotes business component. When the Credit Check user property is set to N instead of Y, the verify function will not perform the credit check.

You can also pick what workflow you want to call to execute the credit check by going to Business Component screen, and then the Quote view and changing the following Business Component User Properties:

- For the Credit Check user property, change the value from Y to N.
- For the Credit Check Workflow user property, change the value from Credit Check Quotes to the workflow you want to use.

For employee applications, in an order, the user performs the credit check by clicking the Credit Check button. To turn off credit check for orders, you can either modify the Credit Check - Order workflow or remove the button from the Payment Detail - Purchase Order applet.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Customizing When Credit Checks Are Performed

A credit check does not need to be performed on every quote or order that is paid for with a purchase order. You can customize when a credit check is performed by modifying the Credit Check - Quotes workflow process and the Credit Check - Orders workflow process.

In the default credit check business processes, credit check will only be performed if all the following conditions apply:

- Payment method is Purchase Order
- The bill-to account's Skip Credit Check flag is Off
- The purchase order amount is greater than the bill-to account's Credit Auto Approval Limit amount

In addition, credit check is not applicable for orders of type RMA and Internal Order, so it is skipped.

CAUTION: If you want to modify the Credit Check - Quotes workflow process, then it is important to know that this workflow process is also used by Siebel eSales to create an order.

For more information about the impact of the Credit Check - Quotes workflow process, see the topic about this workflow process in *Siebel eSales Administration Guide* . If you want to have different workflows for Quotes and eSales, then you will need to create an additional workflow.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Changing the Credit Auto-Approval Limit for Accounts

You can change the credit auto-approval limit for accounts in several different ways:

- **Using Siebel Tools.** When your company wants to change the limit for all new accounts, you can change it in Siebel Tools.

In Tools, go to the Credit Limit Amount field on the Account business component. Set a default value for the Credit Limit Amount field. This configuration changes the default for all subsequently created accounts. When creating an account in the application UI, a user can change this default value for the individual account.

Note: Making this change in Siebel Tools does not affect previously existing accounts.

- **Using a batch eScript.** If you need to change the value for a large number of account records, then you can do so with a batch eScript.
- **Editing the value for an account.** To change the limit for a few accounts, you can manually editing the value of the Credit Auto Approval Limit for a specific account in the Account screen, Credit Profile view.

Note: Do not confuse the Credit Auto Approval Limit field with the PO Auto Approval Limit field.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Skipping Credit Checks for an Account

End users can skip automatic credit checking for particular accounts by changing the value of the Skip Credit Check field. This field is displayed in the Account screen, Credit Profile view. When this check box is checked, it indicates that credit checks must be skipped for the account.

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Restricting Which Employees Can Administer Credit Checking

There are two ways to restrict which employees can administer credit checking on accounts:

- Create a read-only version and a read-write version of the Account Detail - Credit Profile view. Give employees who must administer credit checking access to the read-write view. Give employees who must not administer credit checking access to the read-only view.

- Add a rule or logic to restrict who can update fields related to credit checking. For example, only allow an account's primary sales representative to update these fields.

For more information about restricting employees' access to features, see *Siebel Security Guide* .

This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

Extending the ASI for Credit Check Interface Request

If you want to extend the ASI, then you must add fields to:

- The following external and internal interfaces:
  - Credit Check Interface Response
  - Credit Check Interface Request
- The following internal objects:
  - Credit Check Interface Request - Quote
  - Credit Check Interface Request - Order

This will create the necessary mapping between the Credit Check interface structure and the Siebel Quote and Order business components.

If you add a field, then make sure that the names of the fields in the Internal and External integration objects match. For example, if you want to add the field Priority that maps on quote to Priority-quote and on order to Priority-order, then you would set up the names as indicated in the following table.

Object Name	Name	External Name
Credit Check Interface Request	Priority	none
Credit Check Interface Request - Quotes	Priority	*Mapping of the Priority field to the correct field on the Quotes Business Component*
Credit Check Interface Request - Orders	Priority	*Mapping of the Priority field to the correct field on the Orders Business Component*

For information about how to extend the integration object, see *Integration Platform Technologies: Siebel Enterprise Application Integration* . This task is a step in *Process of Setting Up Purchase Order Credit Checking Applications*.

8 Integrating Order Management with Third-Party Taxation Applications

Integrating Order Management with Third-Party Taxation Applications

This chapter describes how to set up and use Taxware applications. It contains the following topics:

- *About Third-Party Taxation Applications*
- *Installing Taxware Applications on Windows*
- *Setting Up Taxware Applications on Windows*
- *Installing and Setting Up Taxware Applications on UNIX*
- *Enabling Tax Calculation*
- *Taxing Customizable Products*
- *Taxware Integration with Siebel eSales*
- *Customizing Taxware Integration*
- *Handling Unicode Conversion Errors*

About Third-Party Taxation Applications

Taxes are calculated on quotes as part of the eSales - Complete Checkout Process workflow. The Calculate Taxes step calls the Tax Calculator business service, which uses the Taxware Adapter Service business service to make a call to Taxware. The calculated sales tax from Taxware is copied to the quote. When the quote is turned into an order, the tax information is copied to the order.

Note: The preconfigured integration allows tax calculations for the many countries supported by Taxware. For the latest list of supported countries, see www.taxware.com.

Taxware Applications

Taxware International is a third-party service used to calculate the sales tax on orders. Siebel eSales is designed to work with the following Taxware modules:

- **Sales/Use.** This module calculates sales and use taxes on purchases made in the United States and Canada. You can configure Sales/Use for product-specific taxation, using the Taxware Sales/Use Toolkit to map the product's part number in the Siebel Business Application to its Tax Code in Sales/Use.
- **WORLDTAX.** This module calculates sales and use taxes on purchases made outside the United States and Canada. You can configure WORLDTAX for product-specific taxation, using the Taxware WORLDTAX Toolkit

to map the product's part number in the Siebel Business Application to its Commodity/Service Code in WORLDTAX.

- **STEP.** This module enables sales tax exemptions and works in conjunction with the Sales/Use module. A Siebel eSales customer is exempt from sales tax if the Siebel account with which the customer is associated has a tax certificate lodged in the STEP module. You implement this feature by entering the Siebel account ID (row ID) in the STEP Customer ID field on the STEP certificate entry form.
- **Universal Tax Link (UTL).** UTL integrates external applications with the Sales/Use and WORLDTAX modules through a common gateway.
- **VERAZIP.** Used for address verification, this module matches state, ZIP Code, city, and county information to make sure that an address is correct and complete, so that the Sales/Use Tax Application can locate the correct taxing jurisdiction.

For more information, see your Taxware documentation.

> **Note:** A product's part number can be found in the Products screen of your Siebel Business Application. You find the row ID of an account by clicking the menu button and selecting About Record with the account selected in the Accounts screen.

Installing Taxware Applications on Windows

Siebel Business Applications support the use of Taxware software on the Windows NT and Windows 2000 operating systems.

The tasks for installing Taxware applications on Windows are as follows:

- Installing Sales/Use on Windows
- Installing STEP on Windows
- Installing UTL on Windows
- Installing WORLDTAX on Windows
- Setting up the Windows environment

> **Note:** The drive on which Taxware applications are installed must have at least 1GB of free disk space. Over time, the sizes of Taxware applications increase as log files and tax certificates are generated.

For information on installing Taxware applications, see the Taxware documentation. For information on which versions of Taxware are supported by Siebel Business Applications, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

> **Note:** For Siebel CRM product releases 8.1.1.9 and later and for 8.2.2.2 and later, system requirements and supported platform certifications are available from the Certification tab on My Oracle Support.

Setting Up Taxware Applications on Windows

After installation, you must do certain tasks, including setting an administrator user name and password, checking environment variables, loading files, and putting your company information into the application.

The tasks for setting up Taxware applications you installed on Windows are as follows:

- Setting Up Sales/Use on Windows
- Setting Up STEP on Windows
- Setting Up WORLDTAX on Windows

For information on setting up Taxware applications, see your Taxware documentation.

Installing and Setting Up Taxware Applications on UNIX

Siebel Business Applications support the use of Taxware software on Oracle Solaris and AIX operating systems.

The tasks for installing and setting up Taxware applications on UNIX are as follows:

- Installing Taxware Applications on UNIX
- Setting Up Sales/Use on UNIX
- Setting Up WORLDTAX on UNIX

Note: The drive on which Taxware applications are installed must have at least 1GB of free disk space. Over time, the sizes of Taxware applications increase as log files and tax certificates are generated.

After installation, you must do certain tasks, including setting an administrator user name and password, checking environment variables, loading files, and putting your company information into the application.

For information on installing and setting up Taxware applications, see the Taxware documentation. For information on which versions of Taxware are supported by Siebel Business Applications, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Note: For Siebel CRM product releases 8.1.1.9 and later and for 8.2.2.2 and later, system requirements and supported platform certifications are available from the Certification tab on My Oracle Support.

Enabling Tax Calculation

To use Taxware applications with Siebel eSales, you must set certain parameters for the eSales Object Manager component of the Siebel Server.

This topic contains the following subtopics:

- *Enabling Tax Calculation with the Sales/Use Tax System*

- *Enabling Tax Calculation with WORLDTAX*

Enabling Tax Calculation with the Sales/Use Tax System

To use the Sales/Use Tax System with Siebel eSales, you need to have company information defined in the Sales/Use Tax System Toolkit and on the Siebel Server.

To enable tax calculation with the Sales/Use Tax System

1. In the Sales/Use Tax System Toolkit, define your Company ID and Company Location, for example *SIEBEL* and *SAN MATEO*.
2. In your Siebel employee application, navigate to Administration - Server Configuration screen, Servers, Components, and then Parameters view.
3. In the Components list, select eSales Object Manager.
4. In the Component Parameters list, set values for the eSales Object Manager parameters listed in the table that follows.

Parameter	Value
Company ID for Sales Tax	Your company's sales tax ID
Company Location for Sales Tax	Your company's sales tax location
Enable Sales Tax	TRUE
Sales Tax Type	Taxware

Enabling Tax Calculation with WORLDTAX

To use the WORLDTAX application with Siebel eSales, you need to have company information defined in the WORLDTAX Toolkit and on the Siebel Server.

To enable tax calculation with WORLDTAX

1. In the WORLDTAX Toolkit, define your Company ID and Business Location, for example *SIEBEL_FR* and *FR* (France).
2. In your Siebel employee application, navigate to Administration - Server Configuration screen, Servers, Components, and then Parameters view.
3. In the Components list, select eSales Object Manager.

4. In the Component Parameters list, set values for the eSales Object Manager parameters listed in the table that follows.

Parameter	Value
Company ID for Sales Tax	Your company's sales tax ID
Company Location for Sales Tax	Your company's sales tax location
Enable Sales Tax	TRUE
Sales Tax Type	Taxware

5. In Siebel Tools, modify the user properties of the Tax - Configuration (Taxware) integration object, shown in the following table.

Integration Object User Property	Value
SHIPFROMCOUNTRY	ISO country code of the ship-from country (for example, FR for France)
WorldTaxInstalled	Y

SHIPFROMCOUNTRY can also be passed dynamically by modifying the Tax Integration (Siebel Quote Output) integration object and the data map to the Tax Integration (Taxware Input) integration object.

Taxing Customizable Products

Tax calculation is more complicated for customizable products because not all product components may need to be taxed. By default, for tax calculation, Siebel eSales submits the price of the product that is rolled up from its components. This means that if some of the component items do not need to be taxed, then the tax calculation is inaccurate.

You have the option of calculating tax at the component level for any customizable product. For information about setting up taxation at the component level, see Siebel Product Administration Guide.

Taxware Integration with Siebel eSales

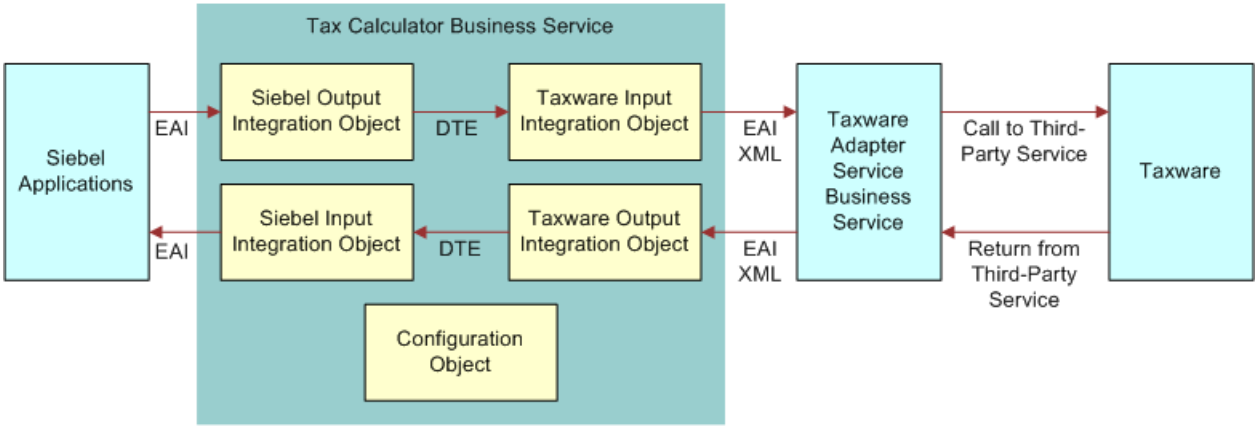
Integrating Siebel eSales with third-party taxation applications involves two business services:

- Tax Calculator. See *Tax Calculator Business Service*.

- Taxware Adapter Service. See *Taxware Adapter Business Service*.

Tax Calculator Business Service

The Tax Calculator business service controls the process shown in the following image.



The steps in the tax calculation process flow are as follows:

- Siebel EAI passes data from a quote record and the method name to a Siebel output integration object.
- The Siebel Data Transformation Engine maps the Siebel output integration object to a Taxware input integration object.
- The Siebel EAI XML Converter converts the Taxware input integration object into XML, using the CC XML Converter Business Service, and passes the XML data and method name to the Taxware Adapter Service.
- The Taxware Adapter Service business service processes the data and uses Siebel eScript to call the Taxware API.
- The Taxware Adapter Service gets the calculated sales tax from Taxware using the Taxware API.
- The Taxware Adapter Service formats the sales tax data into an XML string.
- The Siebel EAI XML Converter converts the XML into a Taxware output integration object.
- The Siebel Data Transformation Engine maps the Taxware output integration object to a Siebel input integration object.
- Siebel EAI updates the quote record with the calculated sales tax.

The Tax Calculator business service has one method, CalculateTax, with the following input arguments:

- Siebel to Vendor Map Name
- Vendor to Siebel Map Name

The Tax Calculator business service user properties are used to set the values given in the following table.

Business Service User Property	Value
Taxware:Adapter Business Service	Taxware Adapter Service
Taxware:Configuration Object	Tax - Configuration (Taxware)
Taxware:Input IntObject Name	Tax Integration (Taxware Input)

Business Service User Property	Value
Taxware:Output IntObject Name	Tax Integration (Taxware Output)

For more information on business services, see *Using Siebel Tools* .

Integration Objects

Integration objects pass static or dynamic values to the business service. The architecture is modular in that you can create and use different integration objects for the same core business service. For example, while the preconfigured integration supports the Quote business component, you can create integration objects to calculate taxes directly on the Order business component.

The Tax Calculator business service uses the following integration objects:

- **Tax - Configuration (Taxware).** Its user properties are shown in the following table.

Integration Object User Property	Value	Comments
AUDFILEIND	2	Audit File Indicator: • 1: Audit File/Suspense File • 2: No Record stored on file • 3: Suspense File • 4: Audit File
EXEMPTUSEFLAG	Y	Exemption Use Flag: • Y: Call STEP, use default record. • L: Call STEP, do not use default record. • P: Call STEP, search by product code. • D: Call STEP, search by reason code and accept default customer. • N: Do not call STEP.
InputIntObjectName	Tax Integration (Taxware Input)	None
OutputIntObjectName	Tax Integration (Taxware Output)	None
SHIPFROMCOUNTRY	US	None
SalesUseInstalled	Y	None
VendorDll	taxcommono	Name of the shared library.

Integration Object User Property	Value	Comments
WorldTaxInstalled	N	None

- **Tax Integration (Siebel Quote).** Its integration components are shown in the following table.

Integration Component	Fields
Quote	Account Id
	Currency Code
	Current Quote Ship To City
	Current Quote Ship To Country
	Current Quote Ship To Postal Code
	Current Quote Ship To State
	EAI Error Message
	Quote Number
	Tax Amount
Quote Item	Extended Line Total
	Extended Quantity
	Line Number
	Line Total
	Part Number
	Quantity
	Row Id
	Sales Product Flag
	Sales Service Flag

Integration Component	Fields
	Tax Amount
	Tax Subcomponent Flag
	Taxable

- **Tax Integration (Taxware Input).** Its integration components are shown in the following table.

Integration Component	Fields That Pass Data from Siebel
Invoice	Currency Code
	Customer Id
	Document Number
	Invoice Number
	Ship To City
	Ship To Country
	Ship To Postal Code
	Ship To State
Invoice Item	Line Number
	Line Total
	Part Number
	Quantity
	Sales Product Flag
	Sales Service Flag
	Ship to City

Integration Component	Fields That Pass Data from Siebel
	Ship to Country
	Ship to Postal Code
	Ship to State

See your Taxware documentation for more fields that can be passed. For information on how to pass more fields to Taxware, see *Passing Data from Other Fields in the Siebel Database*.

- **Tax Integration (Taxware Output).** Its integration components are shown in the following table.

Integration Component	Fields That Pass Data from Taxware		
Invoice	City Tax Amount County Tax Amount Error Message Federal Tax Amount Invoice Number Secondary City Tax Amount Secondary County Tax Amount Secondary State Tax Amount State Tax Amount		
Invoice Item	City Basis Amount	City Completion Code	City Tax Amount
	City Tax Rate	County Basis Amount	County Completion Code
	County Tax Amount	County Tax Rate	Federal Basis Amount
	Federal Completion Code	Federal Tax Amount	Federal Tax Rate
	General Completion Code	General Completion Code Desc	Line Number
	Secondary City Basis Amount	Secondary City Completion Code	Secondary City Tax Amount
	Secondary City Tax Rate	Secondary County Basis Amount	Secondary County Completion Code
	Secondary County Tax Amount	Secondary County Tax Rate	Secondary State Basis Amount

Integration Component	Fields That Pass Data from Taxware		
	Secondary State Completion Code	Secondary State Tax Amount	Secondary State Tax Rate
	State Basis Amount	State Completion Code	State Tax Amount
	State Tax Rate		

See your Taxware documentation for more fields that can be passed.

For more information on integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration* and *Using Siebel Tools* .

Data Maps

Data maps are used to map fields in one integration object to another integration object. You can view data maps by navigating to the Administration - Integration screen, and then the Data Maps view.

The data maps used in tax calculation, shown in the following table are specified in the call to the Tax Calculator business service.

Name	Source Object Name	Target Object Name
TaxCalculation_QuoteToTaxware	Tax Integration (Siebel Quote)	Tax Integration (Taxware Input)
TaxCalculation_TaxwareToQuote	Tax Integration (Taxware Output)	Tax Integration (Siebel Quote)

For more information on modifying integration objects and data maps, see *Integration Platform Technologies: Siebel Enterprise Application Integration* and *Using Siebel Tools* .

Taxware Adapter Business Service

The Taxware Adapter Service business service:

- Receives input from the Tax Calculator business service in XML format.
- Passes the following information to Taxware:
  - Static information from the Tax - Configuration (Taxware) integration object
  - Dynamic information from the Tax Integration (Taxware Input) integration object
- Receives information back from Taxware.
- Passes XML data back to the Tax Calculator business service.

The Taxware Adapter Service consists of the following methods:

- **CalculateTax.** This is the main method for calculating the tax. It converts the input XML string to a property set and calls the Taxware software to get the tax information as a property set. It then converts the tax information to an XML string as an output.
- **Initialize.** This method uses the input property set to initialize the variables for the business service.
- **OpenClose.** This method opens and closes files used by Taxware to calculate taxes.

Audit File

Taxware uses an Audit File to store transactional information for tax reporting purposes. The Audit File for Sales/Use is a text file that you can edit. There is a special editor for the WORLDTAX Audit File that marks transactions that have been altered.

When calculating taxes in Siebel eSales, set the audit file indicator (AUDFILEIND) to 2 (No Record) because the customer has not yet confirmed the order.

To write to the audit file, a separate call to Taxware must be made from the order. This setup requires creating Order integration objects and mappings to the Taxware integration objects.

You can set the value for AUDFILEIND by changing the value in the Tax - Configuration (Taxware) integration object user properties. For orders, you can set a separate default value in the data mapping.

Address Verification for Taxware

Completion codes are returned that indicate the status of the process running UTL if one of the following occurs:

- The customer enters an improper city/state/zip combination.

  In most cases the completion code appears, indicating that a jurisdiction error occurred, which means something was wrong with the zip/state combination.
- The customer enters a city/state/zip combination that maps to more than one jurisdiction.

  Taxware defaults to the most likely match and applies that rate for the transaction. The application generally uses the jurisdiction with the highest population, as rates for heavily populated areas are usually higher than those for areas with less population. However, the default rate used is not always the highest rate out of the possible matches.

Customizing Taxware Integration

You can customize Siebel eSales integration with Taxware. Some examples are passing more fields to Taxware, and performing tax calculations on data in other business components.

This topic contains the following subtopics:

- *Passing Data from Other Fields in the Siebel Database*
- *Passing Data from Fields Not in the Siebel Database*
- *Calling the Tax Calculator Business Service from Other Places in Siebel eSales*
- *Calling the Tax Calculator Business Service from Other Business Components*

Passing Data from Other Fields in the Siebel Database

You can add data fields to those passed to Taxware. You must add the desired field to the appropriate integration objects and map the fields from your Siebel Business Application to Taxware, and from Taxware to your Siebel Business Application.

For more information on modifying integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration* . For more information on modifying scripts, see *Using Siebel Tools* .

To add a data field

1. In Siebel Tools, add the field to the appropriate integration objects.
2. Activate the appropriate fields in Tax Integration (Taxware Input) and Tax Integration (Taxware Output) integration objects.
3. In your Siebel Business Application, navigate to the Integration Administration screen, and then the Data Map Editor.
4. Update the data map so that the appropriate Siebel data is passed to Taxware and an appropriate result is passed back to the Siebel database.

Passing Data from Fields Not in the Siebel Database

There are three ways to pass data from fields that do not exist in the Siebel database:

- Add the field to the database, add it to the necessary integration object, and then update the data map.
- Set an existing business service user property in the Tax Calculator business service, or create a new one, with a static value for all transactions. These values are read when the business service is initialized.
  Any data passed dynamically using the integration objects will overwrite the value in the business service user property for that particular transaction.
- Set a value for an integration object field by setting the value in the data map.
  You cannot set a default value for an integration object user property because the Tax Calculator business service does not look for this value at initialization.

Calling the Tax Calculator Business Service from Other Places in Siebel eSales

You can call a business service from any of the following:

- Button, if the applet the button is in is of the CSSBCBase class
- Workflow
- Script, either Siebel eScript or Siebel VB

For information on configuring applications and calling business services, see *Using Siebel Tools* .

To call the Tax Calculator business service from other places in Siebel eSales

1. Use the GetService() method to get an instance of the business service.

2. Use Service_InvokeMethod() to call a method of the business service.
3. Use SetProperty() to pass values for the business service method arguments that need to be set.

   See *Tax Calculator Business Service* for the arguments of the CalculateTax method of the Tax Calculator Business Service.

Calling the Tax Calculator Business Service from Other Business Components

You can call the Tax Calculator business service from other business components besides Quote, for example Order. You can use the existing Tax Integration (Taxware Input) and Tax Integration (Taxware Output) integration objects for mapping.

To call the Tax Calculator business service from an order

1. Choose where to call the business service (button, workflow, or script).
2. In Siebel Tools, create integration objects for the Order business component to pass the correct data to the business service.
3. In your Siebel employee application, navigate to the Administration - Integration screen, and then the Data Map Editor.
4. Create integration object maps for the Order integration objects and the Taxware integration objects.

Handling Unicode Conversion Errors

Taxware applications use the ASCII codepage. Fields that are passed from Siebel eSales to Taxware applications are therefore converted from Unicode UTF-16 characters to ASCII characters.

This topic contains the following subtopics:

- *Error Prevention During Codepage Conversion*
- *Defining Required Fields in Codepage Conversion*
- *Configuring Error Messages in Codepage Conversion*

For general information about Unicode support in Siebel Business Applications, see Siebel Global Deployment Guide.

Error Prevention During Codepage Conversion

If non-ASCII characters are entered into a required field, then an error is generated during the codepage conversion. The error is contained in the Error Message field of the integration object. The order will not be stopped by an error, but an employee will have to manually calculate the tax. It is therefore important to:

- Generate guidance messages and error messages to prevent customers from entering non-ASCII characters in a required field. A list of required fields is given in the following table.
- Only use lists of values that contain ASCII-compliant values.

Taxware Field	Siebel Field	Siebel Location	Required in Sales/Use	Required in WorldTax
Company ID	Company ID	Configuration file	Yes	Yes
Ship-From Country Code	Ship From Country	Business service integration object	Yes	Yes
Destination/Ship-To Country Code	Ship To Country	Database	Yes	Yes
POO Country Code	From Taxware	Business service integration object	Yes	Yes
POA Country Code	From Taxware (Defaulted to the setting in the Taxware Toolkit.)	Business service integration object	Yes	Yes
Line Item Amount	Line Total	Database	Yes	Yes
State Code	Ship To State	Database	Yes	No
Zip Code	Ship To Postal Code	Database	Yes	No
Document Number	Quote #	Database	No	Yes
Tax Point	Set in Adapter business service	Set in Adapter business service	No	Yes
Transaction Type	Set in Adapter business service (value of 1)	Set in Adapter business service (value of 1)	No	Yes
Seller Registration	From Taxware	Set in the Taxware Toolkit. You can pass dynamic data by configuration.	No	Yes
WT Code	Siebel Sales Product Flag (set to 01) or Siebel Service Product Flag (set to 23)	Siebel Sales Product Flag (set to 01) or Siebel Service Product Flag (set to 23)	No	Yes
Commodity Code	Part Number	Database	No	Yes
Primary Currency Code	Currency Code	Database	No	Yes

Defining Required Fields in Codepage Conversion

You can define which fields that are passed to Taxware are required and whether a field is required for WorldTax, Sales/Use, or both applications.

To define required fields in codepage conversion

1. In Siebel Tools, from the View menu, select Options.
2. In the Development Tools Options dialog box, select Object Explorer.
3. Move Integration Object to the far pane of the dialog box and click OK.
4. Select the integration object Tax Integration (Taxware Input).
5. Select Integration Component Field from the Object Explorer.
6. To make a field required, check the External Required column.
7. To make a field required for:
    - Both WorldTax and Sales/User, leave the External DataType as String.
    - WorldTax only, change the External DataType to StringWT.
    - Sales/User only, change the External DataType to StringSU.

Configuring Error Messages in Codepage Conversion

You can perform the following configurations related to codepage errors:

- You can configure the eSales - Complete Checkout Process workflow so that the order process stops when a codepage error is returned by Taxware. The default behavior is to allow the order process to continue.
- You can configure the error messages displayed in the Quote Summary and Order Confirmation views of your employee application.
- You can change the error messages that are displayed to customers on the Quote Form applet - Summary (eSales) applet and Order Form Applet - Confirmation (eSales) applet of your Siebel eSales Web site.

To configure error messages in codepage conversion

1. In Siebel Tools, select either the Quote Form - Summary (eSales) applet or Order Form Applet - Confirmation (eSales) applet.
2. Select Applet User Prop in the Object Explorer.
3. Replace the EAI Error Message text with the text of your choice in the user property that has the value LabelTaxNotCalculated.

9 Integrating Order Management with Oracle ATG Web Commerce

Integrating Order Management with Oracle ATG Web Commerce

This chapter describes how to integrate Siebel Business Applications with Oracle ATG Web Commerce. It includes the following topics:

- *About Integrating Siebel Business Applications with Oracle ATG Web Commerce*
- *Setup Tasks for Integrating Siebel Business Applications with Oracle ATG Web Commerce*
- *Process of Exporting Records to Oracle ATG Web Commerce*
- *Maintaining Product Data for Integration with Oracle ATG Web Commerce*

About Integrating Siebel Business Applications with Oracle ATG Web Commerce

Siebel Business Applications have special features for offline or assisted channels, such as contact centers and partners, which enable you to manage core business data, such as customers and products, and to integrate presales and postsales processes.

Oracle ATG Web Commerce has special features for online channels, such as Web stores, mobile devices, and kiosks. These features provide the performance, flexibility, and customer experience that are required for online channels.

Integrating Siebel Business Applications with Oracle ATG Web Commerce provides the following benefits:

- Provides a combined solution that gives you a unified view of the customer and enables you to work on all channels.
- Enables you to harness the full power of each application.
- Enables you to integrate all objects in the Siebel product catalog entities with ATG Web Commerce, while supporting the full, life-cycle requirements of these entities

For more information about integrating Siebel Business Applications with Oracle ATG Web Commerce, see 1358285.1 (Article ID) on My Oracle Support, *Siebel ATG Implementation Guide*.

Related Topics

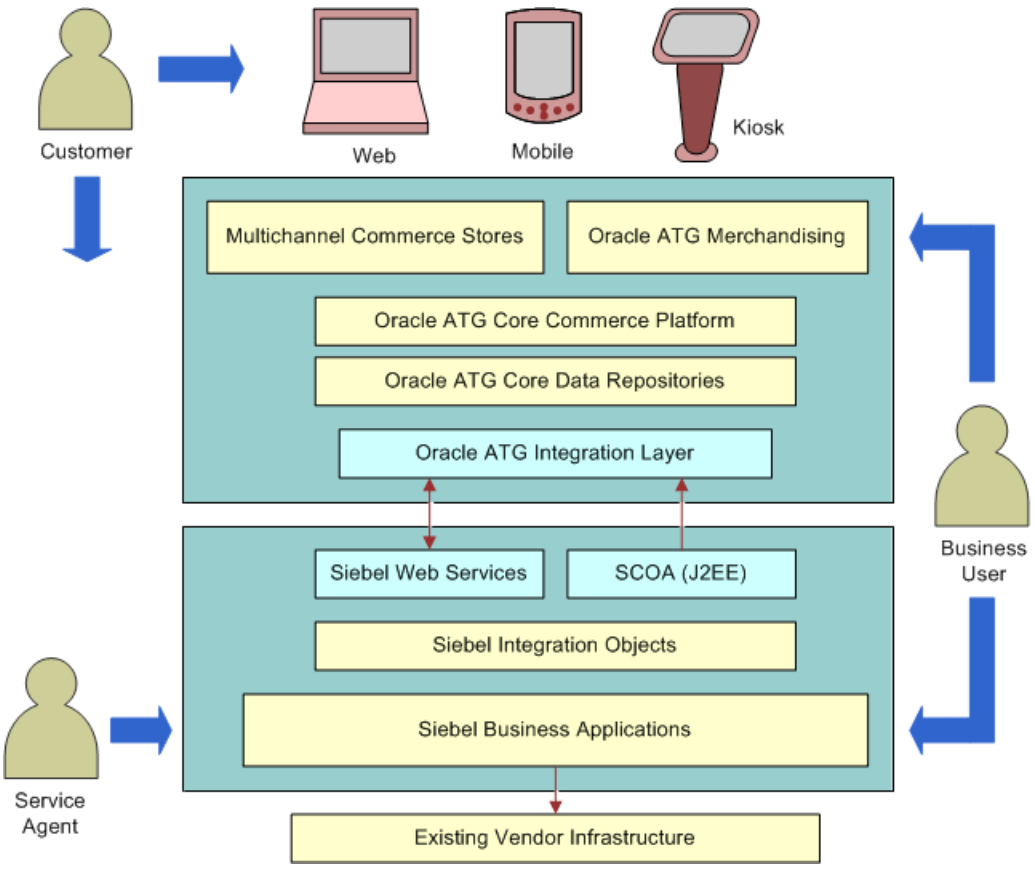
Architecture of the Integration of Siebel Business Applications with Oracle ATG Web Commerce

About Siebel Commerce Object Aggregator

About Exporting Siebel Product Catalogs to Oracle ATG Web Commerce

Architecture of the Integration of Siebel Business Applications with Oracle ATG Web Commerce

The following figure shows the overall architecture of the integration of Siebel Business Applications with Oracle ATG Web Commerce.



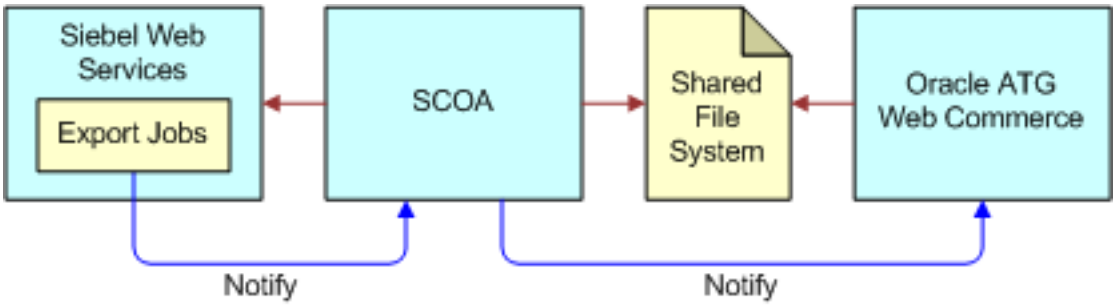
The important elements of this architecture include:

- **ATG Integration Layer.** For more information, see *ATG Web Commerce and Siebel Reference Integration*, document Id 1358285.1 on My Oracle Support.
- **Siebel Web Services.** For more information, see *Siebel CRM Web Services Reference* .
- **Siebel Commerce Object Aggregator (SCOA).** This application acts as an intermediary between Siebel Web services and ATG Web services. For more information, see *About Siebel Commerce Object Aggregator*.

About Siebel Commerce Object Aggregator

Siebel Commerce Object Aggregator (SCOA) is a Java 2 Platform, Enterprise Edition (J2EE) Web application used to synchronize Siebel products, catalogs and price lists with ATG Web Commerce. SCOA is a reference application and is delivered as source code for each customer or system integrator to modify and compile. Oracle expects customers to modify the Java code of SCOA to suit their middleware needs.

The following figure shows that SCOA acts as an intermediary between Siebel Web services and ATG Web services.



SCOA does the following:

- Optimizes Siebel Web commerce job details
- Stores the result of this job in an in-memory Java database
- Invokes inbound Siebel Web Services to request Siebel contents for each Export Job Detail record
- Aggregates replies from the Siebel application using Java technologies such as Java Objects or XLST/XML transformation
- Creates XML output files in a designated directory under the job name, where ATG Web Commerce integration layer retrieves these files

SCOA uses the following Siebel Web services:

- CatalogWebService
- ISS Web Commerce Export Attributes
- ISS Web Commerce Export Class
- ISS Web Commerce Export Contents
- ISS Web Commerce Export Error
- ISS Web Commerce Export Price Lists
- ISS Web Commerce Export Recommended Products
- ISS Web Commerce Export Volume Discounts
- SCOAExportWS

These Web services are all inbound except SCOAExportWS. For more information about these Web services, see *Siebel CRM Web Services Reference* .

SCOA uses the following ATG Web services:

- importFromSiebel
- getJobStatus

SCOA is delivered as reference source code and can be compiled into a Web application repository file. It can be deployed on any full scale J2EE application server or on any Java application server that runs servlet containers.

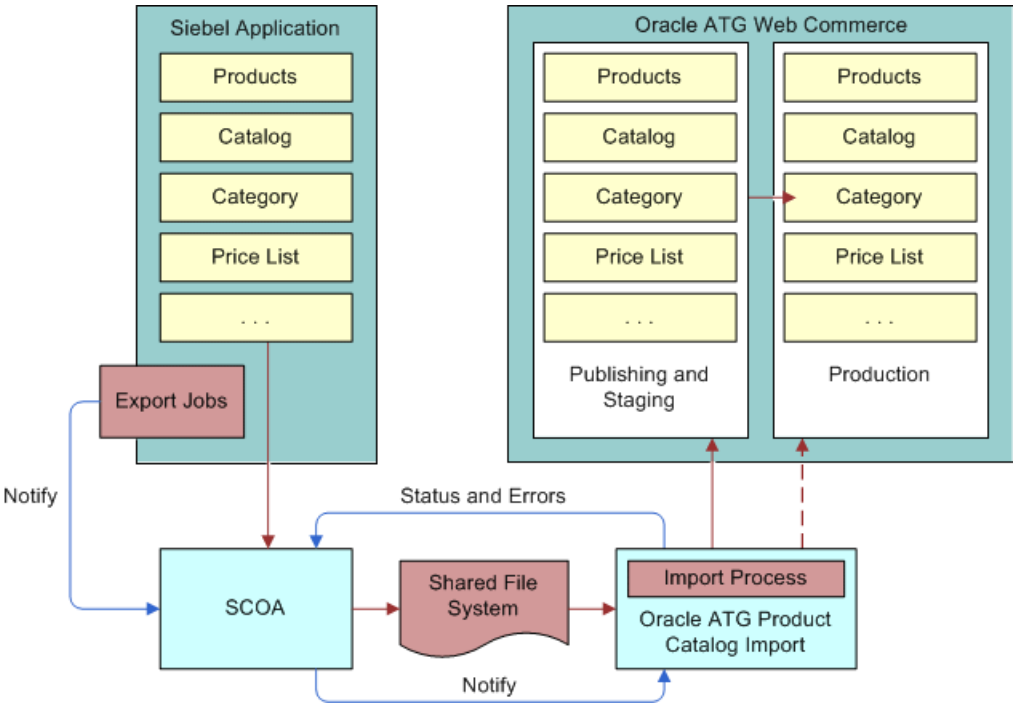
About Exporting Siebel Product Catalogs to Oracle ATG Web Commerce

The following figure shows how Siebel product catalogs are exported to ATG Web Commerce.

Note: The administrator must manually add entities used by the Siebel Product Catalog, such as price lists, products, volume discounts, and so on, to a Web Content Export Content Job, in order to make these associated entities available in ATG Web Commerce.

When you export product catalogs, the Siebel application:

- Aggregates Siebel Product Catalog entities into an export job
- Exports any level of Siebel structured object. The functionality ensures that the parent object exists in ATG Web Commerce.
- Supports both product initialization and product updates of entities in the Siebel product catalog
- Supports lifecycle changes of these entities, with the object in the Siebel application viewed as the master
- Includes the full structure of exported promotions and configurable products
- Automatically exports a product's class when you export the product
- Automatically exports a price list line item's Volume Discount when you export the line item



Setup Tasks for Integrating Siebel Business Applications with Oracle ATG Web Commerce

You must install Siebel Commerce Object Aggregator.

In addition, you can use a number of system preferences to set up the integration of Siebel Business Applications with Oracle ATG Web Commerce. These setup tasks are not required. The Siebel application exports records to Oracle ATG Web Commerce if you do not change the default values of these system preferences. However, you might have to change these system preferences to make the Siebel application behave in the way that you want.

These installation and setup tasks are described in the following topics:

- *Installing Siebel Commerce Object Aggregator*
- *Enabling ATG Web Commerce Integration*
- *Enabling Cascading Deletions of Web Content Export Job Details*
- *Enabling the Export of Nested Catalog Hierarchies to Oracle ATG Web Commerce*
- *Creating Separate Web Content Export Jobs for Each User*

Installing Siebel Commerce Object Aggregator

You must install Siebel Commerce Object Aggregator (SCOA) before setting up ATG Web Commerce Integration.

To install Siebel Commerce Object Aggregator

1. Navigate to the REPPATCH directory of your Siebel Tools installation.
2. Download and extract the file SCOA-Snapshot.zip.

   This file contains documentation about how to install SCOA.

Enabling ATG Web Commerce Integration

The System Preference SiebelCommerceInteg is set to true by default. This is the value that is needed, so you do not have to change it to enable the integration of Oracle ATG Web Commerce.

The Siebel application checks the value of SiebelCommerceInteg when one of the Siebel product catalog entities is physically deleted by the administrator. When this system preference is set to TRUE, the Siebel application inserts a record into the open Job with a type of *<entity>-Delete*.

For example, if a user deletes a category from a catalog, and this preference is TRUE, then the Siebel application finds an open Export Job (or creates one) and inserts a new record with a type of *Category-Delete*. Runtime events for each of the Siebel product catalog business components control this behavior through conditional expressions.

Thus, this system preference helps to keep the data in Oracle ATG Web Commerce synchronized with the data in the Siebel application, by deleting entities in Oracle ATG Web Commerce if they are deleted in the Siebel application.

The following business components contain logic in the PreDeleteRecord Runtime Event that checks the SiebelCommerceInteg system preference. If the system preference is TRUE, then the application does not insert a marker record in the Web Commerce Export Details when a user physically deletes a record in one of these business components:

- Product Catalog
- Catalog Category Admin
- Internal Product
- Price List
- Price List Item
- TNT SHM Volume Discount
- Volume Discount

- Volume Discount Item
- ISS Promotion
- Key Feature
- Sales Tool by Product
- Product Recommendation

If this system preference is mistakenly set to False, rather than the default value of True, then you can enable integration once again using the following procedure.

To enable ATG Web Commerce integration

- Set the system preference SiebelCommerceInteg to True.

To modify specific entity behavior for record deletion

1. Query the runtime events for the business components listed in *Enabling ATG Web Commerce Integration*, or query for `%InsertMarkerRecord%` in the Action Set Name List Column of the view Administration - Runtime Events, Events view.
2. Call a custom business service for desired behavior.

Enabling Cascading Deletions of Web Content Export Job Details

The system preference, SiebCommIntPhysicalDeletes, determines whether there is a cascading deletion of related export job details when the reference record (Catalog or Category record) is physically deleted from the business component. The default value of this system preference is N, and you must change it to enable cascading deletions.

If this system preference is set to Y, then all Web commerce export details of the open export job that are of type Category or Category Product are physically deleted from the job when the reference record is deleted. The deletion of the child records occurs in a cascade format where the parent field is set to object name.

To enable cascading deletions of Web content export job details

- Set the system preference SiebCommIntPhysicalDeletes to Y.

Enabling the Export of Nested Catalog Hierarchies to Oracle ATG Web Commerce

The system preference, SiebCommExpCtlgCtgyHier, determines whether the full category hierarchy is exported when you create Web content export jobs. The default value is N. You must change the value to Y to use a full catalog export if the Siebel catalog contains many layers of nested categories.

To enable the export of nested catalog hierarchies to Oracle ATG Web Commerce

- Set the system preference SiebCommExpCtlgCtgyHier to `Y`.

  Note: Setting this System Preference to Y may have a performance impact on jobs with many detail records.

Creating Separate Web Content Export Jobs for Each User

The system preference, SiebCommIntOpenJobSearchSpec, determines which export job header the export job detail records are added to. The system preference applies to the business component ISS Web Commerce Export Contents - Header. The default value is as follows:

```
[Status] = "Open"
```

This setting means that the detail records are added to the header record that is open. If you want each user to have an export job header of his or her own, with only the details that the user creates, then you can change this search specification and filter the search by user.

To create separate Web content export jobs for each user

- Set the value of the system preference SiebCommIntOpenJobSearchSpec as follows:

```
[Created By] = LoginName() and [Status] = "Open"
```

Process of Exporting Records to Oracle ATG Web Commerce

Exporting records to ATG Web Commerce requires the following:

- **Web content export job details.** You must create one or more Web content export job details by selecting and exporting records in the Administration - Product, Administration - Pricing or Administration - Catalog screen.
- **Web content export job header.** You must submit the Web content export job header that has these Web content export job details as line items.

The Siebel application automatically creates and maintains Export Contents - Header records in the Administration - Order Management screen. When you create Web content export job details, the Siebel application automatically adds them to an Export Contents - Header record as line items.

The appropriate header record is found based on the search specification for SiebCommIntOpenJob, as described in *Creating Separate Web Content Export Jobs for Each User*. If none is found, then it creates one base on the value.

Each user can have only one Web content export job header that has not been submitted. If you do not have any unsubmitted Web content export job headers, then the Siebel application automatically creates one when you create a Web content export job detail and adds the Web content export job detail to the job header as a line item. If you do have an unsubmitted Web content export job header, then the Siebel application adds Web content export job details that you create to this header as additional line items.

To export records from the Siebel application to ATG Web Commerce, perform the following tasks:

- Creating Web content export job details in one or more of the following ways:

- *Creating a Web Content Export Job Detail for Siebel Products or Promotions*
- *Creating a Web Content Export Job Detail for Siebel Catalogs*
- *Creating a Web Content Export Job Detail for Siebel Price Lists*

- *Submitting an Export Contents Job*
- *Correcting Errors in an Export Contents Job*

Creating a Web Content Export Job Detail for Siebel Products or Promotions

You may include any type of product in an export job, without limitation based on data value. Because Siebel Business applications treat promotions as a type of product, you can export promotions in the same way that you export products.

The following best practices are recommended:

- Use Siebel Query By Example to construct predefined queries used to build the set of products to send to ATG.
- Make sure the resulting set of products is of reasonable size, because larger record sets can negatively impact performance. It is best to build jobs with a few hundred records or less.

You can create an export job by using the Update Commerce App button or menu option on the following views of the Administration - Product screen:

- Products view (button)
- Product Definitions view (button)
- Any view that includes the Product Form as its header, which you navigate to by clicking the view tabs in the Products view (menu option)

The following procedure uses the Products list as an example, but you can also perform the same task in the other views.

To create a Web content export job detail for Siebel products

1. Navigate to the Administration - Product screen, then the Products view.
2. Query for the products that you want.
3. Select or make a multiple selection of the records of the products that you want.
4. Click Update Commerce App.
5. Optionally, select or deselect the following check boxes in the Update Commerce App dialog box:
    - **Include Related Products.** If this check box is selected, then the Siebel application checks whether there are any upsell recommendations or cross-sell recommendations for selected root products. If recommendations exist, then the Siebel application creates an Export Job - Detail record of Type = Product for each associated product recommendation. It also sets the Include Related Products flag to True, so SCOA calls the product recommendation's Web service to include the appropriate associated products for the root product.
    - **Full Structure.** This check box is selected by default. If it is selected, then SCOA retrieves the full structure of products, both breadth (attributes of the classes that products belong to) and depth (the full hierarchy of each product or promotion), and the Siebel application sends this structure to ATG. This functionality applies to only product definitions that have a structure, and it ignores nonstructured products.

> **Note:** Export Full Structure does not send the product definition of each component of the hierarchy to ATG. You must manually create an Export Job - Detail record for each component of the hierarchy.

- **Include Collateral.** If this check box is selected, then the export includes images, key features, literature, and news that are attached to the products in the Administration - Product screen, Collateral view.

6. In the dialog box, click Export.
7. Review the message that indicates the number of records that have been sent for export.

Creating a Web Content Export Job Detail for Siebel Catalogs

To export nested Siebel categories correctly, you must set the system preference SiebCommExpCtlgCtgyHier to Y. This setting turns on the functionality that exports all children of a given level within the catalog hierarchy as well as all parents. For more information, see *Enabling Cascading Deletions of Web Content Export Job Details*.

Only catalogs of the type Buying can be exported to Oracle ATG Web Commerce. If an export job includes catalogs of other types, then SCOA does not create XML files for these nonbuying catalogs, so they are not exported to ATG Web Commerce.

The following best practices are recommended for exporting catalogs:

- Try to aggregate related commerce data by creating Web content export job details for catalogs and for the corresponding products and price lists that are line items of one separate, export contents job.
- Create exports at the appropriate level of granularity. For example, if many products in a category have been modified, then export the category. If only a few products in a category have been modified, then export only those products
- Make sure the resulting set of catalogs is of reasonable size, because larger record sets can negatively impact performance. It is best to build jobs with a few hundred records or less.

You can create an export job by using the Update Commerce App button or menu option on the following views of the Administration - Catalog screen:

- Catalogs view (button)
- More Info view (button)
- Smart Answer Settings view (button)
- Categories view (button)
- Access Groups view (button)
- Catalog Translations view (button)
- Catalog Detail view (menu option)
- Category Detail view (menu option)

The following procedure uses the Catalogs view as an example, but you can also perform the same task in the other views.

> **Note:** Adding a catalog to a job does not automatically add the Product records of the catalog to the job. An administrator must manually add the products of the catalog to the job.

To create a Web content export job detail for Siebel catalogs

1. Navigate to the Administration - Catalog screen, then the Catalogs view.
2. Query for the catalogs that you want.
3. Select or make a multiple selection of the records of the catalogs that you want.
4. Click Update Commerce App.
5. In the Update Commerce App dialog box, click Export.
6. Review the message that indicates the number of records that have been sent for export.

Creating a Web Content Export Job Detail for Siebel Price Lists

You can include any price list in an export job without any limitations that are based on data values.

The following best practices are recommended for exporting catalogs:

- Try to aggregate related commerce data by creating Web content export job details for price lists and for corresponding products and catalogs that all are line items of one separate export contents job.
- Create export jobs at the appropriate level of granularity. For example, if many line items in a price list have been modified, then export the price list. If only a few line items in a price list have been modified, then export just those line items.
- Make sure the resulting set of price lists is of reasonable size, because larger record sets can negatively impact performance. It is best to build jobs with a few hundred records or less.

You can create an export job using the Update Commerce App button or menu option on the following views of the Administration - Pricing screen:

- Price Lists view (button)
- More Info view (button)
- Price List Line Items view (button)
- Service Pricing view (button)
- Volume Discounts view (button)
- Volume Discount Line Items view (menu option)

The following procedure uses the Price List view as an example, but you can also perform the same task in the other views.

> **Note:** Adding a price list to a job does not automatically add the Product records of the price list to the job. An administrator must manually add the products of the price list to the job.

To create a Web content export job detail for Siebel price lists

1. Navigate to the Administration - Pricing screen, then the Price Lists view.
2. Query for the price lists that you want.
3. Select or make a multiple selection of the records of the price lists that you want.
4. Click Update Commerce App.

5. (Optional) Complete the following check boxes in the Update Commerce App dialog box:
   - **Use Effective Date Of.** If you enter a date in this field, then Price List Line Item records are included in the Export Job only if they are valid on this date. For a line item to be included, the date in this field must be between the Start and End Effective Date of both the Price List and Price List Line Item
   - **Include Volume Discounts.** If this check box is selected, then the Siebel application checks the Price List Line Item records of selected price lists to find any associated Volume Discount records and creates an Export Job Detail record for each Volume Discount record.
6. In the dialog box, click Export.
7. Review the message that indicates the number of records that have been sent for export.

Submitting an Export Contents Job

After creating Web content export job details for products, catalogs, and price lists, you export them to a format that can be used by Oracle ATG Web Commerce by submitting the export contents job that has these export job details as its detail records.

The details of the export contents job are submitted to SCOA through the Web service SCOAExportWS. For more information about this Web service, see *Siebel CRM Web Services Reference* .

To submit an export contents job

1. Navigate to the Administration - Order Management screen, then the Web Commerce Export Contents view.
2. Select one unsubmitted Export Contents - Header record.
3. (Optional) Check the Export Contents - Detail records to make sure that you have added the Web content export job details that you want with the options that you want.

   The Detail fields, which are filled in by the Siebel application when you create Web content export job details, are described in the following table.

Field	Comments
Type	The values are: Catalog, Category, Category Product, Class, Price List, Price List Line Item, Product, Promotion, and Volume Discount. These values represent the entity that was exported in the Web content export job.
Action Code	The values are: - **User Requested.** This value is assigned to records explicitly selected by the user. - **Reference.** This value is assigned to any parent record if the user requested records that the Siebel application has added because the records are required in Oracle ATG Web Commerce. For example, if the user selects a category, then the Siebel application automatically adds the parent catalog. - **Delete.** This value is assigned to any record if the user deletes the corresponding entity record after creating the Web content export job.
Object Name	Displays the name of the entity. For example, if you are exporting a product, then this field displays the product name.

Field	Comments
Parent Name	Displays the name of the entity's parent record. For example, if you are exporting a category record, then this field displays the name of the catalog that contains this category. If there is no parent record, then this field is blank.
Use Effective Date Of	If you are exporting a price list, and if you selected the Use Effective Date option, then this field displays the date that you specified. For more information, see *Creating a Web Content Export Job Detail for Siebel Price Lists*.
Full Structure	If you are exporting a product, and if you selected the Full Structure option, then this check box is selected. For more information, see *Creating a Web Content Export Job Detail for Siebel Products or Promotions*.
Include Related Products	If you are exporting a product or promotion, and if you selected the Related Products option, then this check box is selected. For more information, see *Creating a Web Content Export Job Detail for Siebel Products or Promotions*.
Include Volume Discounts	If you are exporting a price list, and if you selected the Include Volume Discounts option, then this check box is selected. For more information, see *Creating a Web Content Export Job Detail for Siebel Price Lists*.
Include Collateral	If you are exporting a product, and if you selected the Include Collateral option, then this check box is selected. For more information, see *Creating a Web Content Export Job Detail for Siebel Products or Promotions*.

4. In the Export Contents - Header record, select a value in the Error Handling Instruction field, described in the following table.

Field	Comments
Error Handling Instruction	Select one of the following values: o **Abort On Error.** ATG Web Commerce aborts the job on the first error encountered and restores any changes that might have occurred. o **Complete With Errors.** ATG Web Commerce completes the job even if there are nonfatal errors. o **Rollback With Errors.** ATG Web Commerce aborts the job on the first error encountered and restores any changes that occurred, but it continues to process all records to find any additional errors that might exist.

5. Click Submit Job.

6. While the job is running, click Refresh periodically to refresh the screen, so you can view the progress of the export job by looking at the two status fields of the Export Contents - Header record, described in the following table.

Field	Comments
Status	This field has the following values: o **Open.** The Siebel product administrator is adding job line items using the Update Commerce functionality. o **Submitted.** The Siebel product administrator has clicked Submit to submit this job. o **In Progress - SCOA.** SCOA has received the new job. o **Error Processing.** SCOA has found errors while processing the job. o **In Progress - ATG.** SCOA has sent a request to ATG to retrieve the job. o **Error Extracting.** ATG has found errors while extracting the job from the SCOA XML files. o **Error Mapping.** ATG has found errors while mapping the job from the SCOA XML files. o **Error Importing.** ATG has found errors while importing the Job from the SCOA XML files. o **Completed.** ATG has completed the job.

Correcting Errors in an Export Contents Job

If an export job has errors, then you can revise the export job to correct the error. The fact that there are errors is indicated by the header's Status field, which has one of the following values: Error Extracting, Error Importing, or Error Mapping.

To correct errors in an export contents job

1. Navigate to the Administration - Order Management screen, then the Web Commerce Export Contents view.
2. Select the job that you want in the Export Contents - Header List.
3. Click Revise.

   > **Note:** Revise is only available if one of the following values is in the Status field: Error Mapping, Error Extracting, Error Importing.

   The Siebel application performs a cascade copy of the contents of this job into the current export contents job with the status of open, or into a new export contents job if there is no current one with the status of open. It copies only the Export Content Details that have a status not equal to Complete.
4. Select this new job header record.
5. Click the Error Messages tab to view the error messages generated by the previous job.
6. Return to the originating data and correct the problems causing the errors.

   For example, the Oracle's ATG Web Commerce might have generated an error when trying to associate a category product to a product because the Product record was never exported. In this case, you must find the Product record and add it to this job.
7. Click Submit Job to resubmit the new job.

Because this new job contains only the records that have not been processed successfully during the previous job, processing time is minimized.

Maintaining Product Data for Integration with Oracle ATG Web Commerce

This topic describes the methods that you must use to work with product-related data in the Siebel application to have the results reflected properly in Oracle ATG Web Commerce. These methods give the Siebel application and ATG Web Commerce access to identical versions of the product and their supporting objects in both applications, which is necessary to avoid incorrect cross-channel orders.

In most cases, you must use End Effective Date to make the object unavailable. However, you can delete nonversioned product data, such as collateral, literature, or recommended products.

Removing a Product from a Catalog and Category

To remove a product from a catalog and category in both the Siebel application and ATG Web Commerce, set the effective end date in the Category view for the product, and then create a Web content export job to export the change to ATG Web Commerce.

To remove a product from a catalog and category

1. Navigate to the Administration - Product screen, then the Products view.
2. Select the product that you want.
3. Click the Category tab.
4. In the categories list, for all categories that you want, enter a value in the Effective End Date field that is before the current date.
5. To verify your work:
    a. Navigate to the Administration - Catalog
    b. In the Catalogs list, click the Name field of the catalog that contains the category that you removed the product from.
    c. In the Categories list, select the category that you removed the product from.
    d. Click the Products tab, and confirm that the Product is no longer in this category.
6. Click Update Commerce App.

Removing a Catalog

To remove a catalog, including its categories and its categories' products, from both the Siebel application and ATG Web Commerce, set the effective end date of the catalog, and then create a Web content export job to export the change to ATG Web Commerce.

Physically deleting a Catalog record, Category record or Category Product record is also supported, but it is not recommended because it becomes impossible to trace which objects have been removed.

To remove a catalog

1. Navigate to the Administration - Catalog screen, then the Catalog view.
2. In the record for the catalog that you want, enter a value in the Effective End Date field that is before the current date.

   This field is in the Administration - Catalog applets:

   - Catalog Admin Catalog List Applet
   - Catalog Admin Catalog Detail Applet
3. Click Update Commerce App.

Removing a Price List

To remove a price list, including its line items, from both the Siebel application and ATG Web Commerce, set the effective end date of the price list, and then create a Web content export job to export the change to ATG Web Commerce.

Physically deleting a price list and price list line item record is also supported, but it is not recommended because it becomes impossible to trace which objects have been removed.

To remove a price list

1. Navigate to the Administration - Pricing screen, then the Price Lists view.
2. In the record for the price list that you want, enter a value in the Effective To field that is before the current date.

Deleting Product Collateral

Product records cannot be deleted. However, collateral, literature and recommended products that are attached to a product can be deleted.

To make collateral, literature and recommended products unavailable in Oracle ATG Web Commerce, you can delete the record in the Siebel application instead of using End Effective Date. If you delete this data, and the system preference for SiebelCommerceInteg is set to True, then a record of type Delete is inserted in the Export Job Details applet, and SCOA sends this record to Oracle ATG Web Commerce.

10 Integrating with Oracle Intelligent Advisor

Integrating with Oracle Intelligent Advisor

This chapter describes how to integrate Siebel Order Management with Oracle Intelligent Advisor to give product selection advice. It includes the following topics:

- *About Creating Product Advisors*
- *Process of Setting Up Oracle Intelligent Advisor to Create Product Advisors*
- *Using Oracle Intelligent Advisor Product Advisors*

About Creating Product Advisors

You can integrate Oracle Intelligent Advisor with Siebel Order Management to create a rule base that guides customers through purchasing decisions by:

- Recommending products, services, or courses of action based on user needs.
- Selecting and configuring products based on product features and user requirements.

Process of Setting Up Oracle Intelligent Advisor to Create Product Advisors

To set up Oracle Intelligent Advisor to create product advisors, the administrator performs the following steps:

1. *Integrating the Siebel Application with Oracle Intelligent Advisor*
2. *Creating the Oracle Intelligent Advisor Rule Base*
3. *Verifying the Integration*

Integrating the Siebel Application with Oracle Intelligent Advisor

You must integrate the Siebel application with Oracle Intelligent Advisor by specifying a symbolic URL.

To integrate the Siebel Application with Oracle Policy Administration

1. In the Siebel application, navigate to Administration - Integration, then WI Symbolic URL List.
2. In the Name field, query for AdvisorsRuleSet
3. If it is not present create a new record with that name.

4. Complete the necessary fields, as shown in the following table.

Field	Value
URL	http://<Server>:<Port>/web-determinations/ with OIA server and Port details For example: http://slc02vaa.us.oracle.com:7065/web-determinations
Fixup Name	InsideApplet
SSO Disposition	IFrame

5. In the Symbolic URL Argument form (the child applet), complete the necessary fields, as shown in the following table.

Field	Value
Name	IFrame
Required Argument	TRUE
Argument Type	Command
Argument Value	iFrame width=750 height=500 frameBorder=On **Note:** Adjust the Adjust the width and height to suit your requirements for displaying the Advisor popups.
Append Argument	TRUE
Sequence	1

6. Log out of the Siebel application, and then log in again.

Creating the Oracle Policy Automation Rule Base

You must create Oracle Intelligent Advisor rule bases that are needed to help with purchasing decisions. The rules that are needed depend on your business model and the products you sell.

For more information creating rule bases, see the Oracle Intelligent Advisor documentation library on Oracle Technology Network.

Verifying the Integration

After you have set up the integration and the rule bases, you can verify the integration.

To verify the integration

1. Navigate to the Sales Order or Quotes Screen, then to the Catalog View and the Get Advice view.
2. Click the Advisors button.

   > **Note:** Right-click this button if you want to open the OIA screen in a different browser window or tab, so you can continue to use the Siebel application without having to close the Advisor.

   What appears depends on how OIA was configured on the server and on what location on the server the URL points at. If you defined multiple of rule bases, it is best to display a list of Advisors, so the user can choose one of the rule bases you created. If you defined only one rule base, it is best to display that rule base directly, without requiring the user to select it from a list.
3. If a list is displayed, click one of the Advisors in the list.

   An Advisor is displayed in the same popup window that the list was displayed in.

Using Oracle Policy Automation Product Advisors

After administrators have set up Oracle Intelligent Advisor to create product advisors, users can access the rule bases to guide purchasing decisions.

The procedure for the user to access the rule base is the same as the procedure that the administrator uses to verify the integration. For more information, see *Verifying the Integration*.

11 Distributed Order Orchestration

Distributed Order Orchestration

This chapter describes distributed order orchestration. It includes the following topics:

- *About Oracle Fusion Distributed Order Orchestration*
- *Scenarios for Distributed Order Orchestration*
- *Tasks for Using Distributed Order Orchestration*

About Oracle Fusion Distributed Order Orchestration

Oracle Fusion Distributed Order Orchestration has the following features:

- It works to normalize processes and data across multiple-order capture and order-fulfillment applications, providing a single view of all order and fulfillment information.
- It aggregates orders from a variety of order capture applications and groups them to create a combined fulfillment plan that makes the best use of company resources.
- It includes native exception management and embedded business analytics.

You can integrate Siebel Customer Order Management with Oracle Fusion Distributed Order Orchestration, allowing you to view the fulfillment information from other applications directly within the Siebel application. This integration enables you use the Siebel application to:

- Track the fulfillment status of orders.
- Revise or cancel orders only if they have not yet been shipped.
- Track the status of return merchandise authorization orders.
- Check the availability of products that customers order.

Note: Before you can use distributed order orchestration, you must integrate distributed order orchestration. For more information, see *Integrating Distributed Order Orchestration*.

Scenarios for Distributed Order Orchestration

This topic gives the following example of how distributed order orchestration might be used. You might use distributed order orchestration differently, depending on your business model. This topic includes the following scenarios:

- *Submitting and Tracking an Order*
- *Revising an Order*
- *Canceling an Order*
- *Submitting a Return Merchandise Authorization Order*

- *Checking Availability to Promise of a Product*

Submitting and Tracking an Order

A new customer telephones a call center to place an order. The call center agent enters information to create a new customer account. Then the agent enters the order and the payment information.

Two weeks later, the customer calls to ask about the status of the order. The call center agent sees that the order status is *Shipped* and says that the order was shipped two days ago. The agent tells the customer which fulfillment center the order was shipped from and what the tracking number for the shipment is.

Another new customer places an order and calls two weeks later to ask about the status of the order. The call center agent sees that the order status is Partial Shipped. The call center agent navigates to the Orchestration view for the order, which shows that half of the items were shipped two days ago and half were shipped one day ago. The call center agent gives the fulfillment center and tracking number for both shipments.

Because ordering and fulfillment are integrated, the agent can track fulfillment, using the same Siebel application that is used to place the order. If the order is fulfilled with several shipments, then distributed order orchestration creates an orchestration record in the Siebel application for each of the shipments.

Revising an Order

A customer places an order through a call center. The next day, the customer telephones the call center to revise the order, changing the quantity from five items to four items. The call center agent checks the status and sees that the order is still being processed. The agent changes the quantity from five to four.

Another customer places an order and telephones the next day to revise the order, changing the quantity from five items to four items. The call center agent checks the status of the order and sees that it has been shipped. Because it has been shipped, the order record is read-only, and the call center agent cannot change the quantity. The call center agent tells the customer that he cannot change the order because it has already been shipped. Because ordering and fulfillment are integrated, the agent cannot change orders that have already been shipped.

Canceling an Order

A customer places an order through a call center. The next day, the customer telephones the call center to cancel the order. The call center agent checks the status and sees that the order is still being processed. The agent cancels the order.

Another customer places an order and telephones the next day to cancel the order. The call center agent checks the status of the order and sees that it has been shipped. Because it has been shipped, the order record is read-only, and the call center agent cannot cancel it. Because ordering and fulfillment are integrated, the agent cannot cancel orders that have already been shipped.

Submitting a Return Merchandise Authorization Order

A customer telephones a call center to return recently purchased merchandise. The call center agent creates a return merchandise authorization (RMA) order and gives the customer the address where the merchandise must be shipped.

The customer telephones again five days later and asks for the status of the order. The call center agent says the merchandise has been received in new condition. Because ordering and fulfillment are integrated, the agent knows that the fulfillment center has received the returned merchandise.

Note: Users cannot revise RMA orders. Distributed order orchestration does not support exchanging returned merchandise for different merchandise.

Checking Availability to Promise of a Product

A customer telephones a call center to buy a product and asks if the product is in stock. The call center agent navigates to the Fulfillment view, selects the product, and clicks Inquire. The Siebel application displays a message saying how many units of the product are in stock and which fulfillment centers stock them.

The customer places an order for the product. Because ordering and fulfillment are integrated, the agent can check inventory to see which products are in stock.

Note: Reserving products and scheduled lines are not supported by DOO.

Tasks for Using Distributed Order Orchestration

Customer service agents or call center agents using distributed order orchestration can perform the following tasks:

- *Tracking Orders with Distributed Order Orchestration*
- *Revising Orders with Distributed Order Orchestration*
- *Cancelling Orders with Distributed Order Orchestration*
- *Using Return Merchandise Authorization Orders with Distributed Order Orchestration*
- *Checking Availability to Promise with Distributed Order Orchestration*

Note: Before agents can perform these tasks, you must integrate distributed order orchestration. For more information, see *Integrating Distributed Order Orchestration*.

Tracking Orders with Distributed Order Orchestration

If you have set up distributed order orchestration, then orders and fulfillment are integrated, and you can track fulfillment with the same Siebel application that you use for taking orders.

To track orders with distributed order orchestration

1. Navigate to the Sales Order screen, then the List view.
2. Click the Order # field of the order that you want to track.
3. Click the Shipping view tab.
4. Click the Orchestration Items view tab.

A list of all products in the order appears, with the product name, order item ID, quantity, scheduled ship date, expected delivery date, source inventory location, and status of each.

Revising Orders with Distributed Order Orchestration

If you have set up distributed order orchestration, then the Siebel application enables you to revise active booked order before its status becomes Shipped. The rules are defined using data validation manager.

To revise an order with distributed order orchestration

1. Navigate to the Sales Order screen, then the List view.
2. Select the record for the order that you want to revise.
3. In the form for the order, click Revise.

   If the order has not been shipped, then you can click the Revise button. However, if the order has been shipped, then the Revise button is disabled.

Cancelling Orders with Distributed Order Orchestration

If you have set up distributed order orchestration, then the Siebel application enables you to cancel an order before its status becomes Shipped. The rules for disabling this cancellation functionality are defined using data validation manager.

To cancel an order with distributed order orchestration

1. Navigate to the Sales Order screen, then the List view.
2. Select the record for the order that you want to cancel.
3. Revise the Sales Order if possible.
4. Expand Sales Order Form, and provide the Cancel Reason.
5. Change the status to Booked.

   The Cancel Order button is enabled.
6. Click Cancel Order.

Using Return Merchandise Authorization Orders with Distributed Order Orchestration

If a customer wants to return merchandise for a refund, then the sales agent can submit an RMA order. When the merchandise is received, the warehouse records that it received the merchandise in an application that is integrated with Siebel Order Management. Because the applications are integrated, distributed order orchestration updates the status field of the RMA order in the Siebel application.

If the customer asks for the status of the RMA order, then the sales agent can check the Status field in the Siebel application.

Creating RMA Return Orders

To create an RMA order, the sales agent creates a new order and selects RMA Return at the order type.

The RMA return order is a new order that is not linked with the original order. RMA orders cannot be revised. RMA orders do not specify whether they are returned for credit or for cash. Distributed order orchestration supports RMA return orders.

To create RMA return orders

1. Navigate to the Orders screen, then the List view.
2. In the Service Orders list, add a new record, and complete the necessary fields.

   For more information about creating service orders, see *Siebel Field Service Guide* .
3. In the Type field, select RMA Return.
4. From the menu, select Submit.

Checking the Status of RMA Return Orders

Distributed order orchestration updates the status field of the RMA order asynchronously after the product is received and the receipt is entered in the application that is integrated with the Siebel application.

To check the status of RMA Return Orders

1. Navigate to the Orders screen.
2. Find the order you want, and look at the value in the Status field.

Checking Availability to Promise with Distributed Order Orchestration

Availability to promise (ATP) enables you to check whether items that a customer requests are available in inventory.

Before you can use ATP, you must use distributed order orchestration to integrate the Siebel application with the inventory management application. For more information, see *Integrating Distributed Order Orchestration*.

To check availability to promise

1. Navigate to the Quote screen, then the List view, or navigate to the Sales Order screen, then the List view.
2. Click the Quote Number or Order Number link of the Quote or Order record that you want.
3. Click the Shipping view tab and then the Fulfillment view tab.

   The Fulfillment view contains a list of products with the quantity available for each.
4. Check availability to promise:

   a. Select the record for the product that you want.
   b. In the Quantity field, enter the quantity of the product that you want.
   c. Click Inquire.

      A message appears telling you whether the product is available to promise.

12 Creating and Managing Catalogs

Creating and Managing Catalogs

This chapter describes how to create and administer catalogs. It includes the following topics:

- *About Siebel Catalog*
- *Roadmap for Creating Catalogs*
- *Process of Setting Up Catalogs*
- *Process of Setting Up Categories*
- *Building Catalog Hierarchies*
- *Example Process of Creating Display Templates for Categories*
- *Process of Setting Up Attribute Search*
- *About Dynamic Product Catalogs*
- *About the Siebel Sales Catalog in Open UI*

About Siebel Catalog

Siebel catalog provides core functions that are used by a number of applications to organize master data, for both administrative and display purposes. The data types used by some applications are shown in the following table.

Application	Master Data Type
Siebel eSales	• Products • Literature
Siebel Sales Catalog	• Products • Literature
Siebel Self-Service	• Solutions • Smartscripts • Literature • Resolution Items

In Siebel eSales, catalogs allow customers to browse for products and literature of interest. If you also want your customers to be able to browse for training courses or event information, then your company must also have licenses for Siebel Training or Siebel Events, respectively.

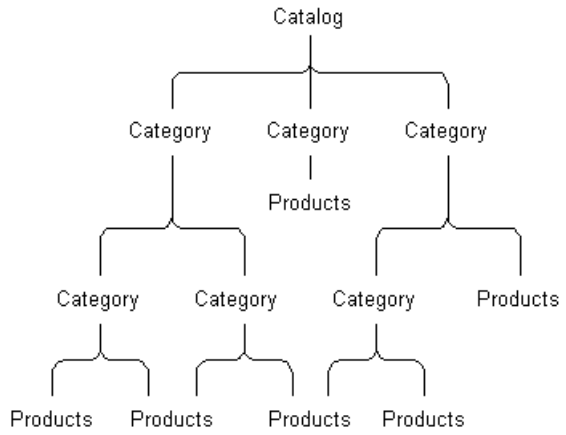
Siebel Sales Catalog, which allows employees to browse for products and literature, is available for use with the Siebel Quotes or Siebel Order modules of the Siebel Partner Relationship Management, Siebel Sales, and Siebel Call Center applications.

> **Note:** If you are using Siebel Open UI, the Siebel Sales Catalog provides three new motifs for viewing Catalog Master Data. These new motifs include Tile, Detail, or List displays, and they provide viewing features that do not respond to many of the configuration instructions in this chapter. This chapter includes notes indicating which features do not apply to Open UI Sales Catalog.

Roadmap for Creating Catalogs

A catalog is a collection of items arranged in a logical hierarchy of categories. A schematic showing the relationships between a catalog, categories, and products is shown in the following figure.

- Categories can contain content items and other categories (called subcategories).
- Each category can belong to only one catalog and one parent category.
- Products can be associated with multiple categories.



Catalogs are created and managed using the following steps:

- **Setting up the catalog.** For more information, see *Process of Setting Up Catalogs*
- **Setting up its categories.** For more information, see *Process of Setting Up Categories*
- **Building the catalog hierarchy.** For more information, see *Building Catalog Hierarchies*

Process of Setting Up Catalogs

To set up catalogs, perform the following tasks:

- *Creating Catalogs*
- *Assigning Access to Catalogs*

- *Selecting Languages for Catalog Translation*

This process is a step in *Roadmap for Creating Catalogs*.

Creating Catalogs

You create catalogs on the Catalogs page in the Administration - Catalog screen.

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

To create a catalog

1. Navigate to the Administration - Catalog screen.
2. Add a new record to the Catalogs list and complete the necessary fields in the record and the More Info form. Some fields shown in the following table are entered using the form under the More Info tab.

Field	Description
Name	Enter a name for the catalog.
Effective Start Date	Enter the date when the catalog becomes visible to end users.
Effective End Date	Enter the date when the catalog stops being visible to end users.
Private	Select this checkbox to make the catalog visible only to users who belong to the access groups associated with it. If the catalog is private, then all of its categories will also be private.
Active	Select this checkbox to make the catalog visible to end users.
Catalog Type	Choose Buying from the list if you want the catalog to be displayed on the Web site and in the employee application. Catalog Type can be used to control what is displayed by setting search specifications on applets. For example, you could create two catalog types named Customer and Partner. You could then create a view with an applet that has a search specification to display only catalogs of type Partner and a second view with an applet that has a search specification to display only catalogs of type Customer.
Sequence	Enter a number to control how this catalog is displayed. Catalogs are displayed in ascending order by their Sequence field.

Field	Description
	Note: It is strongly recommended that you make an entry in this field. Records where this field is empty appear either before or after records where this field is populated, depending on the database platform, but the records with the Sequence field empty are randomly sequenced.
Image	Select an Image file associated with the catalog.
Version	Enter a version number for the catalog. This field allows you to copy a catalog and increment the version number. The catalog is identified by a unique combination of name and version number.
Smart Answer Catalog File, Smart Answer Master Server	To add Smart Answer, specify the catalog file and master server. For more information, see *Siebel Smart Answer Guide* .

Assigning Access to Catalogs

Access groups control which groups of users can see particular catalogs. Access groups are associated with catalogs under the Access Groups tab in the Administration - Catalog screen.

Note: Before you can perform this step, you must create the access group. For information on creating access groups, see *Siebel Security Guide* .

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

To associate an access group with a catalog

1. Navigate to the Administration - Catalog screen.
2. In the Catalogs list, click the name of a catalog.
3. Click the Access Groups view tab.

   The access groups associated with the catalog appear.
4. Add a new record to the Access Groups list.

   A dialog box appears.
5. Select an access group, and then click OK.

   The access group appears under the Access Groups tab.

Selecting Languages for Catalog Translation

You can set up your catalogs in multiple languages. This setup is done under the Catalog Translations tab in the Administration - Catalog screen.

For more information on creating multilingual applications, see *Siebel Global Deployment Guide* .

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

To select a language for catalog translation

1. Navigate to the Administration - Catalog screen.
2. In the Catalogs list, click the name of a catalog.
3. Click the Catalog Translations view tab.
4. Add a new record to the Catalog Translations list.
5. Click the Code field select button.
   The Language Name dialog box appears.
6. Select a language, and click OK.
   The language name and code appear in the Catalog Translations list.
7. In the Description field, type a description of the catalog in the new language.

Process of Setting Up Categories

Categories group content items, such as products and literature.

To set up categories, perform the following tasks:

- *Creating Categories*
- *Adding Products to Catalogs*
- *Adding Literature to Catalogs*
- *Updating the Category Item Count*, and if necessary, *Removing the Category Item Count*
- *Assigning Access to Categories*
- *Selecting Languages for Category Translation*

This process is a step in *Roadmap for Creating Catalogs*.

Creating Categories

Categories are created in the Administration - Catalog screen under Categories. They are displayed to the end user in ascending order by their Sequence field.

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

To create a category

1. Navigate to the Administration - Catalog screen.

2. In the Catalogs list, click the Name link for the catalog to which you want to add categories.

   The Categories list appears with the categories belonging to that catalog.
3. Add a new record to the Categories list, and complete the necessary fields. Some fields are described in the following table.

Field	Description
Name	Enter a name for the category
Display Name	Enter the name that the end user sees.
Sequence	Enter a number that controls the sequence in which categories are displayed. Categories are displayed in ascending order starting with 0.
Image	Select an image file associated with the category.
Active	Select this check box to make the category available.
Private	Select this check box if you do not want the category to inherit access properties from its parent category. If the parent category has been marked private, then this category also must be private.
Effective Start Date	Enter the date when the category becomes available for use.
Effective End Date	Enter the date when the category stops being available.
Usage	Select the type of category.
Display Template	If you are using the high-interactivity user interface, you can select a template for displaying the category to the end user. **Note:** This field does not apply if you are using Siebel Open UI. The following display templates are available by default: - **Add to Cart.** Use this when you want customers to move items from a list to the Shopping Cart one at a time. - **Category Only.** Use this template for categories that have subcategories but no products. - **Multiselect Checkbox.** Use this template when you want the customer to select items from a list and move all selected items to the Shopping Cart at one time. - **Product Basic.** Use this template to include the name of the product, a description, and quantity.

Field	Description
	○ **Product Rich.** Use this basic template to include just the name of the product and a description. For information on creating new display templates, see *Example Process of Creating Display Templates for Categories*.

Adding Products to Catalogs

You add products to catalogs by associating them with categories that belong to those catalogs.

Note: Before you can perform this step, you must define products. For information on defining products, see *Siebel Product Administration Guide* .

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

To associate a product with a category

1. Navigate to the Administration - Catalog screen, then the Catalog Administration view.
2. Click the Name link for a catalog.

   The Categories list displays the categories belonging to that catalog.
3. Select a category with which to associate products.
4. In the link bar for the category, click Products.

   The products associated with the category appear.
5. Add a new record to the Products list.

   The Add Internal Products dialog box appears.
6. Select products, and then click OK.

   The products are associated with the category and appear under the Products tab.

Adding Literature to Catalogs

You can make literature available to employees or customers by creating catalog categories and associating documents with them. Customers can click the Literature Catalog link in the eSales Web site to access the catalog and the documents in it. You can also create new applets for your Web site to display literature.

Note: This topic does not apply if you are using Siebel Open UI sales catalogs, which do not display literature.

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

To associate a document with a category

1. Navigate to the Administration - Catalog Screen.

2. In the Catalogs list, query to find the Self-Service Catalog catalog.
3. Click the Self-Service Catalog link in the Name field.
4. Select a category or add a new one.
5. In the link bar, click Literature.
6. In the Literature list, add a new record.
   The Add Literature dialog box appears.
7. In this dialog box, select the literature that you want, and click OK.

For more information about working with literature, see Siebel eService Administration Guide.

Updating the Category Item Count

The category item count displayed at run time reflects the total number of items in each category. If you are using the high-interactivity user interface, you must update this count after adding new items to categories.

Note: This task does not apply if you are using Siebel Open UI.

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

To update the category item count

1. Navigate to the Administration - Catalog screen.
2. In the Catalogs list, click the Name link for a catalog.
   The Categories list displays the categories belonging to that catalog.
3. Select a category.
4. From the Categories applet menu, select Update Count.
   The category item count is updated for that category.

Removing the Category Item Count

The category item count reflects the total number of items in the category. Certain users might not have access to all of the items in a category. If so, the count would not be accurate for those users.

If this item does not meet your needs, then you can use Web Tools to remove the display of the count from the following applets by making the count inactive:

- Product Catalog RootSubCategory List Applet (eSales)
- Product Catalog SubCategory List Applet (eSales)

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

To remove the category item count

1. In Web Tools, create or open a Workspace.
2. Select the Applet object.
3. Query for the appropriate applet.
4. Select Base, expand the Applet Web Template object, and then select Applet Web Template Item.

The Applet Web Template Items list appears.

5. In the Count row, select the Inactive check box.

   The color of the Count record text changes to red.
6. Save the Workspace and submit the changes for delivery.

Assigning Access to Categories

Access groups control which groups of users can see particular categories. Access groups are associated with categories under the Access Group tab.

If you have already associated an access group with a catalog, as described in *Assigning Access to Catalogs*, then that group of users also has access to all the categories in the catalog, and you do not have to assign it access to the categories. Assign access to categories only if you want access to individual categories to users who do not have access to the entire catalog.

> **Note:** Before you can perform this step, you must create the access group. For information on creating access groups, see *Siebel Security Guide* .

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

To associate an access group with a category

1. Navigate to the Administration - Catalog screen.
2. In the Catalogs list, click the name of a catalog.
3. In the Category Detail applet, select the category you want.
4. Click the Access Groups view tab.

   The access groups associated with that category appear.
5. Add a new record to the Access Groups.

   A dialog box appears.
6. Select an access group, and then click Add.

   The access group appears in the Access Groups list.

Selecting Languages for Category Translation

You can choose the languages for which to translate category information. For more information on creating multilingual Web sites, see *Siebel Global Deployment Guide* .

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

To select a language for category translation

1. Navigate to the Administration - Catalog screen.
2. In the Catalogs list, click the name of a catalog.
3. In the Category Detail applet, select the category you want.

4. Click the Category Translations tab.

   Languages for translation appear.
5. Add a new record to the Category Translations list and complete the necessary fields, described in the following table.

Field	Description
Code	Select a language. After you make this selection, the code for the language is displayed in the Code field, and the name of the language is displayed in the Language field.
Display Name	Name in the new language that the Siebel eSales Web site visitor sees.
Description	Brief description of the category in the new language.
Extended Description	More detailed description of the category in the new language.

Building Catalog Hierarchies

A catalog hierarchy determines the structure and navigation paths in the catalog. Hierarchies are built and modified using the buttons in the Categories list. Preceding the Categories list, there are four buttons that are used to promote, demote, move up, and move down categories.

This task is a step in *Roadmap for Creating Catalogs*.

To create subcategories

1. Navigate to the Administration - Catalog screen.

   The Catalogs list appears.
2. Click the name of a catalog whose hierarchy you wish to modify.

   The Categories list displays the categories belonging to that catalog.
3. Use the move up and move down buttons to place a category you wish to make a subcategory beneath the category you wish to make its parent.
4. With the child category selected, click the demote one level button.

   The subcategory appears in a tree as a node under its parent. The tree can be expanded and collapsed.
5. Arrange other categories under the parent category as necessary.
6. Select each category in turn and click the demote one level button.

   The subcategories are placed in the tree under the other subcategory. The Sequence field can be used to display the subcategories in the correct order on the Categories page and on the Siebel eSales Web site.

7. Use the buttons as necessary to extend the catalog hierarchy.

   Subcategories are displayed under categories on the Siebel eSales Web site in ascending order by sequence, and then in alphabetical order by display name.

Example Process of Creating Display Templates for Categories

Display templates control how catalog categories are displayed to the end user. A number of templates are available by default, as described in *Creating Categories*. You can also create custom display templates.

Note: This topic does not apply if you are using Siebel Open UI sales catalogs.

You create and edit display templates in Siebel Tools, and then add them to the Siebel eSales application using your employee application's administration screens.

The example shown in this process uses a display template named Product New, which uses a view named Product Catalog Category New Detail View (eSales).

For more information on creating Siebel views and applets, and on dynamic drilldown behavior, see *Using Siebel Tools* .

To create a new display template, perform the following tasks:

- *Creating a Category Detail View*
- *Creating Drilldown Objects for the View*
- *Editing the Web Layout of the New Display Template*
- *Adding the New Display Template and View to the Application*

Note: You must also add the view to the appropriate responsibilities, for example Siebel Administrator and Web Registered User.

Creating a Category Detail View

You create the view and add it to the Siebel eSales Catalog screen using Siebel Tools.

This task is a step in *Example Process of Creating Display Templates for Categories*.

To create a detail view for the new display template

1. In Siebel Tools, lock the eCatalog project.
2. Select the View object.
3. Query for Product Catalog Category Detail View (eSales).
4. Copy the record, and then give the copy a new name, for example Product Catalog Category New Detail View (eSales).
5. Select the Screen object.
6. Query for eSales Catalog Screen.

7. Expand the Screen object, and then select Screen View.
8. Add Product Catalog Category New Detail View (eSales) to the Screen Views list.

Adding the View to the Siebel eSales Catalog Screen

In Siebel Tools, use the standard method for adding the view to the Catalog screen. For more information, see *Using Siebel Tools* .

Creating Drilldown Objects for the View

You create drilldown objects for the view using Siebel Tools.

This task is a step in *Example Process of Creating Display Templates for Categories*.

To create drilldown objects for the view

1. In Siebel Tools, select the Applet object.
2. Query for Product Catalog RootSubCategory List Applet (eSales).
3. Expand the Applet object, and then select Drilldown Object.

   The Drilldown Objects list appears.
4. Copy the Product Basic record.
5. In the new record, fill in the fields shown in the following table, leaving the rest the same as in Product Basic.

Field	Value
Name	Product New
View	Product Catalog Category New Detail View (eSales)

6. Expand Drilldown Objects, and then select Dynamic Drilldown Destination.

   The Dynamic Drilldown Destinations list appears.
7. Copy the Product Basic record.
8. In the new record, fill in the fields shown in the following table, leaving the rest the same as in Product Basic.

Field	Value
Name	Product New
Field	Display Template Lang Indep Name
Value	Product New
Sequence	3

Field	Value

9. Repeat Step 6 through Step 8 for the Product Basic and Product Rich drilldown objects.
10. Repeat Step 2 through Step 9 for Product Catalog SubCategory List Applet (eSales).

Editing the Web Layout of the New Display Template

You edit the Web layout of the new display template using Siebel Tools, and then compile the project.

For more information on editing Web layouts, see *Siebel Tools Online Help* .

This task is a step in *Example Process of Creating Display Templates for Categories*.

To edit the Web layout of the new display template

1. In Siebel Tools, select the View object.
2. Query for Product Catalog Category New Detail View (eSales).
3. Right-click on the view record, and then choose Edit Web Layout.
   The Web layout appears.
4. Edit the layout, and then save it.
5. Submit the Workspace for delivery.

Adding the New Display Template and View to the Application

To add the new display template to the application, you must do the following in the Application Administration screen:

- Add the display template to the List of Values.
- Add its view to the Views list.
- Add its view to the appropriate responsibilities in the Responsibilities list.

This task is a step in *Example Process of Creating Display Templates for Categories*.

To add the new display template and view to the application

1. Navigate to the Application Administration screen, and then the List of Values view.
   The List of Values appears.
2. Add a new record.
3. Fill in the fields in the following table.

Field	Value
Type	PROD_CTGY_DISPLAY_TEMPLATE
Display Value	Product New

Field	Value
Language Independent Code	Product New
Language	English-American

4. Choose Views from the Show drop-down menu.

   The Views list appears.
5. Add a new record.
6. Fill in the fields in the following table.

Field	Description
View Name	Product Catalog Category New Detail View (eSales)
Description	Product Catalog Category New Detail View (eSales)
Local Access	Check the check box.

7. Choose Responsibilities from the Show drop-down menu.
8. Add Product Catalog Category New Detail View (eSales) to the appropriate responsibilities, for example Siebel Administrator.
9. Restart the Siebel Server.

Process of Setting Up Attribute Search

Users can choose to search for products using values of the attributes of those products. This search is known as *attribute search*, and it is sometimes also known as *parametric search*.

Attribute search allows users to take advantage of the class system. Instead of entering free-form search strings into a full-text search engine, users of attribute search can select from known attributes (for example, color, size, shape, and maximum temperature) of a particular class of products.

Use of Attribute Search

Sales representatives access attribute search as follows:

- Click Search in the link bar of the Catalog view of a Quote or Order. The Attribute Search form appears.
- Select a product family in the Select a Product Family list. Attributes and other product details specific to the product family appear in the search form, which appears after the product family.

- Selects specific values for one or more of these attributes and product details. There are drop-down lists that the user can use to select attributes. The user can also enter values manually, including comparison operators.
- Click Search. Search results appear on the far side of the form, including all products in the catalog that match these search criteria.

Some examples of possible attribute searches are:

- **Multiple attribute search.** The user enters values for more than one attribute. For example, in searching for a car, the user can select Red as the value for the Exterior Color attribute and Gray as the value for the Interior Color attribute. The search returns all cars in the catalog with red exteriors and gray interiors.
- **Attribute-price search.** The user enters values for an attribute and for the price. For example, in searching for a coat, the user can select black as the value for the Color attribute and enter less than 250 (<250) as the value for price. The search returns all coats in the catalog that are black and cost less than $250.

To set up attribute search, perform the following tasks:

- *Making Classes Searchable*
- *Making Attributes Searchable*
- *Removing the List Price Field from Attribute Search*

Making Classes Searchable

Product families are those classes that have been made searchable.

You must set the Searchable checkbox separately for each class that you want to make searchable. Subordinate classes in the hierarchy do not inherit this property from their parent classes.

See *Siebel Product Administration Guide* for more information on the class system.

This task is a step in *Process of Setting Up Attribute Search*.

To make a class searchable

1. Navigate to the Administration Product screen, and then the Product Classes view.
2. In the Product Classes list, select the Locked Flag checkbox for the class that you want.
3. In the Versions list, in the record for the Workspace version, select the Searchable checkbox.
4. Click Release New Version.

Making Attributes Searchable

You can choose which attributes are searchable.

This task is a step in *Process of Setting Up Attribute Search*.

To make an attribute searchable

1. Navigate to the Administration Product screen, and then the Product Classes view.
2. In the Product Classes list, select the Locked Flag checkbox for the class that the attribute is in.
3. In the Versions list, drill down on the Workspace version.
4. Click the Attributes view tab.
5. In the Attributes list, in the record for the attribute, select the Searchable checkbox.

6. In the Product Classes list, select the class and click Release New Version.

Removing the List Price Field from Attribute Search

You can remove the List Price field from a attribute search.

This task is a step in *Process of Setting Up Attribute Search*.

To remove the List Price field from an attribute search

1. In Siebel Tools, select the Applet in the Object Explorer.
2. Query for the Applet Attribute Search Applet.
3. In the Object Explorer, select Applet User Prop.
4. Select the Show Price Attribute user property.
5. Change the Value from Y to N.

About Dynamic Product Catalogs

The dynamic product catalog is displayed when the user is creating a quote or an order in an application that uses the high-interactivity interface.

Note: This topic does not apply if you are using Siebel Open UI, which does not support dynamic product catalogs.

This screen has enhancements that are useful for employees placing orders.

- In addition to the usual header and line items area, it has a product selection work area, which consolidates all product selection mechanisms and makes them accessible from one location. This work area displays all product selection mechanisms using the high-interactivity interface, improving performance.
- It has a Show Navigator button that the user can click to view a menu of the structure of the catalog, which allows the user to jump immediately to a product.
- It gives users access to the persistent Quote or Order header and selected product information.
- It allows users to select attributes in the catalog, while browsing or searching for products, before adding the products to a quote or order.
- It allows users to edit the header and line items, and only the data is refreshed, not the view.
- It performs only minimum screen refresh.
- It can show multiple price information.
- It is integrated with eligibility and compatibility, which are supported in all product selection mechanisms, including Catalog, Configurator, product pick-applets, Add Item and Search.
- It is integrated with promotions.
- It includes an integrated search tool, which allows you to search using S_PROD_INT fields and price. The list of search criteria can be configured using Siebel Tools.
- It allows users to select Favorites, predefined product instances that the user can add to the current line item. Users can choose to add the entire list of Favorites or individual items. Users can edit Favorites before adding them to the line item. Users can add multiple products using the Quick Add add applet.

- Users can view all the assets of the account of the current quote or order. Users can perform the asset-based ordering operations, Modify, Disconnect, Suspend, and Resume.
- For inactive quotes and orders, which the user, it suspends the user's ability to add or edit items.
- It is administered using the standard catalog administration screen and it supports all catalog visibility, including price list based visibility, without configuration.
- It can be configured using standard Siebel tools object changes.

The dynamic catalog is also fully integrated with Configurator:

- When the user clicks Customize, Configurator appears in the catalog in the header and line item applets. Header and Line item applets are read-only in this view
- Configurator UI refresh does not refresh the whole view.
- Child component display does not refresh the whole view.
- Tab change does not refresh the whole page.
- It can be configured using Siebel Tools so it does not display header and line items applets.

For more information about how end users work with the dynamic catalog, see *Adding Line Items to a Quote or an Order*.

About the Siebel Sales Catalog in Open UI

Siebel Sales Catalog in Open UI uses the features that Open UI makes available in Siebel CRM. The most important are the Tree Explorer and Tile Renderer, which are available in quotes, sales orders and service orders.

About Navigating the Catalog in Siebel Open UI

The user navigates through the catalog using separate applets, each having its own renderer and modeler.

The Quote Catalog List Applet OUI Drop-Down Applet

The renderer for this applet is responsible for displaying the accessible catalogs to the user in a drop-down control. To customize the applet, use the manifest administration functionality to reference a new renderer or a new model for the drop-down. For more information, see *Configuring Siebel Open UI* .

When the user works with the Catalog drop-down list, this list displays the catalogs that are available to the user each time the view is refreshed. The catalogs that are available depend on the access control features described in *Assigning Access to Catalogs*.

When the user selects a catalog, the child applet supported by the tree explorer control displays the appropriate categories for the user.

To change this behavior of the tree applet, you must modify the renderer or model file. To customize the applet, use the manifest administration functionality to reference a new renderer or a new model for the tree explorer. For more information, see *Configuring Siebel Open UI* .

The Category Explorer Applet

The Category Explorer applet provides a collapsed view of all visible categories when the catalog is selected.

If the user selects the *expand* icon of the category without placing the cursor on the category, the control opens the category and navigates the cursor to highlight the first available subcategory. The Category Products applet displays the products of the highlighted category.

When the user selects a category without clicking the *expand* icon, the category remains in its current state and the available products of the Category Products applet are displayed.

When the user navigates to a different root category within the same catalog, the previously expanded root category is collapsed.

About Adding Products to the Shopping Cart in Siebel Open UI Catalog Browse View

The user can add products using several applets.

The Quote Catalog Product List Applet - Add to Cart Tree Applet

This applet provides three distinct user experiences, so the user can choose whichever they prefer to add products to the Shopping Cart from the Catalog.

The renderer for this applet is responsible for displaying the accessible products using one of the three modes by default. To customize the applet, use the manifest administration functionality to reference a new renderer or a new model for the drop-down. For more information, see *Configuring Siebel Open UI*

To switch between Compact, List, and Detail displays, click the following buttons:

- **Compact Display.** Displays the products as Compact Tiles.

  If no eligibility or compatibility rule exists for a product, the tile includes only the pricing details for the product. If an eligibility or compatibility rule exists for a product, the tile includes two separate sections, one describing the pricing details and the other describing the rule violation.

  From the pricing details section, the user can:

  - change the quantity.
  - click the edit promotion or customize product icon to configure the promotion or product and save it to the Cart.
  - click the Shopping Cart icon to add the product to the Cart.

- **List Display.** Displays the products in a grid layout. The user can:

  - Click the Customize and Edit Promotion buttons to modify products and promotions. These buttons become active or inactive as the user select products.
  - Click the Add Item button used to add the selected button to the Shopping Cart.

- **Detail Display.** Display product details in an expanded tile, which is similar to the compact tiles but has more room for descriptive text.

  Use the arrow buttons to navigate through products.

About Viewing the Shopping Cart in Siebel Open UI Catalog Browse View

The user clicks the Shopping Cart icon to display the Shopping Cart in the Browse Catalog view. By default, the Cart is collapsed and only expanded by user notification.

The Quote Item List Applet - Short OUI

The renderer for this applet is responsible for displaying the products selected and configured within the existing quote. To customize the applet, use the manifest administration functionality to reference a new renderer or a new model for the drop-down. For more information, see *Configuring Siebel Open UI* .

Using the Shopping Cart

When the user navigates to the Browse Catalog view, the Shopping Cart is collapsed by default. To access the Cart, the user clicks the Shopping Cart icon, and the Shopping Cart drop-down appears with the products that have been added.

The user can click the Garbage Can icon in the Shopping Cart to remove a product that has been added previously.

The Messages list, which appears after the Shopping Cart, includes any upsell or cross-sell messages that are displayed as a result of the user's previous selections.

The user can collapse the Shopping Cart, by clicking the Shopping Cart icon again.

13 Setting Up Product Recommendations

Setting Up Product Recommendations

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

- *About Product Recommendations*
- *Scenario for Product Recommendations*
- *Creating Product Recommendations*
- *Turning Off Product Recommendations*
- *Guidelines for Implementing Product Recommendations*
- *Product Recommendation Technical Reference*
- *Configuring Recommendation and Scoring Logic*
- *Unified Messaging Service and Product Recommendations*

> **Note:** Instead of using the methods described in this chapter, you can create recommendation rules in tabular form using either adjustment groups or dynamic matrices. This method is recommended when you are creating recommendation rules for many different products at the same time. For more information, see the topics about adjustment groups and dynamic matrices in *Siebel Pricing Administration Guide* .

About Product Recommendations

Product recommendations allow a marketing administrator to define up-sell and cross-sell recommendations for a product. The marketing administrator enters the text of the recommendation message and a score indicating how important this recommendation is compared with other recommendations for the product.

When a sales representative uses the product catalog to add a product to a quote or order, the Messages list displays the recommendations for this product in order of their score.

The sales representative can click the Accept button in the Messages list to accept a recommendation, automatically adding the recommended product to the quote or order. The sales representative can click the Reject button to reject a recommendation, removing the recommended product from the Messages list.

If an upsell recommendation is rejected, then it will not be shown again to the same account or contact. If a cross-sell recommendation is rejected, then it will not be shown again in the same quote or order, but it is shown again if the same account or contact creates a new quote or order for the product. The administrator can change this behavior by editing the message type responses definition in the Unified Messaging Service.

For more information about editing the message type responses definition to change rejected recommendation processing, see *Customizing Duplicate Message Suppression for Product Recommendations*. For more information about working with the Unified Messaging Service, see *Siebel Order Management Infrastructure Guide* .

The way in which the message is displayed depends on how the message type was defined in the Administration - Order Management, Message Types view. For more information about defining messages, see *Siebel Order Management Infrastructure Guide* . The recommendations list can be displayed in two ways:

- If the display mode for the message was set to passive when the message type was defined, the messages are displayed in the UMF Messages applet in Catalog view and UMF popup applet (triggered using the red flag icon) in Line Items view.
- If the display mode for the message was set to active when the message type was defined, the message(s) are displayed one after another in a modal dialog box in the descending order of the score. User must select a response to the message (such as Accept or Reject) or close the dialog box before proceeding.

In addition to displaying product recommendations that have been defined, the Messages list also automatically displays a recommendation for every product promotion that the customer is eligible for. This functionality only applies to product promotions, which are defined in the Administration - Product screen, Product Promotions view, and not to other types of promotions. For more information about product promotions, see *Siebel Pricing Administration Guide* .

Scenario for Product Recommendations

This topic gives one example of how product recommendations might be used. You might use product recommendations differently, depending on your business model.

A marketing administrator is defining recommendations associated with a 2.0 megapixel digital camera. The product administrator has already defined this product and related products. The marketing administrator is creating recommendations for products that have been defined.

Market research has shown that it is often possible to persuade customers to buy a 3.2 megapixel camera instead of a 2.0 megapixel camera by telling them that higher resolution is needed to print full-sized pictures. The marketing administrator defines a recommendation associated with the 2.0 megapixel camera that has the values:

- **Message Type.** Upsell Recommendation.
- **Recommended Product.** 3.2 megapixel camera that is in the same product line as the 2.0 megapixel camera.
- **Message.** Higher resolution allows you to print pictures up to 8’ x 10’.
- **Score.** 20. (This value is the highest number in this marketing administrator’s scoring system. It is possible to use higher numbers as scores. Without configuration, the accepted scores are 0 to 100.)

Market research has also shown that it is often possible to persuade customers to buy extra memory cards with their digital cameras by telling them that more memory is needed to store large numbers of pictures. The marketing administrator wants this recommendation to be high on the Messages list but lower than the upsell message, because customers must decide what camera they want before they decide what size memory card they want.

For the 2.0 megapixel camera, the marketing administrator recommends a small memory card by defining a recommendation with the values:

- **Message Type.** Cross-sell Recommendation.
- **Recommended Product.** 8 megabyte memory card that is compatible with the 2.0 megapixel camera.
- **Message.** An extra memory card is needed to store more than 20 pictures.
- **Score.** 19.

The marketing administrator goes on to define more recommendations for other products. In addition, the marketing administrator defines a product promotion that offers a 5 percent discount if you buy both a 2.0 megapixel camera and a memory upgrade.

A customer who has seen the 2.0 megapixel camera in a catalog telephones to order it. The call center representative creates an order for the customer and adds this camera as a line item.

The sales representative sees the first recommendation in the Messages list and suggests that the customer may want to buy a 3.2 megapixel camera instead, to print 8x10 pictures. The customer says that it is not necessary to print such large pictures. The sales representative clicks Reject.

The sales representative looks at the recommendation that is now at the start of the Message list and suggests the customer may want to buy a memory card to store more than twenty pictures. The customer agrees that it may be necessary to store more pictures. The sales representative clicks Accept to add the memory card to the order.

The customer now qualifies for the promotional discount for buying this camera and memory card. A recommendation appears in the messages list saying the customer is eligible for this promotion. The customer is glad to receive the discount, and the sales representative clicks Accept to add the product promotion to the order.

Creating Product Recommendations

To improve performance, you must not add more than 10 recommendations for each product. Each recommendation must be eligibility checked and priced, which can slow performance. Users only view the first three or four recommendations in the Messages list, so there is no benefit to defining more than 10.

After creating the product recommendation, you can also specify translations for the message text.

Note: Before the marketing administrator can create product recommendations, the product administrator must have already created all the products involved. For more information about creating products, see *Siebel Product Administration Guide* .

To create a product recommendation

1. Navigate to the Administration - Product screen.
2. In the Products list, select the product that you want to create a recommendation for.
3. Click the Recommendations view tab, then the Messages subtab.
4. In the Recommendations list, add a new record, and complete the necessary fields, described in the following table.

Field	Comments
Score	Enter a score that is used to determine this recommendation's rank in the Message list.
Recommended Product	Select the recommended product.
Message Type	Select the type of the recommendation. Options are Cross-Sell Recommendation and Upsell Recommendation.
Reason	Enter the message to be displayed in the Messages list.

Field	Comments
Effective From and Effective To	Enter the start date and end date of the time period when this recommendation is in effect.

5. To specify translations for the message text that you entered in the Reason field, in the Translations list that appears after the Product Recommendations list, add new records and complete the necessary fields, described in the following table.

Field	Comments
Language	Enter the language code for the translation. For example, if this record contains the French translation of the message text, then enter the code FRA.
Text	Enter the translation of the message text. This translation is displayed in the Messages list of implementation in the specified language.

Turning Off Product Recommendations

Turning off product recommendations can improve performance.

To turn off product recommendations

1. Navigate to the Administration - Runtime Event screen, and then the Action Sets view.
2. In the Actions Sets list, select the record named Product Recommendation.
3. In the child list, in the record named Raise Signal, deselect the Active checkbox.

Guidelines for Implementing Product Recommendations

The following guidelines improve the performance of product recommendations:

- Avoid heavy eligibility and pricing processing.
  - The product recommendation procedure is executed every time the line item focus changes. It must run very quickly, or it will slow performance.
  - If the customer has slow eligibility or pricing logic (for example, an external call), then configure the underlying procedure so that the slow logic is not executed when called from the product recommendation procedure (for example, by passing an input Mode).
  - Alternatively, remove the eligibility and pricing subprocedure calls from the product recommendation procedure.

- Avoid suppressing duplicate messages for business-to-business accounts.
    - Duplicate suppression processing loads all previous responses for the account at the beginning of the call. Loading more than 100 responses will result in a perceptible delay.
    - For business-to-business selling, configure the upsell and cross-sell message types to suppress duplicates for the quote or order instead of for the account. For more information, see *Customizing Duplicate Message Suppression for Product Recommendations*.
- Simplify the variable maps used to construct the shared context variables and row set.
    - Product recommendation only needs the Product Id from the current line item.
    - All other fields are used for eligibility and pricing of the recommendations.
    - If you are using a limited version of eligibility or pricing, then create a special variable map that only loads the variables that are required.
- If the product recommendation procedure does not run quickly enough to be called every time a line item is selected (for example, because it calls an external real-time marketing engine), then reconfigure the run-time events that run the engine:
    - Only generate recommendations when a product is picked.
    - Add a button to generate product recommendations on demand.

Product Recommendation Technical Reference

In order to customize recommendations to suit your business model, you can modify the controlling workflow, the Recommendation Procedure, and the business service used to implement product recommendations. For more information about workflows, see *Siebel Business Process Framework: Workflow Guide* .

The Recommendation Procedure is a PSP procedure. For more information about PSP procedures, see *Siebel Order Management Infrastructure Guide* .

Product recommendations are turned on by activating the Product Recommendation record in the Action Sets view of the Administration - Runtime screen. This activation calls a signal as its only action. This record is active initially, and you can turn recommendations off and on by deactivating and activating it.

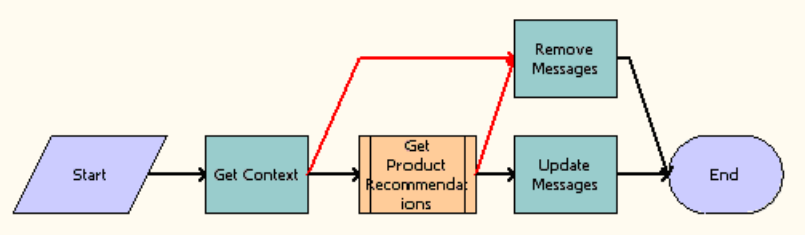
The following topics contain more technical information about product recommendations:

- *Product Recommendation Driver Workflow*
- *Product Recommendation Get Recommended Products*
- *Product UpSell CrossSell Service*

Product Recommendation Driver Workflow

Product Recommendation Driver Workflow is the controlling workflow that is responsible for gathering current context, passing it into the Responsibility Procedure, and passing the result to Unified Message Service.

This workflow is shown in the following figure.



Workflow Description

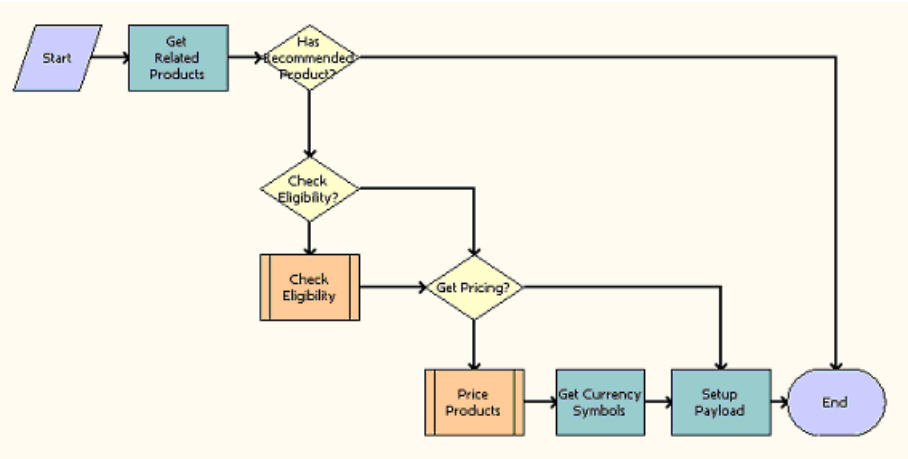
This workflow does the following:

- **Get Context.** Gathers current context. Using Context Service, it collects the quote or order information, and the product information of current line item.
- **Get Product Recommendations.** Calls the PSP process Product Recommendation Get Recommended Products, which uses the product information for the current line item and generates the recommended products.
- **Remove Messages.** Handles errors. If any errors occurred, then this error handling step removes all the messages to prevent showing wrong recommended products.
- **Update Messages.** Using the output of the previous step, populates the recommended products into the user interface.

Product Recommendation Get Recommended Products

Product Recommendation Get Recommended Products is the PSP procedure for recommendations. It is responsible for gathering the recommended products, running them through Eligibility and Pricing, and then constructing them in payload format.

This workflow is shown in the following figure.



Workflow Description

This workflow does the following:

- **Get Related Products.** Finds all the recommended products using the product in the current line item.

- **Has Recommended Product?** Checks if any recommended product exists. If not, skip all the remaining steps.
- **Check Eligibility?** Checks whether the process property EligibilityFlag=Y. If not, skip the Check Eligibility step.
- **Check Eligibility.** Calls the subprocess Product Eligibility & Compatibility - Default, which runs an eligibility test on each recommended products. For more information about this subprocess, see the topic about eligibility in *Siebel Product Administration Guide* .
- **Get Pricing?** Checks whether the process property PricingFlag=Y. If not, skip the Price Products and Get Currency Symbols steps.
- **Price Products.** Calls the subprocess Pricing Procedure - Default, which prices each recommended product according to the current context. For more information about this subprocess, see the technical reference in *Siebel Pricing Administration Guide* .
- **Get Currency Symbols.** Gets currency symbols from the pricing result, and passes them to the payload. The currency symbol is used in the user interface.
- **Setup Payload.** Sets up the required message payload for the message service. This payload is used by the Update Messages step in the driver workflow.

Product UpSell CrossSell Service

This business service includes the following methods:

- *ExecuteUpsell Business Service Method*
- *ExecuteCrossSell Business Service Method*

This business service does not work without the user interface. It works for quote and order entry line items, not for any other business component.

ExecuteUpsell Business Service Method

This method is part of *Product UpSell CrossSell Service*.

Usage

ExecuteUpsell Business Service method is triggered when the user clicks Accept for an upsell recommendation.

This method changes the product on the current line item to the recommended product. Then, depending whether the current product and the recommended product are simple, bundle, or customizable products, it performs the actions described in the following table.

Convert from	Convert to Simple Product	Convert to Bundle Product	Convert to Customizable Product
Simple Product	None.	Generates subcomponents.	Generates default components and attributes.
Bundle Product	Deletes subcomponents.	Deletes original subcomponents that are not in the new bundle. Adds any missing subcomponents.	Deletes original subcomponents of bundle. Generates default subcomponents and attributes of customizable product.

Convert from	Convert to Simple Product	Convert to Bundle Product	Convert to Customizable Product
Customizable Product	Deletes all subcomponents and attributes.	Deletes subcomponents and attributes of customizable product. Generates subcomponents of bundle.	Prompts user to start Configurator.

When one compound product is changed to another, the user must start Configurator to generate missing components and attributes, to delete disallowed components and attributes, and to ensure a valid configuration. As much of the original configuration as possible is retained.

Input

If the business service is called from UMS, then Related Product Id is a required property under the child property set Payload.

If the business service is called from anywhere else, then Related Product Id is the only required property.

Output

None.

ExecuteCrossSell Business Service Method

This method is part of *Product UpSell CrossSell Service*.

Usage

ExecuteCrossSell Business Service Method is triggered when the user clicks Accept for a cross-sell recommendation message.

It adds a new line item for the recommended product to the current quote or order.

Input

If the business service is called from UMS, then Related Product Id is a required property under the child property set Payload.

If the business service is called from anywhere else, then Related Product Id is the only required property.

Output

None.

Configuring Recommendation and Scoring Logic

You can modify the PSP procedure used for product recommendations to call custom recommendation and scoring logic. For example:

- The first step of the PSP procedure gets a list of recommended products. You can modify it to call any business service or workflow to generate this product list.
- The last step of the PSP procedure scores the recommended products to determine their order on the Messages list. You can modify it to call any business service or workflow to score the products.

To avoid degradation of performance, make sure that any custom or external logic executes with acceptable (very low) latency.

Use the Siebel Web Services invocation framework to call external real-time marketing functionality.

For more information about the PSP procedure used for product recommendations, see *Product Recommendation Get Recommended Products*.

Unified Messaging Service and Product Recommendations

Without configuration, product recommendations support two message types, Cross-Sell Recommendation and Upsell Recommendation.

You can modify existing message types, for example to change message response logging and duplicate suppression. You can also define new message types.

The messages use the Unified Messaging Service. For more information about the Unified Messaging Service, see *Siebel Order Management Infrastructure Guide* .

Common tasks for customizing the messaging include:

- *Customizing Message Response Logging for Product Recommendations*.
- *Customizing Duplicate Message Suppression for Product Recommendations*.

Customizing Message Response Logging for Product Recommendations

Without configuration, the Unified Messaging Service logs both Accept and Reject responses for Upsell Recommendations and for Cross-Sell Recommendations.

You can disable logging by editing the message type responses definition.

You can change the field values that are logged by editing the message type payload definition and changing the Product Recommendation PSP Procedure to populate the additional payload fields.

Logged responses are stored in the S_COMMUNICATION table. You can view responses using a Siebel report or Oracle Business Intelligence Applications.

For more information about working with the Unified Messaging Service, see *Siebel Order Management Infrastructure Guide* .

Customizing Duplicate Message Suppression for Product Recommendations

Without configuration, the Unified Messaging Service suppresses the following duplicate product recommendations:

- Upsell recommendations will never be shown to the same account or contact after they have been rejected
- Cross-sell recommendation will never be shown in the same quote or order after they have been rejected. However, if the customer places a new quote or order and picks the same product, then the cross-sell messages that have been rejected in earlier quotes or orders are displayed.

You can edit the message type responses definition to:

- Suppress recommendations after they have been accepted.
- Alter the scope of suppression.
- Disable suppression completely.

Note: Duplicate suppression works only if logging is enabled.

For more information about working with the Unified Messaging Service, see *Siebel Order Management Infrastructure Guide* .

14 Creating a Quote or Sales Order

Creating a Quote or Sales Order

This chapter provides the processes and procedures to create quotes and sales orders. It covers the following topics:

- *Process of Creating a Quote*
- *Process of Creating a Sales Order That Is Not Based on a Quote*
- *Common Tasks for Creating Quotes and Sales Orders*
- *Changing the Default Price List and Rate List*
- *How End Users Work with Product Promotions*
- *Process of Enabling the Automatic Configuration of Customizable Products*
- *Configuring Multiple Customizable Products*

For more information about quotes, see *Working with Quotes*

For more information about orders, see *Working with Sales Orders*

This chapter focuses primarily on sales orders. For details about working with service orders, see *Siebel Field Service Guide* .

Process of Creating a Quote

To create a quote, perform the following tasks.

1. Create the Quote record in one of two ways:
   - *Starting a Quote* You can create a new quote record manually from a number of screens.
   - *Using Auto Quote from an Opportunity* Alternatively, you can create the new quote record by using Auto Quote from an opportunity.
2. *Performing Common Tasks for Creating Quotes and Sales Orders* Because the remaining tasks for entering information about the quote are the same whether you are creating a quote or an order, they are covered in *Common Tasks for Creating Quotes and Sales Orders*.
3. *Verifying a Quote Manually* When you have finished creating a quote, you must verify it to see if you have made any errors.

Starting a Quote

You can start a new quote from a number of screens:

- If you start the quote from the account, contact, or opportunity screen, then your Siebel Business Application automatically copies the account, contact, or opportunity information into the quote. If a price list is associated with the account or opportunity, then it is also entered.
- If you start the quote using the Quotes screen, then you must enter all this information manually.

For information about how to start a quote using an existing asset, see *Using Asset-Based Ordering*

This task is a step in *Process of Creating a Quote*.

To start a quote

1. Perform one of the following actions, depending on which screen you want to use to start the quote:
   - To start a quote from the Quotes screen, navigate to the Quotes screen.
   - To start a quote from an opportunity, navigate to the Opportunities screen, and then the Opportunities List view, drill down on the name of the opportunity for which you want to generate a quote, and click the Quotes view tab.
   - To start a quote for an account, navigate to the Accounts screen, and then the Accounts List view, drill down on the name of the account for which you want to generate a quote, and click the Quotes view tab.
   - To start a quote for a contact, use the site map to navigate to the Contacts screen, Contacts List, and then the Quotes view, and in the Contacts list, select the name of the contact for which you want to generate a quote.
2. In the Quotes list, add a new record. The information in the following table is entered automatically. Not all of this information is visible in the Quotes views of the Opportunity, Account, or Contact screen, but if you create quotes from these views, then the information is visible when you display the quotes in the Quotes screen.

Field	Comments
Sales Rep	Displays the name of the user who created the quote.
Quote #	Displays a unique application-generated number assigned to the quote when it is created.
Revision	Displays the number indicating which revision of the quote this record is. When the quote is created, this value is 1.
Created	Displays the date the quote was created.
Active	A checkbox indicating whether the quote is active. When you create a quote, it is automatically selected, since the quote is active. **Note:** The quote must be designated as active in order to attach files to it or otherwise update it.
Status	Displays the initial status of the quote, In Progress. The status can be changed later, as the quote process continues. For example, if you convert the quote to an order, then the status changes to Order Placed. Other statuses for the quote are defined by your company. Some of the statuses companies often use include Approved, Rejected, Expired, Placed, and Active.
Effective As Of	Defaults to the created date. You can change this default, if appropriate.

Field	Comments
Effective Through	Defaults to the effective date plus 30 days. You can change this default if the quote is available for a different length of time.
Currency	Displays the default currency for the quote.

3. Enter the following information in the new record:

Field	Comments
Name	Enter a name for the quote. If you do not enter a name, then the quote number is assigned as the name.
Price List	Enter a price list to be used for the quote, if one is not assigned automatically through association with the account. Available price lists are restricted by the currency associated with this quote.
Discount	(Optional.) Enter a Override Policy Discount discount for this quote. This discount is used instead of any pricing rules associated with a price list. For more information, see *Applying a Manual Discount*. **Note:** If you created the quote from the Quotes views of the Opportunity, Account, or Contact screen, then you must drill down on the quote name to the quote in the Quotes screen to view this field of the Quote record.
Opportunity	(Optional.) Select an opportunity associated with the quote. The account associated with opportunity is automatically inserted. **Note:** If you created the quote from the Quotes views of the Opportunity, Account, or Contact screen, then you must drill down on the quote name to the quote in the Quotes screen to view this field of the Quote record.
Account	Select the account with which the quote must be associated, if one has not been entered automatically. **Note:** If you created the quote from the Quotes views of the Opportunity, Account, or Contact screen, then you must drill down on the quote name to the quote in the Quotes screen to view this field of the Quote record.
Last Name and First Name	Select the contact for this quote, if one has not been entered automatically.

Field	Comments
	Note: If you created the quote from the Quotes views of the Opportunity, Account, or Contact screen, then you must drill down on the quote name to the quote in the Quotes screen to view this field of the Quote record.
Due	(Optional.) Enter a due date for the quote. **Note:** If you created the quote from the Quotes views of the Opportunity, Account, or Contact screen, then you must drill down on the quote name to the quote in the Quotes screen to view this field of the Quote record.

Using Auto Quote from an Opportunity

If sales representatives are working with an opportunity where the products have already been specified, then they can save time by using the Auto Quote button of the Opportunities screen to generate the quote.

Products are automatically copied from the opportunity to the quote when the Auto Quote check box for the product is selected. When end users create opportunities, they use the Product view of the Opportunity screen to specify products for the opportunity. When they add a product to this list, the Auto Quote check box is automatically selected, but they can exclude a product from a quote by clearing this check box.

This task is a step in *Process of Creating a Quote*.

To use Auto Quote from an opportunity

1. Navigate to the Opportunities screen, and then the Opportunities List view.
2. Click the name of the opportunity for which you want to generate a quote.
3. Click the Quotes view tab.
4. In the Quotes list, click Auto Quote.

   A new quote record appears. The products from the opportunity are copied into the quote.
5. Fill in the Name and Price List fields for the quote.

Performing Common Tasks for Creating Quotes and Sales Orders

After starting the quote, you must enter all the information associated with it. Because the tasks for entering this information are the same whether you are creating a quote or an order, they are covered in *Common Tasks for Creating Quotes and Sales Orders*.

This task is a step in *Process of Creating a Quote*.

Verifying a Quote Manually

In your company, quotes may be verified manually or may be verified automatically through a custom workflow process. Salespeople must check with the application administrator to find out how the verification process is handled in the company and whether they must verify quotes manually.

This task is a step in *Process of Creating a Quote*.

Salespeople can also verify quotes while working with them. For example, they can verify a quote after changing it, to see if the change has introduced errors.

When an end user verifies a quote manually, the application:

- Verifies that the price list is valid. This includes checking that:
  - The price list exists in the dialog box
  - The price list starts before the quote's start date and ends after the quote's end date (if the quote start date and end date exist)
- Verifies that the products in the quote's line items are valid. This verification includes checking that:
  - The product exists in the dialog box
  - The product starts before the quote's start date and ends after the quote's end date (if the quote start date and end date exist)
  - The product has valid attributes
  - The product has the correct list price
- Checks to see if a configurable product has been added. If it has, then verifies that all required components are included
- If a purchase order has been entered as a payment type, then checks the customer's credit status,

Use the following procedure to manually verify a quote.

To verify a quote manually

1. Navigate to the Quotes screen, and then the Quotes List view.
2. In the Quotes list, click the name of the quote you want to verify.
3. Click the Line Items view tab.
4. In the Quote header, click Verify.

   Your Siebel Business Application checks the validity of the products, price lists, and totals shown in the quote. When verification is complete, the application displays a message. Any problems in the quote are detailed.

Use the following procedure to manually verify a quote line item.

To verify a quote line item manually

1. Navigate to the Quotes screen, and then the Quotes List view.
2. In the Quotes list, click the name of the quote you want to verify.
3. Click the Line Items view tab.
4. In the Quote header, click Verify.

Your Siebel Business Application checks the validity of the products and price lists for the line item and the customer's eligibility to purchase the line item. When verification is complete, the application displays a message. Any problems in the line item are detailed.

Process of Creating a Sales Order That Is Not Based on a Quote

This topic describes how to create a new order that is not based on a quote.

> **Note:** If you have already created a quote, then create the order by converting that quote to an order, rather than by using the procedures in this topic. The order automatically includes all the details from the quote, so you do not have to reenter them. For more information, see *Converting a Quote to an Order*.

Creating an order that is based on a quote allows you to verify the quote to check for any errors. If you are creating a complicated order that might have errors, then you must do it by creating quote, verifying the quote, and then converting the quote to an order. For more information, see *Process of Creating a Quote*.

To create an order that is not based on a quote, perform the following tasks:

1. *Starting an Order*. You can create a new order record from a number of screens.
2. *Performing Common Tasks for Creating Quotes and Sales Orders*. Because the remaining tasks for entering information about the order are the same whether you are creating a quote or an order, they are covered in *Common Tasks for Creating Quotes and Sales Orders*.
3. *Submitting an Order*. When the customer is ready to place the order, the end user submits the order. For more information, see *Submitting an Order*.

Starting an Order

This topic describes how to start a new order from a number of screens:

- If you start the order from the Account, Service Request, Project, Contact, Campaign, or Contract screen, then your Siebel Business Application automatically copies information from that screen into the order.
- If you start the order using the Orders screen, then you must enter all this information manually.

This task is a step in *Process of Creating a Sales Order That Is Not Based on a Quote*.

For information about how to start an order using an existing asset, see *Using Asset-Based Ordering*.

To start an order

1. Perform one of the following tasks, depending on which screen you want to start the order from:
   - To start an order from the Orders screen, navigate to the Sales Orders screen, and then the List view.
   - To start an order from a quote, account, service request, project, contact, campaign, or contract, navigate to the screen for the object and to the list view, drill down on the name of the record with which the order is associated, and click the Orders view tab.

   > **Note:** If you start the order from a record that has account, contact, or other information, then your Siebel Business Application copies the relevant information into the order.

2. In the Orders list, add a new record. The information in the following table is added automatically.

Field	Comments
Order #	Displays a unique application-generated number assigned to the order when the order is created.
Status	New orders are assigned the status Pending. The status can be changed later, as the order process continues. Some of the statuses companies often use for sales orders include Open, Awaiting Approval, Approved, Complete, Booked, In Transit, Shipped, and Cancelled.
Status as of Date	Initially, reflects the current date and time. When the status is changed, this field also updates to show the date and time when the status changed. **Note:** If you created the order from some screen other than the Sales Orders screen, then you must drill down on the order name to display the order in the Orders screen to view this field.
Priority	The priority of the order. New orders are assigned the priority Medium.
Order Date	The date and time the order was created.
Version	The version associated with the order. When the order is created, this value is typically 1. The number is incremented every time the order is revised. For details see *Revising an Order*.
State	The state of the order. For example: open, closed, pending. Note that the state might not be the same as the status.

3. In the Type field, select the type of order you are creating.
4. If you are in the Orders list at the end of the Accounts screen, the Quotes screen, the Service Request screen, or another screen, then drill down on the order number to display the Order screen.
5. In the Sales Order form, click the Show More button.

6. In the Sales Order form, review the information that has been copied into the order, and make any changes to it, as appropriate. Some fields are described in the following table.

Field	Comments
Last Name	Select the contact for this order, if it has not been entered automatically.
Account	Enter the account to which this order is sold, if it has not been entered automatically.
Site	When an account has more than one address, select the correct address for this order.
Currency	Select the currency for the order.
Price List	Select the price list associated with this account. Price lists that can be assigned to this order are restricted by the currency.

Performing Common Tasks for Creating Quotes and Sales Orders

After you start the order, you must enter all the information for it. Because the tasks for entering this information are the same whether you are creating a quote or an order, they are covered in *Common Tasks for Creating Quotes and Sales Orders*.

This task is a step in *Process of Creating a Sales Order That Is Not Based on a Quote*.

Submitting an Order

When the customer is ready to place the order, the end user submits the order.

This task is a step in *Process of Creating a Sales Order That Is Not Based on a Quote*.

For information about how to submit an order, see *Submitting an Order*.

Common Tasks for Creating Quotes and Sales Orders

After you create either a quote or an order record, complete the following common tasks:

1. *Adding Line Items to a Quote or an Order*
2. *Working with Line Items of Quotes and Orders*
3. *Selecting a Nondefault Price List*
4. *Viewing and Recalculating Prices*
5. *Applying a Manual Discount*

6. *Adding Shipping Information*
7. *Calculating Shipping Charges*
8. *Calculating Taxes*
9. *Checking Availability to Promise (ATP)*
10. *Entering Payment Information for Quotes or Orders*

Adding Line Items to a Quote or an Order

Users can add products to a quote or order as line items in the following ways:

- *Adding Multiple Items at One Time*
- *Adding Line Items One at a Time*
- *Adding Line Items by Browsing the Catalog*
- *Adding Line Items by Using a Favorite*
- *Adding Write-In Products*
- *Adding a Package Line Item*
- *Adding a Service Product to a Quote or Order*

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

Adding Multiple Items at One Time

The Add Items feature allows an end user to add multiple items at one time and to specify the quantity and attributes for those items at the same time.

> **Note:** Your company can set a flag in Siebel Tools to show only the products in the selected price list. For more information, see *Using Siebel Tools* .

To add a product to a quote or order using Add Items

1. Navigate to the quote or order to which you want to add items.
2. Click the Line Items view tab.
3. In the Line Items list, select Add Items from the menu.
4. In the Pick Products dialog box, select the products to add to the quote.
   a. Enter the quantity in the Order Qty field for the products you want to add and click Add.
   b. To specify attributes for a product, click Attributes in the link bar and specify the attributes in the Line Item Attributes list.
   c. When you are finished specifying product quantity and attributes, click OK.

   The products appear in the Line Items list.

Adding Line Items One at a Time

The New Record menu option allows an user to enter one record at a time. This method is useful when the end user already knows the product name.

To add items one at a time

1. Navigate to the quote or order to which you want to add items.

2. Click the Line Items view tab.
3. In the Line Items list, click New.
4. In the Line Items list, in the Product field, select the product.
5. Make any necessary changes to the quantity or attributes.

Adding Line Items by Browsing the Catalog

While end users are creating quotes or orders, they can select products by browsing the product catalog. Some products may also have deals associated with them, which the end user can select to obtain special pricing for a customer. For more details about working with the Catalog, see *About Dynamic Product Catalogs*.

You add line items differently if you are using Siebel High Interactivity User Interface or Siebel Open UI.

To to add line items by browsing the catalog using Siebel Open UI

1. Navigate to the quote or order to which you want to add items.
2. Click the Catalog view tab.
3. Use the Catalog drop-down list to select one of the available catalogs.
4. Use the Category Explorer Selector to select a category in the catalog.

   The products in that category appear.
5. Navigate through the list of products to find the one you want.
6. If the product is a promotion, you can use the Edit Promotion control to edit the promotion.
7. If the product is customizable, you can use the Customize control to customize it.
8. Click Add Item to add a product to the Shopping Cart.

To to add line items by browsing the catalog using Siebel High Interactivity User Interface

1. Navigate to the quote or order to which you want to add items.
2. Click the Catalog view tab.
3. In the Browse list, click one of the Categories.

   The products in that category appear.
4. If you want to select a simple product with attributes, then you can modify the values of the attributes in the Items list.

   Modifying attributes in the catalog is optional. You can also select attributes after adding the product as a line item.
5. Click Add Item to add a product.

Adding Line Items by Searching the Catalog

While end users are creating quotes or orders, they can select products by browsing the product catalog. Some products may also have deals associated with them, which the end user can select to obtain special pricing for a customer.

For more details about working with the Catalog, see *About Dynamic Product Catalogs*.

To search the catalog to add line items

1. Navigate to the quote or order to which you want to add items.
2. Click the Catalog view tab.
3. In the Catalog view link bar, click Search.
4. In the Search form, enter criteria in one or more search fields, and click Search.
5. In the Products list, select a desired product.

6. If this product has attributes, then select the values of the attributes in the Product Attributes list, which appears after the Products list.

   Selecting attributes in the catalog is optional. You can also select attributes after adding the product as a line item.
7. In the Products list, click Add Selected Items.

Adding Line Items by Using a Favorite

Product selection favorites are groups of products and attributes that have been saved for reuse:

- Some product selection favorites are public, and are available to anyone using the Siebel Business Application. Public templates are created by the product administrator.
- Other product selection favorites are private, and are available only to a specific user. Private favorites are templates that end users create for themselves.

When the end user adds a favorite, the items and associated attributes are copied into the quote or order as line items. The end user can make any additional changes to the line items, such as configuring a product with attributes or components. For information about creating favorites, see *Creating Product Selection Favorites*.

To add a favorite to the current quote or order

1. Navigate to the quote or order to which you want to add items.
2. Click the Catalog view tab.
3. In the Catalog view link bar, click Favorites.
4. To quickly add a product, enter the information in the Product Name or Part Number text box, and click Add Item.
5. Alternatively, add items using the Favorites list:
   a. In the Favorites list, select a record.

      The line items for the selected record are displayed in the Products list, which appears after the favorites list.
   b. Select one or more products in the Products list.
   c. If a product has attributes, then select the values of the attributes in the Product Attributes form, which appears after the Products list.

      Selecting attributes in the catalog is optional. You can also select attributes after adding the product as a line item.
   d. Click Add Selected Items.

Use the following procedure to create a private product template.

To create a private product template

1. Make sure the current quote or order contains the line items you want to save in the favorite.
2. In the Quote or Order header summary form, from the menu select Save As Favorite.
3. In the Save As Template dialog box, enter a name for the template and click Save.

   The items in the quote or order are saved as a private template.

   > **Note:** If two quote items have same integration id, then only one quote item is copied to private product template.

Adding Write-In Products

An end user can also create and add a write-in product for items that are not included in the price list.

To add a write-in product

1. Navigate to the quote or order to which you want to add the product.
2. Click the Line Items view tab.
3. In the Line Items list, click New.
4. In the Product field, click the Select button.
5. In the Pick Product dialog box, in the Starting With text box, type **Write** and then click Go.

   A product named Write-In Product is entered as a line item.
6. Enter the price and other information for this product in the Line Item record.

   You can enter the price in the Manual Price Override or Manual Discount Amount field.
7. In the Item Details view, in the Comment field, enter the product's name or description.

Adding a Package Line Item

The end user can use the Line Items list to create a group of products identified as a package. The *package price* is the rolled-up price of all the products in the group. This method is useful when the end user wants to give a price break to a customer using the purchase of two or more items.

To add a package line

1. Navigate to Line Items view for the quote or order where you want to create the package.
2. In the Line Items list, add all the line items that you want to include in the package.
3. Select all the line items that you want to include in the package.
4. From the Line Items menu, select Package.

   All the selected items are added to the package. A new line item record named Package is added, and the selected line items appear indented under it to show that they are part of the package.

Use the following procedure to remove products from a package.

To remove products from a package

1. Navigate to Line Items view for the quote or order that contains the package.
2. In the Line Items list, select one or more products that are part of a package.
3. From the Line Items menu, select Unpackage.

   The selected items are removed from the package and are no longer indented. If you selected all the items in the package, then the line item named Package is deleted.

Adding a Service Product to a Quote or Order

Service products are generally associated with a physical product. For example, a warranty extension for a new computer would be set up as a service product.

When an end user adds a service product to a quote or order, the price of the service product is based on its list price plus a percentage based on a service method. The price, percentage, and service method are all specified in the price list for the service product. The service method specifies how the service price is calculated.

For example, if the list price of a service product is $100, the service method is Net Price (of the covered product) and the percentage is 10, then the price of the service is $100 (its own list price) plus 10 percent of the net price of the product with which it is associated.

You use different methods for adding service products to a quote and to an order.

To add a service product to a quote

1. Navigate to the quote to which you want to add a service product.
2. Click the Line Items view tab.
3. In the Line Items list, select the line item for the covered product for which you want to add a service product.
4. From the Line Items menu, select Service.
5. In the Pick Sales Service Product dialog box, select the appropriate service product and click OK.

   The product is added to the Line Items list and the total price appears in the Totals form.

Use the following procedure to add a service product to an order.

To add a service product to an order

1. Navigate to the order to which you want to add a service product.
2. Click the Line Items view tab.
3. In the Line Items list, add the product with which the service product will be associated.
4. In the Line Items list, add a record for the service product.

   a. In the Product field, enter the name of the of service product.

      The base price of the service product appears.

   b. In the Covered Product field for the service product, click the select button.

      > **Note:** The Covered Product field can only be edited when the line item is a service product. For other products, this field is read-only.

   c. In the dialog box, select the physical product with which this service product is associated and click Add.

      The price now reflects the service method and percentage, as well as the base price.

Working with Line Items of Quotes and Orders

After adding products to a quote or order as line items, users may need to work with the line items in the following ways:

- *Viewing the Price Waterfall*
- *Deleting a Line Item*
- *Canceling an Order Line Item*
- *Customizing a Product*
- *Changing Component Quantities*

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

Viewing the Price Waterfall

The end user can view a *price waterfall*, which displays all the steps that went into the calculation of the line item, beginning with the start price and listing all the discounts, markups and final price.

The waterfall can be used to answer customers' questions about how prices were calculated.

To view a price waterfall

1. Navigate to the quote or order.
2. In the Line Items list, click the net price of a line item.

   A waterfall dialog box appears, listing all the steps in determining that net price.

Deleting a Line Item

Sometimes an end user may need to delete a line item to remove a product from a quote or an order.

> **Note:** To delete a component of a product with components, the end user must do so in the configuration session. The end user cannot delete a component by deleting a record in the line items list, as described in the following procedure.

To delete line items

1. Navigate to the quote or order.
2. Click the Line Items view tab.
3. In the Line Items list, select the item.
4. Click Delete.

Canceling an Order Line Item

If an order status is Booked or Awaiting Shipping, then a customer service representative or sales agent can cancel one or more line items within an order instead of canceling the entire order.

To cancel order line items

1. Navigate to the relevant order line item.
2. Check Request To Cancel for the order line.
3. Select a reason from the Cancel Reason list.
4. Click Submit to submit the revised order.

   > **CAUTION:** The Cancel Order button cancels the entire order.

Customizing a Product

End users can customize some products in a quote or order by selecting their components. For instance, if a customer is buying a computer, then the customer can select components such as the type of monitor, hard drive, and so on. To create the quote or order, the end user first adds the computer system as a product line item, and then customizes it using Siebel Configurator.

The end user can tell when a product can be configured because the Customize button is enabled when the product record is selected. When a customizable product has components, the end user can expand the item to see the components included in the customizable product.

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

Note: The customize feature is only available if you have licensed Siebel Configurator.

To customize a product

1. Navigate to the quote or order that contains the product that you want to configure.
2. In the Line Items list, select the line item that contains the product with components, or add a new record for a product with components.
3. With the customizable product selected, click Customize to start a configuration session.
4. In the selection pages, check to see that the item is configured correctly and make any necessary or desired changes.
5. If you are working in a quote, then click the menu button in the quote header and then click Verify.

   Note: When you choose the Verify menu option in a quote, the quote is verified as complete or incomplete. Part of the verification process checks to see whether the product with components is incomplete. If you removed an item from the product with components that is part of the minimum configuration, then you must click Configure again to start another configuration session and add it back in.

6. When the product is configured properly, click Submit.

Changing Component Quantities

A customer might want to order a different number of components than what is set as the default in a product with components.

Typically, end users change the quantity of a component in the configuration session that starts when they click the Customize button in the Line Items list. When an end user clicks Customize, Siebel Configurator checks all configuration rules to make sure that the change in quantity is appropriate.

Your company can decide to allow changes to component quantities directly in the Line Items list in the quote or order by setting a user property in Siebel Tools. For more information, see *Configuration with Siebel Tools*.

When this user property has been set, the Qty field in line item for components of products with components is editable. If the Qty field is read-only, then this user property has not been set.

Selecting Product Attributes for Line Items

Some products have attributes that the end user can choose. For example, a shirt may have the attribute of size, with the possible values S, M, L, and XL.

If the product has both components and attributes, then the user selects both the attributes and components during the configuration session, as described in *Customizing a Product*.

If the product is a simple product with attributes, meaning that it has attributes but does not have components, then the user can either:

- Select the attributes in the catalog, before adding the product to the quote or order, as described in *Adding Line Items by Browsing the Catalog* and *Adding Line Items by Searching the Catalog*.
- Select the attributes for the line item, after adding the product to the quote or order, as described here.

To select attributes for a line item

1. Navigate to the line item.
2. In the link bar of the Line Items view, click Attributes.

3. In the Line Items list, select the line item whose attributes you want to choose.
4. In the Line Items Attributes list, which appears after the Line Items List, specify the values for the attributes. as described in the following table.

Field	Description
Name	Displays the name of the attribute
Value	Select the value of the attribute
Action	Displays the action code for the line item. For more information about action codes, see *Action Codes*.
Data Type	Displays the data type of the attribute. For example, if the user chooses yes or no, then the data type of the attribute is boolean; if the user chooses among options that are written out, then the data type of the attribute is text.

Selecting a Nondefault Price List

When products are added to a quote or order, their prices are based on the price list associated with the order. The default price list is typically based on the account, but sales representatives can also select a different price list in the quote or order.

For more information about price lists, see *Siebel Pricing Administration Guide* .

To select a nondefault price list

1. Navigate to the quote or order.
2. In the Quote form or the More Info form, in the Price List field, select a price list.

Viewing and Recalculating Prices

The user can view the prices many times while developing a quote and order.

Prices are recalculated automatically when the user performs certain actions. The user can also recalculate prices manually at any time, to see the latest price.

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*

This topic describes how the user can view and recalculate prices. It includes the following subtopics:

- *Factors That Impact Prices*
- *When Prices Are Recalculated*
- *When to Manually Recalculate Prices*

Factors That Impact Prices

The prices for quotes and orders are based on:

- **Price list.** Determines the list price for each product and the promotional prices for products that have them. For information about setting up price lists, see *Siebel Pricing Administration Guide* .
- **Application-based policy discounts.** Discounts and markups that are applied automatically, such as volume discounts and deals associated with the account contacts. For information about setting up these policy discounts, see *Siebel Pricing Administration Guide* .
- **Manual discounts.** Discounts that are applied manually by sales representatives while they are placing an order, in addition to the application discounts that are defined by the pricing administrator and applied automatically. For information about manual discounts, see *Applying a Manual Discount*.
- **Shipping charges.** Determined by the shipping carrier and method indicated on the Fulfillment view tab for the quote or order. For information about shipping charges, see *Calculating Shipping Charges*.
- **Taxes.** Calculated on each line item in the quote or order. For information about taxes, see *Calculating Taxes*.

When Prices Are Recalculated

Prices in a quote or order are automatically recalculated:

- Each time the end user adds an item to a quote or order
- When the end user changes the quantity for a line item
- When the end user changes the price list in the header
- When the end user verifies a quote
- When the end user adds a service product associated with a covered product

Note: Line items are excluded from repricing when the Override Policy Discount field is selected. This field is selected when the end user applies a manual discount to the line item. For more information, see *Applying a Manual Discount*. If a user wants a line item to be included in a repricing calculation, then the user must be sure that the Override Policy Discount field is not checked.

When to Manually Recalculate Prices

At times when prices are not automatically recalculated, end users can manually reprice an individual line item or all line items.

End users must reprice individual line items when they have changed a parameter at the line level (other than product or quantity) that could affect price.

End users must reprice all line items when:

- They want to have a bundle or aggregate-type pricing rule triggered.
- They have changed a field on the quote or order that may affect the price. For example, changing the account might result in using different pricing factors that are associated with account.

Note: If a quote or order included a product with components, then if you reprice one of the component products, the parent product with components is automatically repriced. For example, if the quote or order includes a computer that is a product with components and includes the monitor, mouse, and keyboard that are components of this product, then if you reprice one of the components, the entire computer is also reprices.

To reprice an individual line item

1. Navigate to the quote or order.
2. Click the Line Items view tab.
3. In the Line Items list, select the line item.
4. In the Line Items list menu, select Reprice.

Use the following procedure to reprice all line items.

To reprice all line items

1. Navigate to the quote or order.
2. From the Line Items list menu, select Reprice All.

Applying a Manual Discount

The net price of a line item in a quote or order is determined by:

- **Start price.** The calculation begins with the start price, which is typically the list price or promotional price of the product in the price list. Attribute adjustments and customizable product adjustments affect the start price.
- **Policy discounts.** The application applies discounts such as promotions, contract prices, and volume discounts.

Manual discounts. The salesperson applies manual discounts. End users can enter manual discounts for individual line items or for the entire quote or order.
For more information about start price and policy discounts, see *Siebel Pricing Administration Guide* .

When users work with pricing information, they can use the fields and button listed in the following table.

Field Name	Comments
Policy Discounts field	Displays a money amount equal to the sum of the discounts that the application has applied automatically, including promotional discounts, contractual discounts, volume discounts, and so on.
Manual Discounts field	Displays a money amount equal to the manual discount that the user applied in one of the manual discount fields (Manual Discount Amount, Manual Discount %, Manual Price Override)
Net Price field	If licensed for Siebel Dynamic Pricing, then in the Discount Negotiation view, the end user can click the Net Price field to display a pricing waterfall that shows how all the adjustments were used to calculate the net price. The net price is the final price arrived at after applying all discounts to the start price.
Manual Discount Amount field	The end user can enter a manual discount amount for a line item. (The three fields that apply discounts to individual line items, Manual Discount Amount, Manual Discount % and Manual Price Override, are mutually exclusive. The end user can only use one of them.)
Manual Discount % field	The end user can enter a manual discount percentage for a line item.
Manual Price Override field	The end user can specify an override price for the line item. This price is used as the net price for the line item. Specifying a value in this field overrides any applicable policy discounts.

Field Name	Comments
Discount field	In the Discount field of the quote or order header, the end user can enter a discount percentage in the that applies to all the line items in the quote or order. However, this discount does not apply if you enter a manual discount for the line item, which overrides the discount entered in the header.
Header Discount Amount	If you enter a percentage in the Discount field of the header, then this field of the Line Item record displays a money amount equal to the value of the header Discount for the line items. To display this field, you may have to select Columns Displayed from the Line Items list menu.
Margin % field	Displays the profit margin at the current price. The margin is calculated as (Net Price - Cost)/Net Price * 100%
Spread Discount Button	In the totals form for the Line Items, the user can click the Spread Discount button and enter a total discount amount, percent, or target total for the entire quote or order. The discount is spread among the line items. **Note:** Spread discount is the best practice for entering a discount for the entire quote or order. It is recommended that end users use this method instead of using the header Discount field.
Override Policy Discount field	Selecting this check box overrides any policy discounts such as volume discounts and contractual discounts, generated by the application. The application does not apply policy discounts to that line item. Discounts depend only on manual discounts. Use this flag if you want the Manual Discount Amount or Manual Discount % to replace the policy discounts rather than being applied in addition to the policy discounts. This flag is deselected by default. Whenever the user selects or deselects this flag, the item is repriced. In earlier releases, this field was named Keep Discount.

These discounts are applied by the default pricing procedures, which may be modified to change the way the net price is calculated. For more information, see the workflow and Pricing Procedure reference in *Siebel Pricing Administration Guide* .

The following topics discuss the different ways an end user can enter a manual discount:

- *Entering a Manual Discount for an Individual Line Item*.
- *Spreading a Manual Discount Across Line Items*
- *Using Discount Negotiations View*

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

Entering a Manual Discount for an Individual Line Item

End users can apply manual discounts to individual line items.

To enter a manual discount for an individual line item

1. Navigate to the quote or order with which you want to work.
2. Click the Line Items view tab.

3. In the Line Items list, select the line item to which you want to apply a discount.

   Note: You can also apply a manual discount to an individual component of a configurable product.

4. In the link bar, click the Line Details.
5. In the Line Details form, enter information in one of the fields described in the following table:

Field	Comments
Manual Discount %	Enter a number that is subtracted from the price as a percentage discount. For example, if you enter 10, then a 10% discount is subtracted from the net price.
Manual Discount Amount	Enter a number that is subtracted from the net price as an absolute amount. For example, if you enter 10, then a $10 discount is subtracted from the net price (assuming that the price is in dollars).
Manual Price Override	Enter a number that is used as a discount price that overrides the net price. For example, if you enter 10, then the discount price for the line item is $10 net price (assuming that the price is in dollars). Specifying a value in this field overrides any applicable policy discounts.

Note: These fields are mutually exclusive, so you can only enter a value in one of them.

Spreading a Manual Discount Across Line Items

You can enter a manual discount that is spread across all line items in two ways:

- *Using the Discount field in the Header*
- *Using the Spread Discount Button*

Using the Discount field in the Header

The Discount field allows you to enter a percentage discount to line items for quotes and orders.

If you enter a discount by using this field and then enter a manual discount for the line item, using the line item's Manual Discount Amount, Manual Discount %, or Manual Price Override field, then the line-item level manual discount overrides the discount entered in the header Discount field.

If you use the Discount field, then the discount applies to all products of any price type. That is, it applies to products with the price type One-Time, Recurring and Usage. When you use this field, be careful that it does not create undesired discounts for products with the price type Recurring or Usage.

To enter a percent discount at the header level for line items of a quote or order

1. Navigate to the Quotes or Orders screen, locate the quote or order with which you want to work, and drill down on the Name link for the quote or order.
2. In the Quote or Order header, click the Show More button, and in the Discount field, select or enter the discount (a percentage value) that you want to apply to the line items in the quote.
3. Add the products that receive this discount. For information about adding products, see *Adding Multiple Items at One Time*.

Using the Spread Discount Button

End users can also click the Spread Discount button and:

- Enter a discount amount discount percent, or target price.
- Spread this discount across all line items, across selected line items, or across line items of a particular type (products, services, or training).

If you enter a discount amount or target total price, then the discount is spread across the line items in proportion to the total price selected as source, which could be list price or net price.

Spread Discount updates the Manual Discount Amount field on a line item, unless the line item already has a Manual Price Override specified, in which case it updates the Manual Price Override.

The spread discount can apply to hierarchical products and services. For example, a computer is a product made of many components. If a customer buying twenty computers gets a discount, then you can enter the discount for the entire quote. When you click Spread, the discount is spread among the twenty computers and among the components of each computer.

The discount is spread across products with the price type One-Time. Spread discount does not involve products with the price type Recurring and Usage. The Current Total field on the popup applet shows the total of all selected products with the One-Time price type.

There is special behavior for the current total field, when the discount is applied to customizable Products and their components. For example, assume quote contains a customizable product which has 10 component products. The end user selects the root customizable product and two of the component products, and then clicks Spread Discount. The current total field on the popup applet shows the total net price of the customizable product and all of its components, not just the total of just the three items that were selected.

When maximum and minimum prices are configured, the application will honor these rules while spreading the discount. For more information on configuring minimum and maximum prices, see the chapter on creating and assigning price lists in *Siebel Pricing Administration Guide* .

Spread Discount honors minimum prices. It discounts a line item until the minimum price is reached. Then it recalculates and spread the remaining discount amongst the other line items.

If you enter a Discount Amount or Target Total, then Spread Discount first calculates a weighted average before distributing discounts. It calculates the weighted average as follows:

- If the Source is List Price:

```
Weighted Average = Line Item List Price / Total List Price
Manual Discount Amount = Weighted Average * Discount Amount
```

- Source = Net

```
Weighted Average = Line Item Net Price / Total Net Price
Manual Discount Amount = Weighted Average * Discount Amount
```

If the spread discount type is Selected Items and one of the selected items is the root of a product with components, then the discount percent applies to or the discount amount is spread across the entire instance of the selected product with components.

To spread discounts across quote or order line items

1. Navigate to the desired quote or order.

2. Click the Line Items view tab.
3. To spread the discount among selected items, select multiple items in the Line Items list.
4. In the Totals form under the Line Items list, click Spread Discount.
5. To spread the discount among items that are a specific product type, select that type in the Product Type list.
6. In the Spread Discount dialog box, complete the necessary fields and click Spread. Some fields are described in the following table:

Field	Comments
Type	Select which items to which you want to spread the discount. The options are: o **Entire Document.** Spreads discount among all the line items. This value is the default if items are not multi-selected. o **Selected Items.** Spreads discount among the selected line items. o **Product.** Spreads discount among all line items that are products. o **Service.** Spreads discount among all line items that are services. o **Training.** Spreads discount among all line items that are training.
Current Total	Displays the quote total before the spread discount is applied, if the type is Entire Document. Displays the total of the selected items before the spread discount is applied, if the type is Selected Items. If one of the selected items is a customizable product, then displays the total for the entire customizable product, not just for components that are selected.
Source Price Field	Select List Price or Net Price to determine how the discount is calculated. If you are using a Discount Percent, then it is calculated as follows: o Source = List Price - Manual Discount Amount (MDA) = Previous MDA + (Discount Percent * List Price) - Net Price = Net Price – (Discount Percent * List Price) o Source = Net Price - Manual Discount Amount (MDA) = Previous MDA + (Discount Percent * Net Price) - Net Price = Net Price – (Discount Percent * Net Price) If you are using a Discount Amount, then it is calculated as follows: o Source = List Price - Manual Discount Amount (MDA) = Previous MDA + ((List Price/Total List Price) * Discount Amount) - Net Price = Net Price – ((List Price/Total List Price) * Discount Amount) o Source = Net Price - Manual Discount Amount (MDA) = Previous MDA + ((Net Price/Total Net Price) * Discount Amount) - Net Price = Net Price – ((Net Price/Total Net Price) * Discount Amount) If you are using a Target Total, then it is calculated as follows:

Field	Comments
	○ Source = List Price - Manual Discount Amount (MDA) = Previous MDA + ((List Price/Total List Price) * (Current Total – Target Total)) - Net Price = Net Price – ((List Price/Total List Price) * (Current Total – Target Total)) ○ Source = Net Price - Manual Discount Amount (MDA) = Previous MDA + ((Net Price/Total Net Price) * (Current Total – Target Total)) - Net Price = Net Price – ((Net Price/Total Net Price) * (Current Total – Target Total))
Discount Amount	Enter the total amount of the discount. This amount is subtracted from the current total and spread among the line items. (If you enter this value, then you must not enter a Discount Percent or Target Total.)
Discount Percent	Enter a percent discount. This discount is applied to the line items. It applies to the source specified in the Source Price field, but it updates the net price. (If you enter this value, then you must not enter a Discount Amount or Target Total.)
Target Total	Enter the total price for the quote after the discount. The difference between the current total and the target total is spread among the line items. (If you enter this value, then you must not enter a Discount Amount or Discount Percent.)

After you click Spread, your Siebel Business Application performs the spread and enters unit net price and discount amount for each line item. If the line items previously had individual discounts, then they are updated based on this spread.

Using Discount Negotiations View

If you have licensed Siebel Dynamic Pricing, then the end-user can use the Discount Negotiations view as a dashboard for negotiating discounts, gaining visibility into prices, and communicated those prices and discounts to customers.

Some key features of the line items list in the Discount Negotiation View include:

- **Pricing waterfall.** In the discount negotiation view, the user can click the Net Price field to display the net price waterfall, which shows all adjustments that were applied to arrive at the final net price
- **Margin % field.** Displays the profit margin for the line item. This value is calculated as (Net Price - Cost)/Net Price * 100%
- **Other key features.** The Discount Negotiation form provides sales users with referential price data, actionable price and discount data, as well as a quick view of how the net price was calculated.

The following fields in the discount negotiation form provide end users with data useful for negotiating discounts:

- Referential Pricing Data:
  - **Cost.** The cost of the product from the price list
  - **List Price.** The list price of the product from the price list
  - **Minimum Price.** The floor price of the product as set by Product or Pricing Administrators in the Price List.

- **Maximum Price.** The ceiling price of the product as set by Product or Pricing Administrators in the Price List.
- **Current Volume Discount.** The current volume discount that has applied to the line item, if applicable
- **Next Volume Discount.** The next volume discount that would apply if to the line item if the quantity were incremented to meet the next volume discount requirements
- **Upsell.** Upsell message for next volume discount quantity break.
- **Margin %.** Displays the profit margin for the line item. This value is calculated as (Net Price - Cost)/Net Price * 100%
- **Line Discount %.** Displays the total discount % for the line item. This value is calculated as (Total Line Discounts/Net Price) * 100%

- Actionable Pricing and Discount Data:
  - **Price List.** Allows the user to choose a price list for the line item that may differ from the price list on the header
  - **Currency.** Allows the user to choose a currency code for the line item that may differ from currency on the header. This option allows the sales user to quote individual line items on the same quote or order in different currencies.
  - **Entitlement.** Allows the user to associate a pricing entitlement with the line item.
  - **Promotion.** Allows the user to associate a promotion with the line item.
  - **Manual Discount Amount.** The end user can enter a manual discount amount for a line item. (The three fields that apply discounts to individual line items, Manual Discount Amount, Manual Discount % and Manual Price Override, are mutually exclusive. The end user can only use one of them.)
  - **Manual Discount %.** The end user can enter a manual discount percentage for a line item.
  - **Manual Price Override.** The end user can specify an override price for the
  - **Line item.** This price is used as the net price for the line item. Specifying a value in this field overrides any applicable policy discounts.
  - **Override Policy Discounts.** Selecting this check box overrides any policy discounts such as volume discounts and contractual discounts, generated by the application. The application does not apply policy discounts to that line item. Discounts depend only on manual discounts. Use this flag if you want the Manual Discount Amount or Manual Discount % to replace the policy discounts rather than being applied in addition to the policy discounts. This flag is deselected by default. Whenever the user selects or deselects this flag, the item is repriced. In earlier releases, this field was named Keep Discount.

- Data used to calculate the net price:
  - **Start Price.** The start price of the product, including any customizable product adjustments or attribute adjustments.
  - **Policy Discounts.** The sum of discounts applied by the application (such as volume discounts, contractual discounts, and so on.)
  - **Manual Discounts.** The sum of manual discounts applied by the user.
  - **Net Price.** The Net Price of the item. Calculated as Start Price - Policy Discounts - Manual Discounts.
  - **Qty.** The quantity of the item.
  - **Extended Net Price.** Calculated as Extended Qty * Net Price
  - **Sub Total.** The One-Time total. If the line item is a customizable product, then this value is the Sub Total of the Net Price for the customizable product and any components with a Price Type of One - Time.
  - **Monthly Sub Total.** The Monthly Sub total for the line item. If the line item is a customizable product, then this value is the Sub Total of the Net Price for the customizable product and any components with a Price Type of Per Month.

These fields of the Discount Negotiation form provide a top-down (highest to lowest) view of how the net price was calculated.

Using this view, the salesperson can see what the profit margin for the product is with the policy discounts that the pricing administrator has set and with different manual discounts that the salesperson is considering offering to the customer. The salesperson can set the manual discounts in this view, and see the margin for each.

To use the Discount Negotiations view

1. Navigate to the quote and line item that you are negotiating the discount for.
2. Click the Pricing view tab.

The Discount Negotiation form is under the Line Items list.

Example of Calculation of Discount Amounts

This topic gives an example of how the application calculated the discount price for three products, based on:

- **Start Price.** $100 list price for each product, defined by the pricing administrator.
- **Policy discounts.** Contractual and market discounts that the pricing administrator defined. Contractual discounts are applied first.
- **Manual Discounts.** Specified by the end user.

The data for the three products in the example is shown in the following table.

	Product 1	Product 2	Product 3
Contractual Discount	10%	None	10%
Market Discount	$10 off	$5 off	None
Manual Discount Amount	$10	None	None
Manual Discount %	None	10%	None

	Product 1	Product 2	Product 3
Manual Price Override	None	None	$5

The prices calculated for the three products using this example data is shown in the following table. This table assumes that the pricing procedure has been set up to apply the contractual discount first and then the market discount. It also assumes that the user did not select the Override Policy Discount Field.

Product	Start Price	Policy Discounts	Manual Discounts	Net Price	Price Calculation
Product 1	$100	$20 ($10 Contractual + $10 Market)	$10 entered in Manual Discount Amount field	$70	$100 (Start Price) – 10% (Contractual) - $10 (Market) - $10 (Manual Discount Amount)
Product 2	$100	$5 (Market)	10% entered in Manual Discount % field	$85.50	$100 (Start Price) - $5 (Market) – 10% (Manual Discount %)
Product 3	$100	$10 (10% Contractual)	$5 entered in Manual Price Override field	$95	$95 (Manual Price Override, overrides start price and policy discounts.)

If the end user selected the Override Policy Discount field, then the products would have the prices shown in the following table. In every case, the price calculation would just be based on the start price and manual discount and would ignore any application generated discounts such as Volume Discounts, Contractual Discounts, and so on.

Product	Start Price	Manual Discounts	Policy Discounts	Net Price	Price Calculation
Product 1	$100	$10 entered in Manual Discount Amount field	0	$90	$100 (Start Price) – $10 (Manual Discount Amount)
Product 2	$100	10% entered in Manual Discount % field	0	$90	$100 (Start Price) – 10% (Manual Discount %)
Product 3	$100	$5 entered in Manual Price Override field	0	$95	$95 (Manual Price Override, overrides start price)

Adding Shipping Information

End users specify the following shipping information for a quote or order:

- Ship-to information, including account name, contact name, and address.
- Shipping method, terms, and carrier, which are factor in shipment cost.
- Whether all the items must be shipped at the same time (Shipping Complete).

- Whether all the items must be delivered at one time (Single Delivery).

When working with a quote, the user specifies one shipping address for the entire quote. When working with an order, the user can specify different shipping addresses for individual line items.

When working with an order, after entering the necessary information, the user clicks Create Shipment to generate shipment records. The application creates one shipment record for line items that have the same values in the following fields: Request Delivery Date, Ship-To Address, Source (Warehouse), Priority, Shipment Method, Carrier.

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

To add shipping information to an order

1. Navigate to the quote or order with which you want to work.
2. Click the Shipping view tab.

   The account, site, contact, and address are automatically populated using the account information.
3. Change the default shipping information, if necessary, and fill in the Shipping form. Some fields are described in the following table.

Field	Comments
Shipping Method	Required. Select the method by which products must be shipped.
Shipment Terms	Required. Select the terms for shipment.
Carrier	Required. Select the carrier to be used for shipping.
Ship Complete	Select this check box to indicate that all items in the order must be shipped together.

Use the following procedure to specify different ship-to addresses for different line items in an order.

To specify different ship-to addresses for different line items in an order

1. Navigate to the quote or order with which you want to work.
2. Click the Shipping view tab.
3. Under the Shipping view tab, click the Line Items view tab.

   The Shipping applet is now a list of all the line items, instead of a form with just one shipping address for the order.
4. In the Shipping list, for any line item which you want to have a different shipping address, enter a name, address, shipping method and carrier.

Calculating Shipping Charges

Shipping charges depend on the shipping method and carrier used.

The end user must enter shipping information before calculating shipping charges. For more information, see *Adding Shipping Information*.

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

To calculate shipping charges

1. Navigate to the quote or order with which you want to work.
2. Click the Payments view tab.
3. In the Totals form, which is under the Payments form, click Calculate Shipping.

   The shipping charge is calculated and appears in the Totals form.

Calculating Taxes

If your Siebel Business Application has been integrated with third-party tax software from Taxware, then an end user can click the Calculate Taxes button to determine taxes and insert them in a quote or order. If your application is not integrated with third-party tax software, then an end user can enter a tax rate manually.

For products with components, taxes may be calculated either on the individual components, or on the extended price for the product with components. The method used is determined through the Product Administration screen.

Note: If you change items or prices in the quote or order, then be sure to recalculate the taxes when you are finished.

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

To calculate taxes by entering a tax rate manually

1. Navigate to the quote or order with which you want to work.
2. Click the Payments view tab.
3. In the Tax Rate field, if a rate is not yet assigned, then enter the appropriate tax rate.
4. If a customer is tax-exempt, then select the Tax Exempt Flag field and enter a reason and tax exempt certification ID.

Use the following procedure to calculate taxes using third-party tax software.

To calculate taxes using third-party tax software

1. Navigate to the quote or order with which you want to work.
2. Make sure the ship-to address is accurate.

   The ship-to address is used by third-party tax software when determining the tax rate. When line items in an order have different ship-to addresses, different tax rates may apply to the line items. For information about entering ship-to information, see *Adding Shipping Information*.
3. Click the Payments view tab.
4. In the Totals form, which is under the Payments form, click Calculate Taxes.

   The tax on the items appears in the Tax field on the Totals form.

Checking Availability to Promise (ATP)

Your Siebel Business Application can check with your back-office software to determine the availability of products in a quote or order, so that an end user can confirm availability for a customer. Using this feature, an end user can request delivery by a particular date, inquire about availability, and then reserve the items. An end user can also request that items be shipped from a particular source, such as a warehouse or other inventory location. If customers decide they do not need all of the items, then the end user can also unreserve items from the quote or order.

An end user can specify a requested date and source for the entire quote or order, or for individual line items. The end user can also request that certain items be shipped together, or that all items be shipped together.

Your Siebel Business Application must be set up to make special calls to your back-office application in order to use this functionality. For details, see *Integrating Order Management with Third-Party Product Availability Applications*

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

To check availability and reserve items in a quote or order

1. Navigate to the quote or order for which you want to check availability.
2. Click the Shipping view tab.
3. In the Shipping form, enter any requested information that applies for the entire quote or order. The following table shows some of the fields you might want to set.

Field	Comments
Requested Date	Enter the date you want the items to be delivered. When entered in the fulfillment form, this date is for the entire quote or order; it is copied onto each line item. Leave this field blank to request the earliest possible date.
Ship Complete	Check this field to indicate that the order needs to be shipped complete or not at all. All line items are assigned the number one in the Package field.
Source	Use this field to request that the items in the quote or order be shipped from a particular source. Leave this field blank to indicate that any source is okay. This source is copied to each line item.
Single Delivery	Select this checkbox to indicate that all line items in the order must be shipped from a single source and have a single delivery date.

4. Click the Fulfillment view tab under the Shipping view, and make any changes to the requested information for individual line items. The following table shows some fields you might want to change.

Field	Comments
Qty	Defaults to the quantity of the line item being ordered.

Field	Comments
Requested Date	Leave this field blank to request the earliest possible date.
Source	Use this field to request that information be shipped from a particular source. Leave this field blank to indicate that any source is okay.

5. To check availability for an individual item, select the line item in the Requested Schedule Lines list, which appears after the Fulfillment form, and then click Inquire.

   Your Siebel Business Application checks with your back-office application to determine availability for the line item, and the fulfillment engine returns information to the Promised Schedule Lines list.
6. To reserve individual items, select the items from the Line Items list, which appears after the Fulfillment form, and then click Reserve.

   Your Siebel Business Application sends a reservation request to your back-office application and the fulfillment engine returns an indication that the items are promised to the Promised Schedule Lines list.
7. To unreserve line items that you have previously reserved, in the Line Items list, select the items and click Unreserve.

Note: You can also reserve items without having previously inquired about them. When a line item is reserved, the requested quantity, date, and source are read-only. If you need to make a change to any of those settings, then you must unreserve the items first.

Splitting Line Items To Accommodate Different Requested Dates

A customer ordering a large quantity of the same item might want to request delivery of part of the order at two or more different times. In this case, the end user can split the line item, assigning quantities and requested dates to each new line. To split a line item, the end user enters two or more records in the Requested Schedule Lines and specifies the quantity and dates for each.

To split a line item across requested dates

1. In the Fulfillment list, select the line item that you want to split.
2. In the Requested Schedule Lines list, click New and enter the requested date and quantity for the first batch.
3. Add another record in the Requested Schedule Lines list for the second batch.
4. Continue to add records until you have the appropriate number of delivery requests. Make sure that the total quantity matches the quantity in the Line Items list.

Entering Payment Information for Quotes or Orders

End users enter billing and payment information in the Payments view tab of a quote or order.

Siebel Order Management provides these payment methods: credit card, purchase order, check, money wire, cash, and stored value. Your company may add other payment methods through list-of-values administration.

If you entered a payment profile for this customer, then when you make a selection in the Payment Method field, the Payment Detail form is automatically populated with the information that you entered in the profile for that payment method. The end user can edit this default information. For more information, see *Creating Payment Profiles*.

This task is part of the *Common Tasks for Creating Quotes and Sales Orders*.

To enter payment information

1. Navigate to the quote or order for which you want to enter payment information.
2. Click the Payments view tab.
3. If payment terms are appropriate for this quote or order, then in the Payment Terms field, select the payment terms.

   Your company's policies determine when payment terms must be used.
4. In the Payment Lines list, whick appears after the Payment form, add a payment line item.

   For the first payment line, the total amount of the order is copied into the Transaction Amount field.

   > **Note:** If this order was created from a quote that included payment information, then that information appears as a payment line item.
5. If the customer will use more than one payment method, then change the amount in the Transaction Amount field to the amount for this first payment method.
6. In the Payment Method field, select the method for the transaction amount.

   The default payment method is Credit Card.
7. In the Payment Detail form, which appears after the Payment Lines list, complete additional fields for the selected payment method.
8. The form changes depending on the payment method selected. For details about important fields for each payment type, see *Payment Fields for Payment Methods*.
9. For credit card and purchase order payments, check the customer's credit.
   - For a purchase order payment, click Credit Check.
   - For a credit card payment, click Authorize.

     > **Note:** For sales orders, you will typically use Authorize before placing the order. Then later, after the order has been filled, you may return and click Settle. When creating field service orders or cash and carry orders, you may want to use Authorize and Settle instead of Authorize at the time of order.
10. Review the Payment Status field and select a status, if appropriate.

    You may return to change this field later, for example, after receipt of a check. For details about changing the payment status, see *Changing the Status of a Payment*.
11. For each additional payment method, repeat Step 4 through Step 10. When you add another payment method, the transaction amount will default to the remaining amount not yet associated with a payment method.

Payment Fields for Payment Methods

Each payment method requires different information. When an end user adds a payment method to an order, the Payment Details form, which appears after the Payment Lines list, shows the appropriate fields for that payment method.

The available buttons on the form also change, to reflect the appropriate actions for that payment type. For example, a cash payment does not need to be authorized or settled, while a credit card payment does.

The tables in this topic list important payment details for each of the payment methods:

- Checks
- Credit Cards
- Purchase Orders
- Wire Transfers
- Stored Values

The following table lists important payment detail fields for checks.

Important Fields	Description
Transaction Amount	Amount for the individual payment transaction.
Payment Method	Check
Payment Type	Type of check being used: Personal, Corporate, or Cashiers.
Check Number	The identifying check number.
Date of Birth	The customer's date of birth.
Checking Account Number	The bank account number for the check.
ABA Code	The American Banking Association code for the bank on the check.
Social Security #	The customer's social security number.
Drivers License State	The state in which the customer has a driver's license.
Drivers License #	The customer's driver's license number.
Bill To Account Name and Address	Name and address of the account being billed for this payment line item. Defaults from the Payment form.
Authorization Code	Code indicating that authorization of the check has been given.
Authorization Date	Date authorization was given for the check. Can be entered manually or configured to be entered automatically.
Check Mailing Address	Your company's mailing address, to tell the customer where to mail the check.
Payment Status	Defaults to Pending Receipt of Check Receipt. Change this value after the check has been received and deposited, and funds received.

The following table lists important payment detail fields for credit cards.

Important Fields	Description
Transaction Amount	Amount for the individual payment transaction.
Payment Method	Credit Card
Payment Type	The type of credit card. Out-of-the-box, available types are Visa, MasterCard, Discover, and American Express.
Credit Card #	The credit card number.
Expiration Month	The month the credit card expires.
Expiration Year	The year the credit card expires.
Issuing Bank	The name of the bank that issued the card.
Issuing Bank Phone Number	The phone number for the bank that issued the card.
Credit Card Holder	Name of the person on the credit card.
Billing Address, City, State, Zip	Address fields for the person responsible for this payment line item. Defaults to the account's bill-to address.
Payment Status	After you click Authorize or Authorize and Settle, the credit card authorization process returns one of the following values as the status: • **Authorized.** Indicates the credit card payment for the transaction amount entered has been approved. • **AVS Failed.** Indicates that the address entered for the customer did not match the address in the Address Verification System (AVS). • **Card Verification Declined.** Indicates this payment was declined because the card could not be verified. • **Charged.** Indicates the charge was authorized and has been charged. • **Declined.** Indicates authorization was declined. • **Need Verbal Authorization.** Indicates that the bank must call the owner of the card for authorization. • **Referral.** Neither approved nor declined. The bank needs to research something related to authorizing this payment.
Authorization Code	Code returned when the transaction was authorized.
Authorization Date	Date the transaction was authorized.

Important Fields	Description
AVS Code	Code provided by the Address Verification System (AVS) to indicate the address for the credit card was verified.
Customer Code	For a corporate customer, a code verifying that the credit card may be used.
Card Verification #	This field contains a four digit unique identifier that is often printed on the back of a credit card. It is used to reduce fraud because only the person holding the card must know it. This field must be deleted after the number is submitted for authorization.
Merchant ID	Your company's identifying number. If more than one Merchant ID is available, then the correct ID for this payment can be selected from list.

The following table lists important payment detail fields for purchase orders.

Important Fields	Description
Transaction Amount	Amount for the individual payment transaction.
Payment Method	Purchase Order
Purchase Order #	Number of the purchase order. May display Pending if the order was created from a quote which had Pending as the PO#.
Payment Type	For a purchase order, the payment type is Payment.
Payment Status	Status of the payment. Defaults to Pending Check Receipt. Change this value after the funds have been received.
Credit Status	Status information returned from the back office after a credit check is done by clicking Check Credit.
Credit Status As Of	Date the credit check status information was returned.
Credit Check Message	Message text returned with the credit status.
Bill To Account	Name of the account responsible for this payment line item.
Bill To Address	Address of the account to be used in relation to this payment line item.
Approval Date	Date that the purchase order payment was approved.
Approved By	The name of the person who authorized the purchase order.

Important Fields	Description
Authorization Code	Code returned when the payment was authorized.
Authorization Date	Date the payment was authorized. Can be the date the funds appear in the account.
Payment Comments	Any comments appropriate for this payment.

The following table lists important payment detail fields for wire transfers.

Important Fields	Description
Transaction Amount	Amount for the individual payment transaction.
Payment Method	Wire Transfer
Originating Bank	The bank sending the transfer.
Branch Bank Address, City, State, Zip	The address of the bank sending the transfer of funds.
Bank Routing #	Routing number for the bank.
Payment Account #	Wire transfer account number.
Date of birth	The customer's date of birth.
Bill To Account Name	Name of the account responsible for this payment line item.
Contact Name	Person to contact in relation to this payment line item.
Contact Phone	Phone number for the contact.
ABA Code	The American Banking Association code for the bank on the check.
Social Security #	The customer's social security number.
Authorization Code	Code returned when the payment was authorized.
Authorization Date	Date the payment was authorized. May be the date you see the funds in the account.
Payment Status	Status of the payment. Defaults to Pending Check Receipt.
Drivers License #	The customer's driver's license number.

Important Fields	Description
Drivers License State	The state in which the customer is licensed to drive.

The following table lists important payment detail fields for stored values.

Important Fields	Description
Transaction Amount	Amount for the individual payment transaction.
Payment Method	Stored Value
Payment Type	Type of stored value. Preconfigured values are Gift Certificate and In-Store Credit
Payment Status	Status of the payment. Defaults to Pending Check Receipt.
Contact Name	Contact name associated with the stored value.
Payment Account #	Number of the account that will provide the payment of the stored value.
Account Name	Name of the account with which the payment is associated. Defaults to the bill-to account.
Authorization Code	Code returned when the payment was authorized.
Authorization Date	Date the payment was authorized. May be the date you see the funds in the account.

Approving a Payment

Depending on how your company has set up payments, certain payment methods might require approvals before they can be accepted.

The initial payment status is set to pending. A user with approval capabilities will then need to change the status to approved.

Note: Approving a payment is not the same as authorizing a payment. Authorization performs a credit check through a back-office application.

To approve a payment

1. Navigate to the quote or order for which payment needs to be approved.
2. Click the Payment view tab.
3. In the Payment Lines list, select the purchase order payment to approve.
4. In the Payment Detail form, which appears after the Payment Lines list, enter your name in the Approved By field and enter today's date in the Approval Date field.

Changing the Status of a Payment

The status assigned to a payment may be set automatically in some cases (such as when a credit card is authorized), but needs to be set manually in others. For example, when the payment method is a check, the initial status is set to Pending Check Receipt. After a company receives funds for the check, an end user can manually change the status of the payment transaction to Authorized, for example.

Your company may also create business processes that automatically update the status under certain circumstances.

The following table lists the possible payment statuses and explains when they are likely to be used.

Status	Comment
New	Payment information was sent, but confirmation of receipt from the back-office application was not received. The message may not have been received.
Authorized	The payment has been authorized.
Declined	The payment has been declined.
Charged	The payment has been charged to a credit card.
Refunded	The payment has been refunded to the customer.
Reversed	The payment shown was previously charged to a credit card, and that charge has been reversed; a credit for the same amount was sent to the credit card.
Referral	Payment has been held by the bank until the bank can gather additional information.
Pending Check Receipt	A check, purchase order, or wire transfer has been received. Receipt of the payment has not yet occurred.
AVS Failed	Indicates that the address entered for the customer did not match the address in the Address Verification System (AVS). Typically used with credit card payments
Card Verification Declined	A third-party authorization application was not able to verify the credit card.
Need Verbal Authorization	This payment needs to be verbally authorized by the credit card company.

To change the status of a payment

1. Navigate to the quote or order for which the payment status needs to be changed.
2. Click the Payment view tab.
3. In the Payment Lines list, select the appropriate status in the Payment Status field.

Checking Credit for a Purchase Order

An end user can check a customer's credit when the payment method is a purchase order.

The end user can also check credit when payment is made using a credit card. For information, see *Authorizing and Settling Credit Card Payments*.

To check credit for a purchase order

1. Navigate to the quote or order which you are working.
2. Click the Payment view tab.
3. If necessary, enter the purchase order information.
4. In the Payment form, click Credit Check.

   Your Siebel Business Application sends an authorization request to your back office or financial application to confirm whether the purchase order is valid and the payment can be authorized.

   The back-office application returns information to three fields: Credit Status, Credit Status As Of, and Credit Check Message.

Authorizing and Settling Credit Card Payments

In an order, an end user can check authorization of payment by credit card and can then settle the payment, to collect the amount authorized.

For credit card authorization to be available, your Siebel Business Application must have implemented integration with another payment processor. For more information, see *Siebel eSales Administration Guide* .

To authorize and settle a credit card payment

1. Navigate to the quote or order with which you are working.
2. Click the Payments view tab.
3. If necessary, add a Payment Line and enter the credit card payment information, or select the payment line containing the credit card payment.
4. Perform one of the following:
   - To authorize the credit card payment, in the Payment Detail - Credit Card form, click Authorize.
   - To settle a payment that has previously been authorized, click Settle.
   - To authorize and settle a credit card payment at the same time, click Authorize and Settle.

   Your Siebel Business Application sends an authorization request to a third-party payment processing application to confirm whether the payment can be authorized, and settled, if appropriate.

> **Note:** When the end user clicks Authorize or Authorize and Settle, credit check calls a payment processing application to approve the transaction. Sometimes an error is returned, or the payment is rejected, and the end user can use the status and any error message returned to determine the appropriate course of action.

Authorizing a Credit Card Payment Manually

When the credit check of a credit card payment returns the Payment status Declined or Referred, the end user will need to assess the reason the payment has not been approved. If the end user determines that the payment is good, then he or she can manually authorize the payment. In such a case, the end user must obtain an authorization code verbally from the credit card company.

To manually authorize a credit card payment

1. Navigate to the order with which you want to work.

2. Click the Payments view tab and select the credit card payment line with the Payment Status of Declined or Referred.
3. In the Authorization Code field, enter the authorization code.
4. When you are prompted to confirm the change to the order, click Yes.

   Your Siebel Business Application accepts the new entry and the credit status on the line item is changed to Authorized. A note appears in the Payment History with the change request date and requestor. This manually entered code is used to validate the order approval with the bank during settlement processes.

Checking Payment History

It is important to be able to explain to customers all of the transactions that relate to their payment method. An end user can see all transactions against a specific payment line by checking the payment history. For example, if a credit card is authorized manually, then that action is noted in the history, along with the date and requestor.

The specific changes that are recorded in the payment history are determined by settings your company makes in the Application Administration screen, Audit Trail view. For complete information about setting up Audit Trail, see *Siebel Applications Administration Guide* .

To check payment history

1. Navigate to the order for which you want to check payment history.
2. Click the Payments view tab.
3. In the Payment Lines list, which appears after the Payments form, select the payment for which you want to see history information.
4. From the menu of the Payment Lines list, select Payment History.

   The Payment History dialog box appears, listing information about changes to the payment item.

Entering Refunds

An end user can enter a refund of a credit card payment for a customer, when appropriate. Refunds are given after a settlement of a payment has occurred and funds have been transferred.

To enter a refund

1. Navigate to the order for which you want to enter a refund.
2. Click the Payments view tab.
3. In the Payment Lines list, which appears after the Payments form, select the credit card payment for which you want to enter a refund, and click Refund.

Changing the Default Price List and Rate List

An end user can change the default price list that is used for both quotes and orders.

To set the default price list

1. Navigate to the User Preferences screen, and then the Price List Sales Methodology.
2. In the Price List field, select the price list to be used as a default.
3. In the Rate List field, select the rate list to be used as a default.
4. If needed, select a sales methodology from the Sales Methodology list.

How End Users Work with Product Promotions

Product promotions can be displayed in employee applications in product catalogs and in the list in the Promotions field in the line items of quotes and orders. For information about how to display promotions in these places, see the topic about promotions in *Siebel Pricing Administration Guide* .

Displaying the promotion in these places gives the end user many options for working with product promotions.

In every case, the price discount that the customer gets because of the promotion is clearly visible on the quote or order.

The following topics describe how end users work with product promotions:

- *About Selecting Product Promotions from a Catalog*
- *About Using Coupon Product Promotions*
- *About Using Bundle Product Promotions*
- *About Product Promotion Recommendations*
- *Editing Bundle Product Promotions*
- *Process of Enabling the Automatic Configuration of Customizable Products*

About Selecting Product Promotions from a Catalog

If promotions are displayed in product catalogs, then they are visible in the Promotions applet of catalogs that contain the products they apply to. The expiration date for each promotion is also displayed to encourage customers to buy the product before the promotion expires.

Sales representatives using the catalog can click Add Promotion to add the promotion to a quote or order and apply the promotional discount. The products associated with the promotion are automatically added to the quote or order.

Sales representatives using the catalog can click Edit Promotion if they want to modify the selections in a promotion. For example, if the promotion requires the customer to purchase one product from the Television product line, then the sales representative clicks Edit Promotion to navigate to a view where the television can be selected.

About Using Coupon Product Promotions

If you display coupon promotions on the line items of quotes and orders, then they are available in the list in the Promotion field of any line item where you add a product that qualifies for the coupon. The list displays the names of all the promotions that could apply to this product, and the sales representative applies the promotion by choosing from this list.

Two possible uses of this feature are:

- **Customer has coupon.** A customer buys a product, and the sales representative adds it to an order as a line item. The customer presents the sales representative with a promotional coupon for that product, and the sales representative chooses that coupon in the Promotion field of the line item to apply the promotional discount to the product.
- **Sales representative suggests coupon.** A customer adds several products as line items to a quote, but is uncertain about whether to buy the products because of the high total price of the quote. The sales

representative uses the Promotion fields of the line items to see which coupon promotions could apply to these products, and offers the customer a promotion to convince the customer to make the purchase.

About Using Bundle Product Promotions

Sales representatives can use bundle promotions displayed in catalogs for the top-down (highest to lowest) selection of promotions, and they can use promotions displayed in the line items of quotes and orders for the bottom-up (lowest to highest) selection of bundle promotions.

Two possible uses of this feature are:

- **Top-down selection of bundle promotion.** A customer wants to buy a specific bundle promotion that has been advertised. The sales representative chooses the promotion from the catalog, and all the product items are added to the order with the promotional discount.
- **Bottom-up selection of bundle promotion.** A customer is placing an order, and the sales representative adds several products as line items to a quote. The recommendation engine displays a message saying that there is a promotion available that includes these products. The customer decides whether to accept this promotion.

About Product Promotion Recommendations

To display product promotion recommendations, the sales representative selects Verify Promotion from the applet-level menu of a Quote or Order header.

A pop-up list displays all the recommended promotions and allows the user to accept or reject each promotion recommendation. The order in which promotions are listed is based on the ranking entered in the Score field of the Promotion record and the percentage of current Quote or Order line items matching the promotion definition.

Promotions are recommended only if the administrator selects the Recommendable flag on the Product Promotions screen, Products view while creating the promotion.

Editing Bundle Product Promotions

After an end user has added a bundle product promotion to a quote or order, the user can edit the product promotion to change the products it includes or the quantities of these products. Then the user can validate the edited promotion to make sure that the new products and quantities still qualify for the promotion.

To edit a bundle product promotion

1. Add the Bundle as a line item of a Sales Order.
2. One record is added representing the bundle promotion and others records are added representing the products in the promotion.
3. In the Line Items list, select the promotion record
4. From Menu, select Edit Promotion.
5. In the Edit Promotion form:
    a. Edit the Quantity field for products to change the products' quantity.
    b. Click the Delete icon next to a product to delete that product.

c. Click the Configure icon next to a product to configure that product.

6. In the Edit Promotion form, after you are finished making edits, Click Validate.

   An error message appears if the quantities or products as edited qualify for the promotion. If so, correct the quantities or products.

7. In the Edit Promotion form, click Save
8. The edited information for the bundle product promotion is added to the Line Items list. In the indented line items under the promotion line item record, quantities are changes, new products are added, and former products are deleted, as you specified in the Edit Promotion form.

Process of Enabling the Automatic Configuration of Customizable Products

When you enable the automatic configuration of customizable products that are based on a promotion definition, you enable the automatic configuration of the following relationships:

- Promotion bundle and product class relationship
- Promotion bundle and product relationship

You also populate the default settings of the promotion on the line item. You can add a promotion line item to a quote or order in any of the following ways:

- By changing the promotion associated with an existing line item either by selecting the promotion from the promotion picker on an existing line item, or by adding a new promotion line item so that it is associated with an existing line item.
- By accepting a bottom-up (lowest to highest) recommendation for a product. This option is available when the user clicks Verify or adds products from the Catalog view.
- By upgrading an existing promotion.

> **Note:** In previous versions of Siebel Order Management (earlier than version 8.0.x), when you added a promotion line item to a quote, the *existing* line items that got associated with the new promotion line item were not automatically configured with the default settings of the new promotion. The default settings of the promotion were populated only when they enter the Siebel eConfigurator session. In addition, the default settings of the promotion were not populated on the existing relationships of the customizable product when they enter the Siebel eConfigurator session.

To enable the automatic configuration of customizable products that are based on a promotion definition, perform the following tasks:

1. *System Preference for enabling automatic configuration of Customizable Products*

See the following topics to learn more about the automatic configuration of customizable products:

- *Workflows to Activate for Automatic Configuration of Customizable Products*
- *Headless Configuration Service*
- *ISS Promotion Management Service*
- *Adding a Promotion Line Item*

System Preference for enabling automatic configuration of Customizable Products

To configure the Promotion Automatch system preference

1. Navigate to the Administration - Application screen, and then the System Preferences view.
2. Add a new system preference where name is Promotion Automatch and System Preference Value is Y.

Workflows to Activate for Automatic Configuration of Customizable Products

You must activate the following workflows to support the automatic configuration of customizable products that are based on a promotion definition.

- *ISS Promotion Upgrade Process*
- *Headless Configuration Web Service*
- *SIS OM Get Future Asset Subprocess*
- *SIS OM Sync Delta Subprocess*

For more information about deploying and activating workflows, see *Siebel Business Process Framework: Workflow Guide* .

Headless Configuration Service

The Headless Configuration business service is required to enable the automatic configuration of customizable products that are based on a promotion definition. It works with or without the user interface. It contains the following public methods

- *ApplyPromotionDefaults Method*
- *Restore Document Method*
- *Save Document Method*

The fourth table in this topic describes the Headless Configuration service user properties that you must set.

ApplyPromotionDefaults Method

This method is automatically triggered by the Siebel Business Application when you apply a promotion, provided the Promotion Automatch system preference is set to Y. This method is not triggered if the Promotion Automatch system preference is set to N, or if the Promotion Automatch system preference is not defined.

This method configures customizable product line items in the current product as follows:

1. It loads the product line item.
2. Runs batch validation on the product line item.

3. Changes the product line item according to the Promotion Automatch argument value.

   The following table describesSiebel CRM behavior when the Promotion Automatch argument value is set to Default Cardinality, and when the Promotion Automatch argument value is set to Min Max Cardinality. For more information about cardinality, see the table in the following section.
4. It writes the line item back to the document

This method does not require a user interface context to be called, and contains the method arguments listed in the following table.

In asset-based ordering (ABO) mode, ApplyPromotionDefaults calls the SIS OM Get Future Asset subprocess workflow, which computes the action codes.

About Cardinality

Cardinality refers to the quantity of the component that a user can select. *Minimum cardinality* governs whether or not selecting items from a relationship is optional or required. *Maximum cardinality* sets the maximum number of items that a user can select from a relationship. *Default cardinality* specifies what quantity of the default product is automatically added to the initial solution that the user sees. For more information about cardinality, see Siebel Product Administration Guide.

Argument Name	Data Type	Input or Output	Comments
Promotion Automatch	String	Input	If set to *Default* Cardinality, then Siebel CRM behavior is as follows: • Non-existing items change to the default quantity. • Existing ports lower than the minimum quantity change to the default quantity. • Existing ports higher than the maximum quantity change to the default quantity. • Existing product ports not violating the cardinality remain the same. • Existing class ports lower than the default quantity change to the default quantity. If set to *Min Max Cardinality*, then Siebel CRM behavior is as follows: • Non-existing items change to the default quantity. • Existing ports lower than the minimum quantity change to the min quantity. • Existing ports higher than the maximum quantity change to the max quantity. • Existing product ports not violating the cardinality remain the same. **Note:** This behavior can occur if neither Min nor Max cardinality on the promotion is specified. • Existing class ports lower than the default quantity change to the default quantity. If set to blank (if no value is set), then Siebel CRM behavior reverts to the existing behavior (that is, the behavior in previous versions of Siebel Order Management). The default setting is blank.
Log Error	String	Input	If set to Y, then logs error messages.

Argument Name	Data Type	Input or Output	Comments
			The default is Y.
Display Error	String	Input	If set to Y, then displays Unified Messaging Framework (UMF) messages; a user interface context is required. If, depending on application of the promotion, the customizable product structure changes, then the UMF message displays the recommendations applet in the dynamic catalog view; or, the user can open the recommendations applet using the recommendation icon. The default is N.
Document Id	String	Input	The Header Id of the quote or order.
Document Type	String	Input	This value must be either Quote or Order.
Integration Object	String	Input	Depending on document type and ABO mode, this argument defaults to either one of the following: • SIS OM Order, or SIS OM Quote • When ABO is turned off: 7.7 Order Entry Integration Object, or 7.7 Quote Integration Object
Search Spec	String	Input	The Search Spec is applied to the business component line item. By default, this argument is empty. However when ApplyPromotionDefaults is called automatically on applying a promotion (by setting the Promotion Automatch system preference to Y), this argument is set by the application to retain only those line items that are associated with that promotion instance. See also the following related Search Spec user properties in the third table in this topic: • Order Default Search Spec • Quote Default Search Spec
Sync	String	Input	If set to Y, then the result is synchronized to the database. The default is N.

Restore Document Method

This method restores the document to its saved state, and contains the method arguments listed in the following table. If an error is encountered during execution of the ApplyPromotionDefaults method, then Restore Document is automatically triggered by the application (provided the Promotion Automatch system preference is set to Y). Restore Document is not triggered if the Promotion Automatch system preference is set to N, or if the Promotion Automatch system preference is not defined.

Argument Name	Data Type	Input or Output	Comments
Document Id	String	Input	The Header Id of the quote or order.
Document Type	String	Input	Document Type must be either Quote or Order.

Save Document Method

This method saves the document in its current state, and contains the method arguments listed in the following table. This method is automatically triggered by the application to save the state of the document before applying the promotion, provided that the Promotion Automatch system preference is set to Y. This method is not triggered if the Promotion Automatch system preference is set to N, or if the Promotion Automatch system preference is not defined.

Argument Name	Data Type	Input or Output	Comments
Document Id	String	Input	The Header Id of the quote or order.
Document Type	String	Input	Document Type must be either Quote or Order.

Headless Configuration Service User Properties

Use the Headless Configuration service user properties to set the values given in the following table.

Business Service User Property	Value
Logging Level	3 **Note:** This value is the threshold log level for error messages.
Order Business Object	Order Entry (Sales)
Order Default Search Spec	`[IsComplexProductRoot]='Y' AND [Action Code] <> LookupValue('DELTA_ACTION_CODE','Deleted')` **Note:** The search specification is appended to the input argument Search Spec.
Order Due Date Field	Requested Ship Date
Order Header Business Component	Order Entry - Orders
Order Line Item Business Component	Order Entry - Line Items
Quote Business Object	Quote

Business Service User Property	Value
Quote Default Search Spec	`[IsComplexProductRoot]='Y' AND [Action Code] <> LookUpValue('DELTA_ACTION_CODE','Deleted')` **Note:** The search specification is appended to the input argument Search Spec.
Quote Due Date Field	Due Date
Quote Header Business Component	Quote
Quote Line Item Business Component	Quote Item
UMF Headless Configuration Messages Source	Headless Configuration
UMF Headless Configuration Error Messages Score	100 **Note:** The UMF message is displayed whenever an apply promotion is undone.
UMF Headless Configuration Error Messages Type	Headless Configuration Error Message **Note:** The UMF message is displayed whenever an apply promotion is undone.
UMF Headless Configuration Messages Source	Headless Configuration Proceed Message **Note:** The UMF message is displayed when changes are made to the customizable product using ApplyPromotionDefaults.
UMF Headless Configuration Messages Score	30 **Note:** The UMF message is displayed when changes are made to the customizable product using ApplyPromotionDefaults.
UMF Headless Configuration Messages Type	Headless Configuration Message

ISS Promotion Management Service

This business service is required to enable the automatic configuration of customizable products that are based on a promotion definition. The ISS Promotion Management Service is used to handle product promotions at run time, such as Apply Promotion, Recommend Promotion, and so on. This business service also provides the functions to support

Unified Messaging and Asset-Based Ordering, such as Load Message and Load Promotion-Related Assets. For more information about the ISS Promotion Management service, see *Siebel Order Management Infrastructure Guide* .

ISS Promotion Management Service User Properties

Use the user properties of the ISS Promotion Management Service to set the values given in the following table.

User Property	Value	Description
Promotion Automatch	Default Cardinality	Specifies Promotion Automatch as input to the Headless Configuration Service business service.
Promotion Automatch Display Error	Y	Specifies Display Error as input to the Headless Configuration Service business service.
Promotion Automatch Log Error	Y	Specifies Log Error as input to the Headless Configuration Service business service.

Adding a Promotion Line Item

The behavior of Siebel Order Management when it adds a promotion line item to a quote or an order when the Siebel CRM system preference Promotion Automatch is set to Y is different from when Promotion Automatch is set to N:

- When Promotion Automatch is set to N (that is, when the automatic configuration of customizable products is not enabled), Siebel CRM behavior is as follows:
  - The existing customizable product line items that are associated with a promotion are not added with the default configuration when you add a promotion line item to a quote or an order.
  - The default settings of the promotion are populated only by manually using Siebel Configurator to add them.

    > **Note:** You must click the Customize button to start Siebel Configurator, where you can manually add the default settings of a promotion.

  - The promotion-specific default settings are not processed on the existing product lines.
- When Promotion Automatch is set to Y (that is, when the automatic configuration of customizable products is enabled), Siebel CRM behavior is as follows:

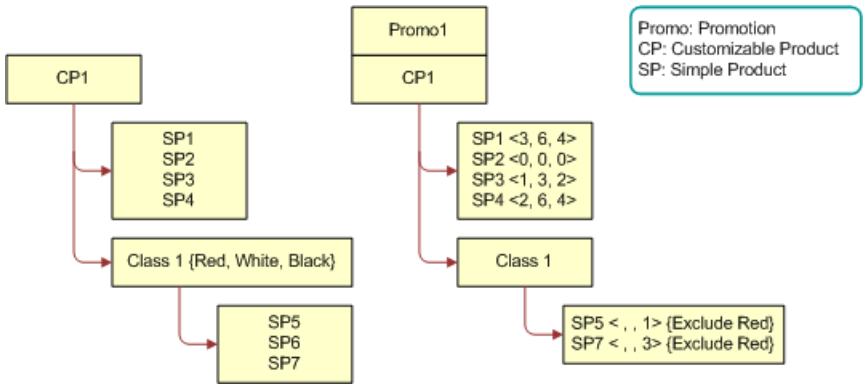
When you add a promotion line item to a quote or an order, the existing root line items that are associated with the promotion are modified according to the Promotion Automatch user property of the ISS Promotion Management Service. For more information about the ISS Promotion Management Service, see *ISS Promotion Management Service*.

> **Note:** This behavior applies whether modifying, upgrading, or recommending a promotion. If a conflict occurs when adding a promotion line item to a quote, an order, or a customizable product, then Siebel CRM automatically reverts to the previous product definitions.

Example: Adding a Promotion Line Item

This example shows the expected output when you apply a promotion to a customizable product that is based on the Promotion Automatch argument value, which can be set to either Default Cardinality or Min Max Cardinality. In this example:

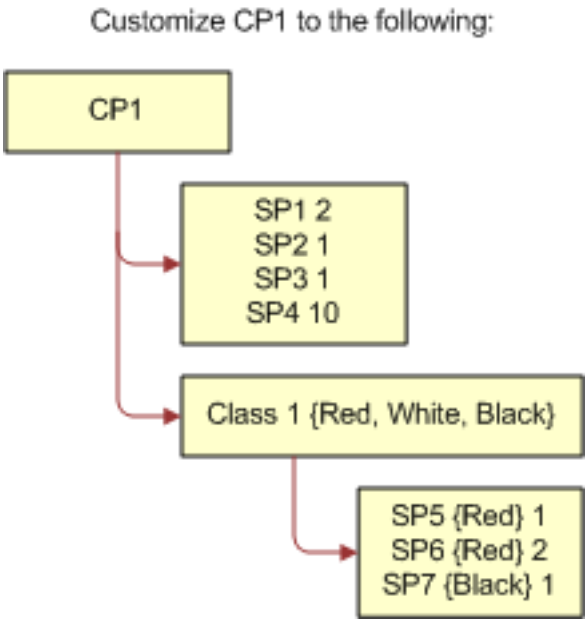
1. A customizable product named CP1 is defined and a promotion named Promo1 is defined as shown in the following figure.
2. CP1 is then customized as shown in the second figure in this topic.
3. The expected output when you apply a promotion (Promo1) to a customizable product (CP1) that is based on the Promotion Automatch argument value is shown in the third figure in this topic.



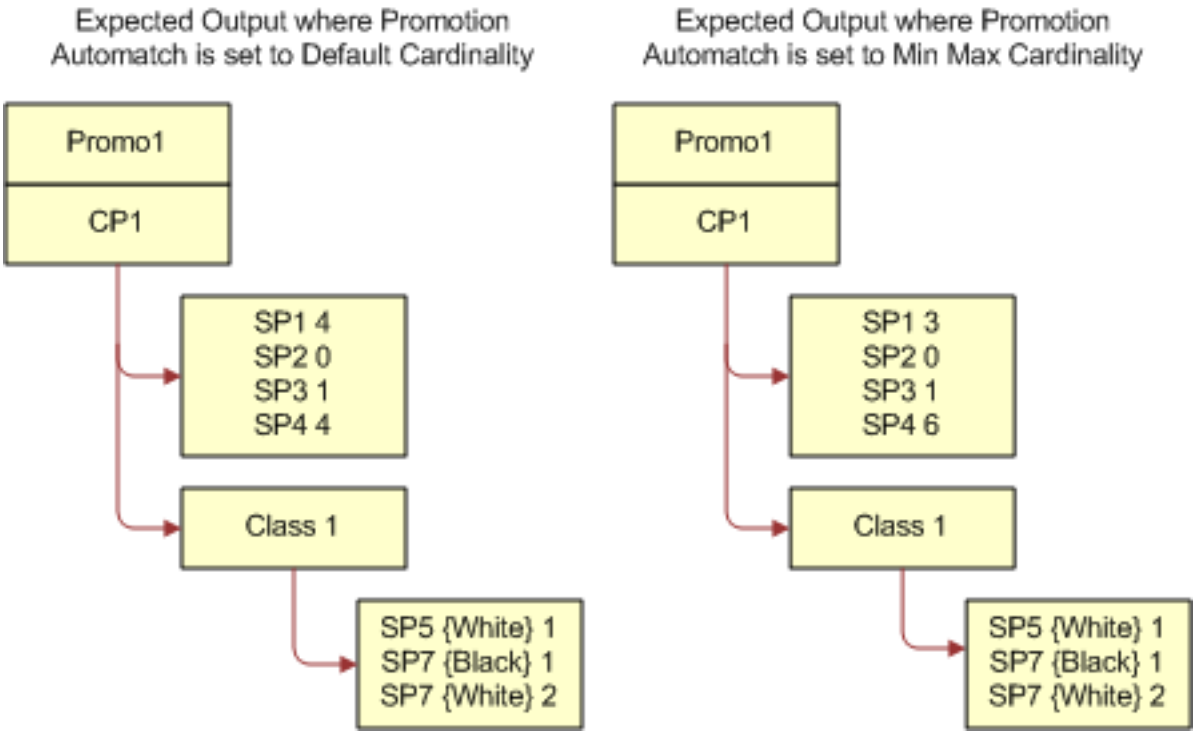
The first figure in this topic shows a product named CP1 with the following structure: CP1 has four simple product relationships defined (SP1, SP2, SP3, SP4) and a Product Class named Class 1. A further three simple product relationships (SP5, SP6, SP7) are defined and associated with Class 1. Class 1 is associated with an attribute that has the values {Red, White, Black}.

> **Note:** The relationships SP1 to SP4 have Domain Type set to Product. The Class 1 relationship has Domain Type set to Class.

The first figure in this topic shows a promotion named Promo1, where CP1 has the following structure: CP1 has four simple product relationships defined (SP1, SP2, SP3, SP4) and a Product Class named Class 1. A further two simple product relationships (SP5, SP7) are defined and associated with Class 1. For SP1, Min is set to 3, Max is set to 6, and Default is set to 4. For SP2, Min is set to 0, Max is set to 0, and Default is set to 0. For SP3, Min is set to 1, Max is set to 3, and Default is set to 2. For SP4, Min is set to 2, Max is set to 6, and Default is set to 4. For SP5, Default is set to 1, and the attribute value Red is excluded for this relationship. For SP7, Default is set to 3, and the attribute value Red is excluded for this relationship.



The second figure in this topic shows the customized CP1 with the following structure: CP1 has four simple product relationships defined (SP1, SP2, SP3, SP4) and a Product Class named Class 1. A further three simple product relationships (SP5, SP6, SP7) are defined and associated with Class 1. Class 1 is associated with an attribute with the values {Red, White, Black}. SP1 has a default quantity of 2, SP2 has a default quantity of 1, SP3 has a default quantity of 1, and SP4 has a default quantity of 10. SP5 has a default quantity of 1 and the attribute value Red. SP6 has a default quantity of 2 and the attribute value Red. SP7 has a default quantity of 1 and the attribute Black.



The third figure in this topic shows the expected output when Promotion Automatch is set to Default Cardinality, which is as follows: SP1 has a default quantity of 4, SP2 has a default quantity of 0, SP3 has a default quantity of 1, and SP4 has a default quantity of 4. SP5 has a default quantity of 1 and the attribute value White. SP7 has a default quantity of 1 and the attribute value Black. SP7 has a default quantity of 2 and the attribute value White.

The third figure in this topic shows the expected output when Promotion Automatch is set to Min Max Cardinality, which is as follows: SP1 has a default quantity of 3, SP2 has a default quantity of 0, SP3 has a default quantity of 1, and SP4 has a default quantity of 6. SP5 has a default quantity of 1 and the attribute value White. SP7 has a default quantity of 1 and the attribute value Black. SP7 has a default quantity of 1 and the attribute value White.

To add a promotion line item to a customizable product that is based on the Promotion Automatch argument value

1. Create the product structure for CP1 and Promo1 as shown the first figure in this topic.
2. Add CP1 to a quote or an order line item, and then customize CP1 as shown in the second figure in this topic.
3. Choose Promo1 for CP1 using the promotion picker.

   The third figure in this topic shows the expected output when Promotion Automatch is set to Default Cardinality, and when Promotion Automatch is set to Min Max Cardinality.

For more information about how Siebel CRM behaves depending on the Promotion Automatch argument value, see the first table in this topic.

Handling Attribute Defaults

Attribute default values are handled as follows when applying a promotion to a customizable product line item. Attribute default values are handled as follows whether modifying, upgrading, or recommending a promotion:

- If the current attribute value of an existing component is allowed by the promotion definition (because it is either listed in an *Include* rule, or omitted from an *Exclude* rule), then the attribute value is retained.
- If the current attribute value of an existing component is not allowed by the promotion definition, or if the component was added (for example, by applying default values), in which case the current attribute value is nonexistent, then:
  - If an Include rule is defined, then the current attribute value is updated with the default value.

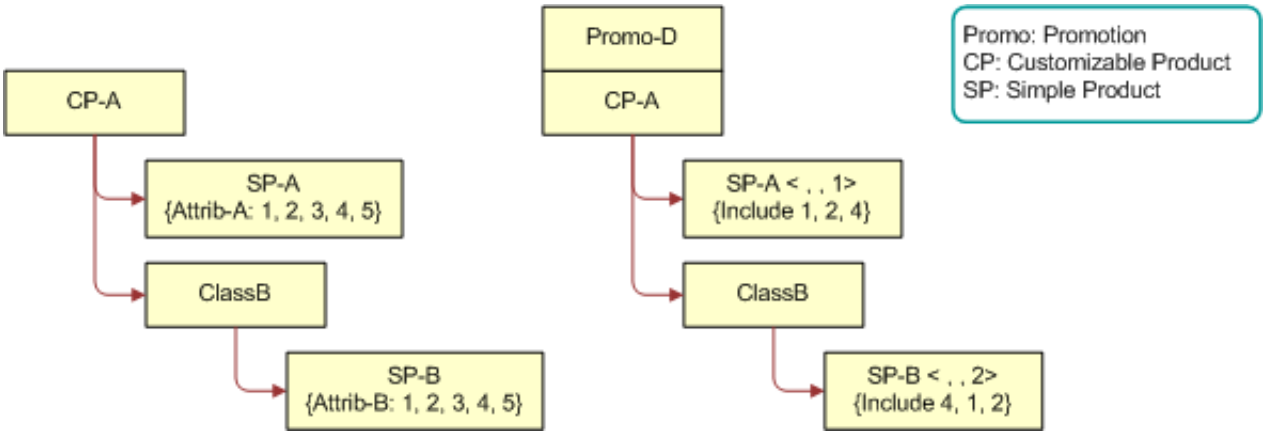
    This default value is controlled by the Sort Specification on the business component ISS Promotion Item Attribute Value. If the Sort Specification is set to Created, then the default value is the first created attribute value for the promotion in the Promotion Item Attribute Value List Applet.
  - If an Exclude rule is defined, then the default value cannot be specified in the rule. The current attribute value is initialized according to the existing behavior.

Example: Handling Attribute Defaults

The following example shows the expected output for attribute defaults when you apply a promotion to a customizable product. In this example, a customizable product named CP-A is defined and a promotion named Promo-D is defined. In this example:

1. A customizable product named CP-A is defined and a promotion named Promo-D is defined as shown in the following figure.
2. CP-A is then customized as shown in the second figure in this topic.
3. The expected output for attribute defaults when you apply a promotion (Promo-D) to a customizable product (CP-A) that is based on the Promotion Automatch argument value is shown in the third figure in this topic.

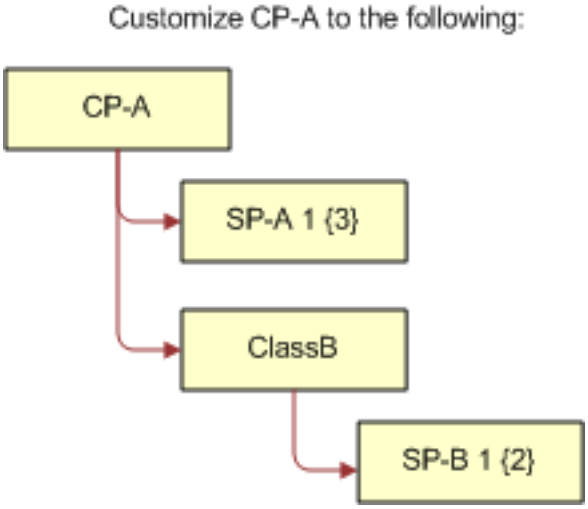
> **Note:** The expected output for attribute defaults when you apply a promotion (Promo-D) to a customizable product (CP-A) is the same whether Promotion Automatch is set to Default Cardinality or Min Max Cardinality.



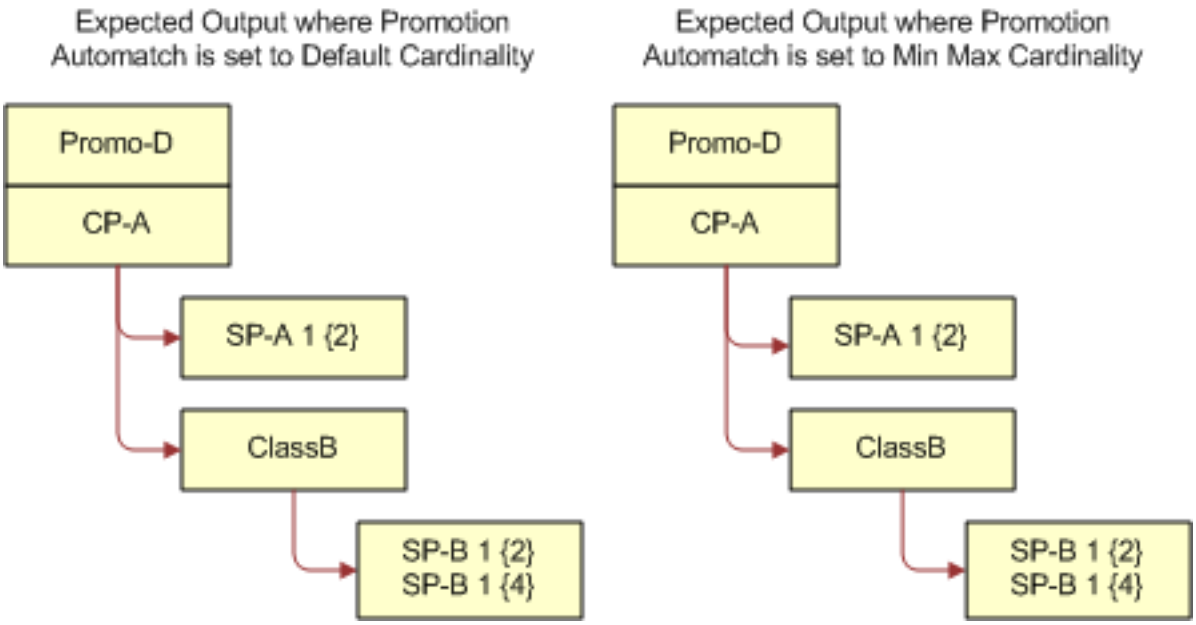
The first figure in this topic shows a product named CP-A with the following structure: CP-A has a simple product relationship named SP-A defined and a Product Class named ClassB. Another simple product relationship named SP-B is defined and associated with ClassB. SP-A is associated with an attribute, Attrib-A, which has the values {1,2,3,4,5}. SP-B is associated with an attribute, Attrib-B, which has the values {1,2,3,4,5}.

Note: The relationships SP-A and SP-B have Domain Type set to Product. The ClassB relationship has Domain Type set to Class.

The first figure in this topic shows a promotion named Promo-D, where CP-A has the following structure: CP-A has a simple product relationship named SP-A defined and a Product Class named ClassB. Another simple product relationship named SP-B is defined and associated with ClassB. For SP-A, Default is set to 1 and the attribute values 2 and 4 (of Attrib-A) are included for this relationship. For SP-B, Default is set to 2 and the attribute values 4, 1, and 2 (of Attrib-B) are included for this relationship.



The second figure in this topic shows the customized CP-A with the following structure: CP-A has a simple product relationship named SP-A defined and a Product Class named ClassB. Another simple product relationship named SP-B is defined and associated with ClassB. SP-A has a default quantity of 1 and the attribute value 3. SP-B has a default quantity of 1 and the attribute value 2.



The third figure in this topic shows the expected output when Promotion Automatch is set to Default Cardinality, which is as follows: SP-A has a default quantity of 1 and the attribute value 2. SP-B has a default quantity of 1 and the attribute value 2. SP-B has a default quantity of 1 and the attribute value 4.

The third figure in this topic shows the expected output when Promotion Automatch is set to Min Max Cardinality, which is as follows: SP-A has a default quantity of 1 and the attribute value 2. SP-B has a default quantity of 1 and the attribute value 2. SP-B has a default quantity of 1 and the attribute value 4.

In both cases (that is, when Promotion Automatch is set to Default Cardinality and Min Max Cardinality), the expected output is the same. Note the following:

- The attribute value of 3 on SP-A is changed to 2 , because 3 was not specified in the include list in Promo-D for SP-A.
- The existing SP-B remains unchanged, because its attribute value of 2 is permitted in Promo-D, although it is not the first in the list.
- The added SP-B has an attribute value of 4.

To apply a promotion to a customizable product

1. Create the product structure for CP-A and Promo-D as shown in .
2. Add CP-A to a quote or order line item, and then customize CP-A as shown in .
3. Associate Promo-D to CP-A using the promotion picker.

The third figure in this topic shows the expected output for attribute defaults when Promotion Automatch is set to Default Cardinality, and when Promotion Automatch is set to Min Max Cardinality.

> **Note:** The expected output for attribute defaults when you apply a promotion to a customizable product is the same whether Promotion Automatch is set to Min Max Cardinality or Default Cardinality.

For more information about how Siebel CRM behaves depending on the Promotion Automatch argument value, see *ApplyPromotionDefaults Method*.

Configuring Multiple Customizable Products

The *New View: Configuring Multiple CPs* enhances user experience, making it easier for users to view and customize multiple Customizable Products or components of a bundle promotion. The Explorer View allows sales representatives to switch back and forth when configuring multiple customizable products during an order capture process.

> **Note:** Existing customers have to apply repository updates to avail the feature.

To configure multiple customizable products

New Provide Scenario

1. Navigate to the Quote or Sales Order view.
2. Create a new document (quote or sales order).
3. Add Line Items to the document, either Customizable Products or bundle promotion.
4. Click Build in the document header .

5. You can also multi-select Line Items and Click Customize to open the Parallel Configurator view.

Edit Promotion Scenario

Sales reps can edit bundle promotion from line items by clicking Edit Promotion or from catalog view. Refer to *OrderManagment Guide Addendum for Communications* for Editing a bundle promotion for more details.

In the Edit Promotion view, agent can click on Build button to open the Explorer view to configure the bundle promotion and the CPs within the bundle promotion.

New View: Configuring Multiple CPs

The new view consists of a Header applet, Explorer applet and the Parallel configurator for the CPs:

- Header applet is collapsible, it shows non-recurring and recurring prices.
  The prices gets updated whenever the user switches from one customizable product to other.
- The Explorer applet shows the bundle promotion and CPs in parent/child hierarchy.
- When the root CP is touched, the flag icon is shown.
- Newly added Items are shown with the plus icon.
- An exclamation mark shows up for eligibility and compatibility warnings.
- Tool tips shows the pricing information and exact E&C message.
- Alert user when user is trying to leave the view with incomplete product configuration using Save All or when switching from one customizable product to other.

Changing Save All button behavior

In configuring multiple customizable products view, when you click **Save All** button, you are shown the following message:

One or more products have incomplete configuration. Click the Cancel button to return and complete the configurator. Click the Incomplete button to Save the incomplete configuration.

And then you have a choice to either click Ok or Cancel.

Developers can change the configuration to block the user if the product configuration is incomplete.

1. In Siebel Web Tools, create a new workspace.
2. Navigate to applet SIS OM Config Playbar Applet.
3. Add following optional Applet User Properties and values:
   Name: `ClientPMUserProp`
   Value: `BlockExitOnIncompleteInstance,BlockExitOnIncompleteInstanceMsg`
   Name: `BlockExitOnIncompleteInstance`
   Value: `True`
   Name: `BlockExitOnIncompleteInstanceMsg`

Value: Any message text you want.

4. Test and deliver the workspace.

Note: If you do not define `BlockExitOnIncompleteInstanceMsg` (remove it from value for `ClientPMUserProp` too), it will display only the first line of the message for several incomplete instances, that is **One or more products have incomplete configuration**.

Get Product Instance Information API

While using configuring multiple customizable products view, GetProdInstInfo API is introduced to get the in-memory product instance information. This information can be used to enforce inter-customizable products rules.

- This API takes the path of the product along with the field name (quantity, action code or other fields) as input and retrieves the values from in-memory instance.
- API can also retrieve XA attribute values.

The output of *GetProdInstInfo* is a Siebel Message containing child property sets:

- Success: A propertyset representation of a successfully retrieved product data path and its values.
- Error: A propertyset representation of unable to retrieve product data path and error reason as a value.

Input Argument	Property path of Line item product to Attribute or Child Product

	Path with Integration Id:
	
	`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xa[AttrName].xf[Value]`
	`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Quantity]`
	`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Action Code]`
	`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Service Id]`

	OR
	Property path of Root product to Attribute or Child Product

	Path with Root Product Id:
	
	`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProduct Id>]#1.xa[AttrName].xf[Value]`
	`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProductId>]#1.xf[Quantity]`
	`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProductId>]#1.xf[Action Code]`
	`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Service Id]`

Output Argument	The output of *GetProdInstInfo* is a Siebel Message containing two child property sets:

	• **Success**: A child propertyset representation of a successfully retrieved product data path(s) and their values • **Error**: A child propertyset representation of unsuccessful product data path(s) and error reason(s) as a value.

Using the GetProdInstInfo API for above use-cases

Input Arguments Examples

Format of Path with Integration ID	Example
`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Quantity]`	`"$Int.[88-2ZLRCG].[88-15HMH]#[88-14SIX]#1.xf[Quantity]"` `"$Int.[88-2ZLRDR].[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[ 88-17KFI]#1.xf[Quantity]"`
`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xf[Action Code]`	`"$Int.[88-2ZLRCG].[88-15HMH]#[88-14SIX]#1.xf[Action Code]"` `"$Int.[88-2ZLRDR].[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[ 88-17KFI]#1.xf[Action Code]"`

`$Int.[<Root LinteItem Integration Id>].[<Relationship Id>]#[<ChildProduct Id>]#1.xa[AttrName].xf[Value]`	`"$Int.[88-2ZLRDR].[1-27LD6]#[88-17KFU]#1.xa[OM_Build].xf[Value]"`
	`"$Int.[88-2ZOU1U].xa[Deductible-Individual$_In Network].xf[Value]"`
	`"$Int.[88-2ZOU1U].xa[X-ray & Lab testing Coverage%_Out Network].xf[Value]"`

`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProductId>]#1.xf[Quar`	`"$.[1SIA-7P09]#1.[88-15HMH]#[88-14SIX]#1.xf[Quantity]"` `"$.[88-17KGC]#1.[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[ 88-17KFI]#1.xf[Quantity]"`
`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProductId>]#1.xf[Acti Code]`	`"$.[1SIA-7P09]#1.[88-15HMH]#[88-14SIX]#1.xf[Action Code]"` `"$.[88-17KGC]#1.[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[ 88-17KFI]#1.xf[Action Code]"`

`$.[<Root Product Id>]#1.[<Relationship Id>]#[<ChildProduct Id>]#1.xa[AttrName].xf[Value]`	`"$.[88-17KGC]#1.[1-27LD6]#[88-17KFU]#1.xa[OM_Build].xf[Value]"`
	`"$.[1-1EVI]#1.xa[Deductible-Individual$_In Network].xf[Value]"`
	`"$.[1-1EVI]#1.xa[X-ray & Lab testing Coverage%_Out Network].xf[Value]"`

Output Arguments Example

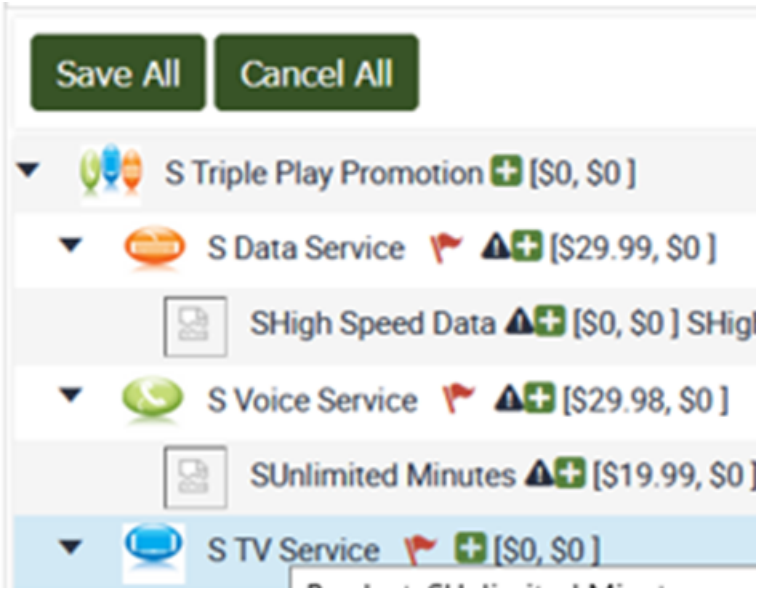
Success Example	Error Example
Child property set [0]	**Child property set [1]**

Success Example	Error Example
c[0] CCFPropertySet@1C5A31B8 p#0 c#9 type="Success" vt=0 value="" { "$Int.[88-2ZLRCG].[88-15HMH]#[88-14SIX]#1.xf[Quantit "2" "$Int.[88-2ZLRDR].[1-27LD6]#[88-17KFU]#1.xa[OM_Build "250" "$Int.[88-2ZOU1U].xa[X-ray & Lab testing Coverage%_Out Network].xf[Value]", "50" "$Int.[88-2ZLRDR].[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[88-17KF]#1.xf[Quantity] "1" "$Int.[88-2ZLRCG].[88-15HMH]#[88-14SIX]#1.xf[Action Code]", "Add" "$Int.[88-2ZLRDR].[1-27LLO]#[88-17KG5]#1.[1-27KNE]#[88-17KFK]#1.[1-27KZI]#[88-17KFI]#1.xf[Action Code]", "Add" }	c[1] CCFPropertySet@1C5A31B9 p#0 c#10 type="Error" vt=0 value="" { "$Int.[88-2ZOU1U].xa[Deductible-Individual$_In Network].xf[Value]", "Invalid Syntax" }

Note: The row-IDs in the above example are for sample data only.

Invoking GetProdInstInfo for Telecommunications Use case

Use case



In a triple play promotion, eligibility of TV service plans (High Definition) is dependent on speed in the data service.

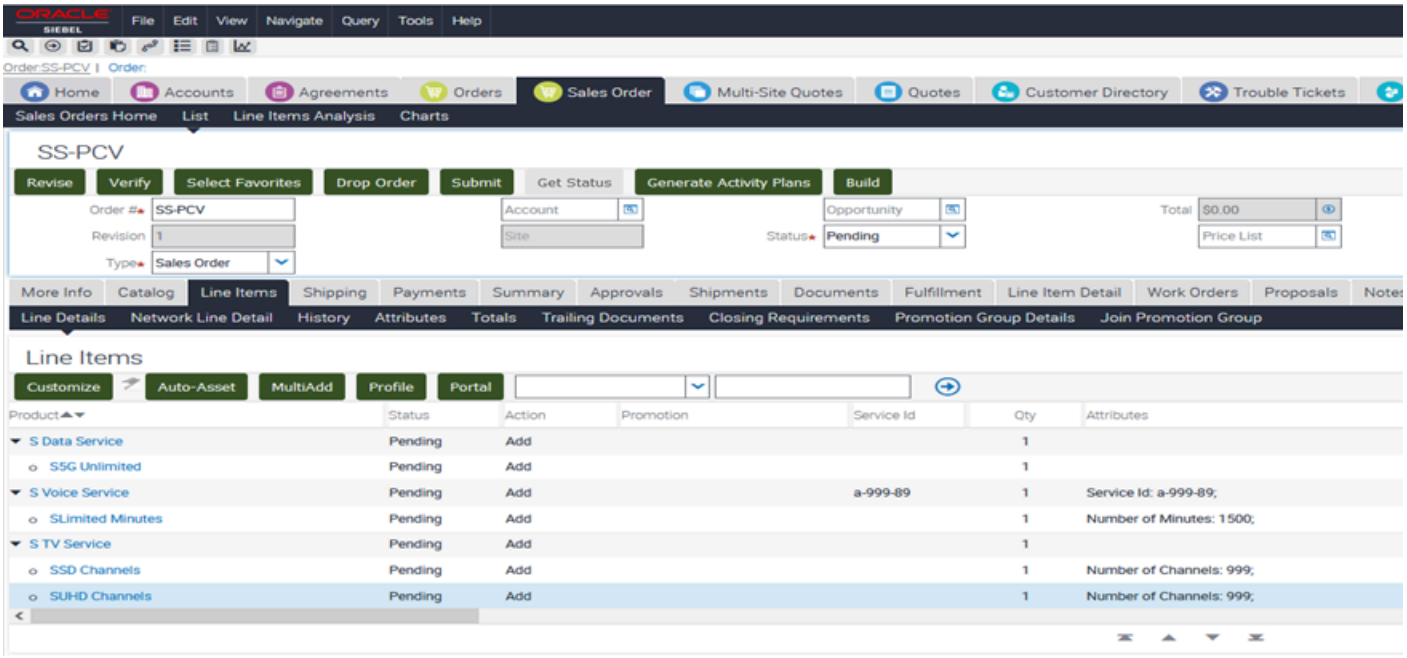
1. Siebel product constraints can be based on linked items. The linked items are initialized when configurator is launched. There is a need to get the values of linked items when users are switching between the root products in the parallel configurator.
    a. Linked Item `DataActionCode` derived from **SHigh Speed Data** from **S Data Service** product. This is set in a profile attribute and mapped to linked item.
    b. Product constraint:
    When DataActionCode = Add excludes selection of SHD Channels from Plan Types
    When DataActionCode = Add excludes selection of SUHD Channels from Plan Types
    c. Click on Build button.
    d. Select the S TV Service to be able to see the SHD and SUHD child products.
    e. Select S Data Service and add the product SHigh Speed Data.
    f. Select the S TV Service again and then SHD Channels and SUHD Channels are shown in Red color. If you try to select it, configurator violation is shown as expected.
    g. Between point ‘a’ and ‘f’ there is no Database interactions and all computation is in-memory.
2. The Line Item IC Field linked to attributes are displayed in the parallel configurator session and can be updated by the user and saved to line items.

The service id value is retained while switching product within parallel configurator.

For example:

a. Click Build button.
b. Select product **S Voice Service** and set the Line item IC field Service Id value as '123'.
c. Select the S Data Service and make some changes in that product.
d. Again select S Voice Service.

   The expectation is the value '123' for Service Id to be retained.
e. Between point 'a' and 'd' there is no Database interactions and all in memory.



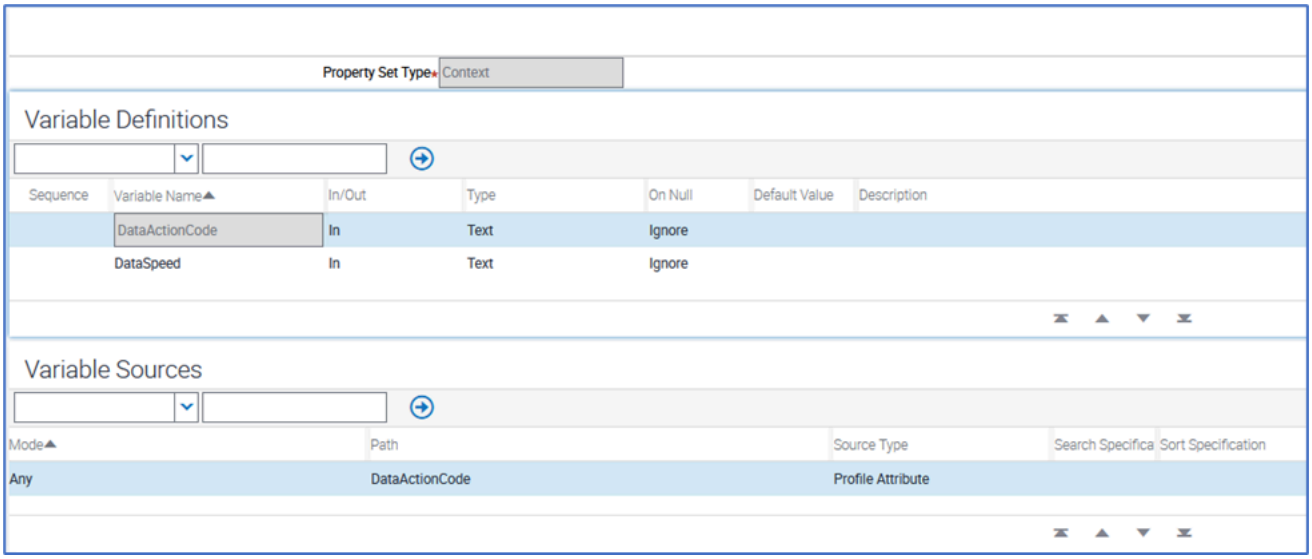
Related topics

- *Define Variable Maps*
- *Create Linked Items*
- *Define constraints*
- *Custom Business Service*
- *Modify Workflows*

Define Variable Maps

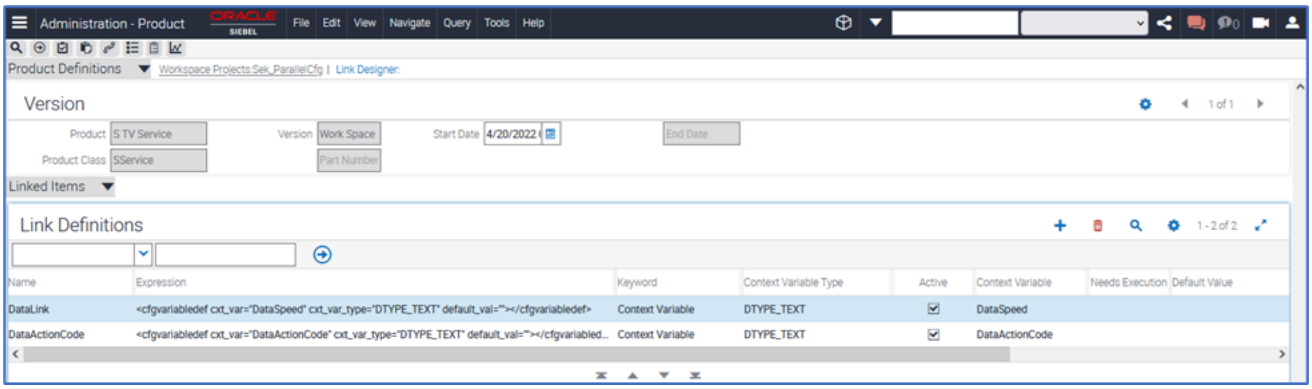
Modify Variable Map Cfg Eligibility Variable Map - Context, add:

VariableDefinitions	Variable Sources – Source Type	Profile Attribute(s)	Comments
DataActionCode	Profile Attribute	DataActionCode	Mapping of variable to profile attribute
DataSpeed	Profile Attribute	DataSpeed	Mapping of variable to profile attribute



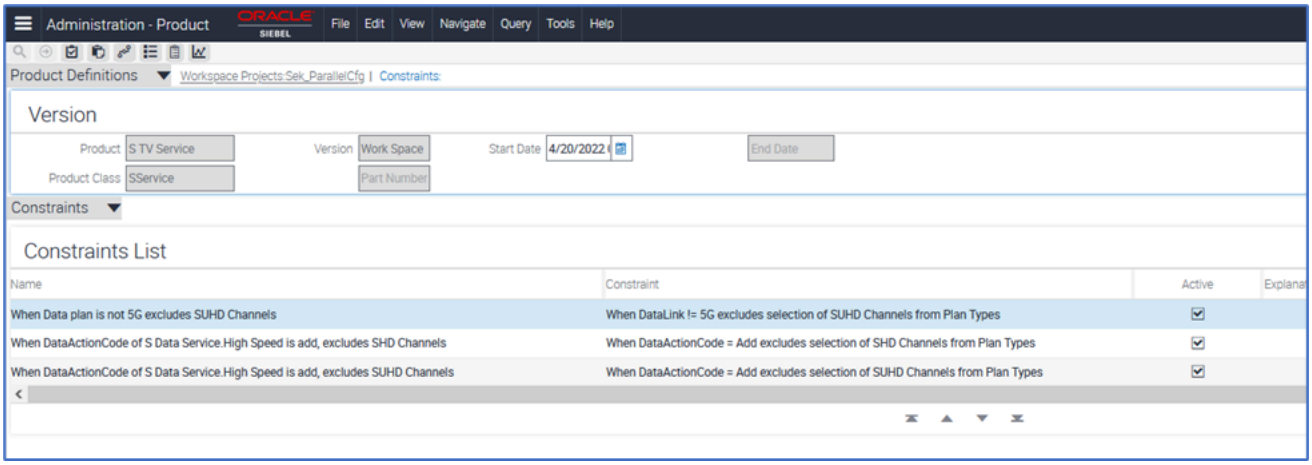
Create Linked Items

Create Linked Items with `Keyword = Context` Variable on the product where constraints are defined:



Define constraints

Define constraints in the Constraints List.



Custom Business Service

Retrieve the product attributes using API and assign to profile attributes using custom business service GetLinteItemProdData with below function.

```
function GetLinteItemProdData(Inputs)
{
 try{
 var Prod_SDataService = "88-31ILGQ" // root product id of S Data Service
 var Prod_STVService = "88-31ILXP" // root product id of S TV Service
 var inp : PropertySet = null;
 var out : PropertySet = null;
 var oBS_InMemUDS : Service = null;

 TheApplication().Trace("START GetLinteItemProdData method" );

 var CurProdId = Inputs.GetProperty("ProductId");
 TheApplication().Trace("Current Product Id ::: " + CurProdId );

 oBS_InMemUDS = TheApplication().GetService("InMemory Upgrade Data Service");
 inp = TheApplication().NewPropertySet();
 out = TheApplication().NewPropertySet();

 if( CurProdId == Prod_STVService)
 {
 var DataPlan5GQty = "$.[" + Prod_SDataService + "]#1.[88-31ILHU]#[88-31ILGG]#1.xf[Quantity]"; // get 5G
 Unlimited Quantity
 inp .SetProperty(DataPlan5GQty , "");

 var HighSpeedACode = "$.[" + Prod_SDataService + "]#1.[88-31ILHU]#[88-31ILEP]#1.xf[Action Code]"; // High
 Speed Action Code
 inp .SetProperty(HighSpeedACode, "");
```

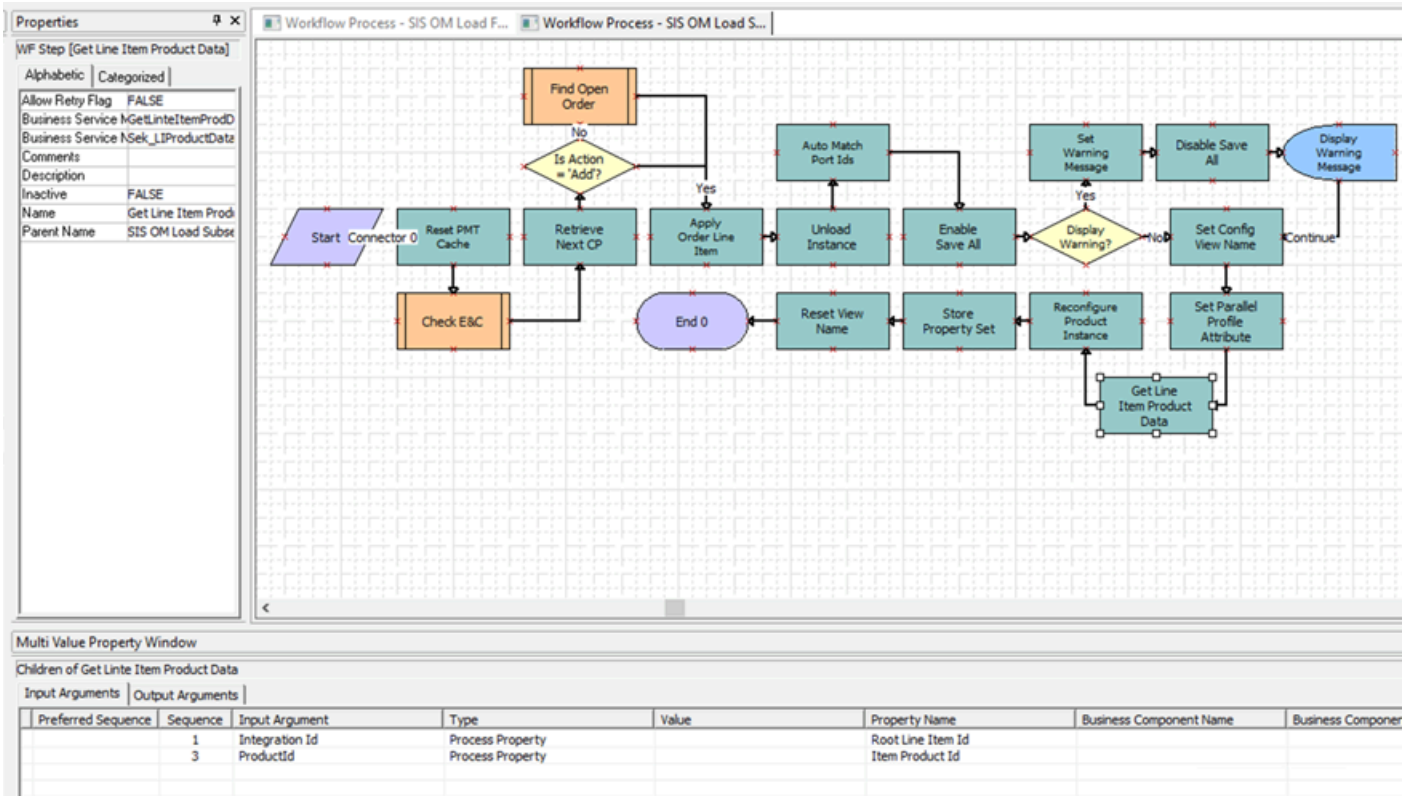
```
oBS_InMemUDS .InvokeMethod( "GetProdInstInfo", inp, out ); // new API for parallel configurator to get
product instance data from in-memory

var outSuc = out.GetChild(0); //Success PS

var outErr = out.GetChild(1); //Error PS

if (outErr.GetChildCount() > 0) //Error occurred
{
TheApplication().Trace(" Number of failed paths: " + outErr.GetChildCount());
}
else
{
var Data5Gqty = outSuc.GetProperty(DataPlan5GQty )
if(Data5Gqty == "1")
{
TheApplication().SetProfileAttr("DataSpeed", "5G");
}
else
{
TheApplication().SetProfileAttr("DataSpeed", "");
}

TheApplication().SetProfileAttr("DataActionCode", outSuc.GetProperty(HighSpeedACode ));

TheApplication().Trace("Profile Attribute DataSpeed ::: " + TheApplication().GetProfileAttr("DataSpeed"));
TheApplication().Trace("Profile Attribute DataActionCode ::: " +
TheApplication().GetProfileAttr("DataActionCode"));
}
PropertySetToFile(out);
}
}finally{
oBS_InMemUDS = null;
inp = null;
out = null;
TheApplication().Trace("END GetLinteItemProdData method" );
TheApplication().TraceOff();
}
}
```

Modify Workflows

Workflows used in the parallel configurator:

1. SIS OM Load First Quote Document
2. SIS OM Load Subsequent Quote Instance
3. SIS OM Load First Order Document
4. SIS OM Load Subsequent Order Instance

Add this same step in the workflow **Get Line Item Product Data** to invoke the business service whenever the first customizable product is loaded or when the user switches between the Customizable Products within the parallel configurator view.


Properties
WF Step [Get Line Item Product Data]
Alphabetic
Categorized
Allow Retry Flag FALSE
Business Service M GetLinteItemProdD
Business Service N Sek_LIProductData
Comments
Description
Inactive FALSE
Name Get Line Item Prod
Parent Name SIS OM Load Subse
Workflow Process - SIS OM Load F...
Workflow Process - SIS OM Load S...
Find Open Order
No
Is Action = 'Add'?
Yes
Start
Connector 0
Reset PMT Cache
Retrieve Next CP
Apply Order Line Item
Unload Instance
Auto Match Port Ids
Enable Save All
Set Warning Message
Yes
Display Warning?
No
Disable Save All
Display Warning Message
Continue
Set Config View Name
Set Parallel Profile Attribute
Reconfigure Product Instance
Store Property Set
Reset View Name
End 0
Check E&C
Get Line Item Product Data
Multi Value Property Window
Children of Get Linte Item Product Data
Input Arguments
Output Arguments
Preferred Sequence
Sequence
Input Argument
Type
Value
Property Name
Business Component Name
Business Componer
1
Integration Id
Process Property
Root Line Item Id
3
ProductId
Process Property
Item Product Id

15 Working with Quotes

Working with Quotes

This chapter describes how end users work with existing quotes. It includes the following topics:

- *About Quotes*
- *Updating Opportunities from Quotes*
- *Including Quote Items in a Forecast*
- *Accessing An Existing Quote*
- *Revising a Quote*
- *Viewing Change History for Quote Line Items*
- *Adding Line Item Fields to the Change History*
- *Converting a Quote to an Order*
- *Converting a Quote to an Agreement*
- *Displaying Charts to Analyze Quotes*

For information about creating a new quote, see *Creating a Quote or Sales Order*.

About Quotes

A quote is an offer to a customer for specific products and services at a specific price. Quotes can be generated by end users such as call center agents, sales representatives, partner employees, and customers using Siebel eSales. The quote may include products and services (configured to show the customer-selected attributes), effective dates, price lists, any discounts by line item and by account, and the price for each item.

For information about creating quotes, see *Creating a Quote or Sales Order*.

Updating Opportunities from Quotes

When a quote is associated with an opportunity, an end user can update the opportunity with new quote information. When an end user updates the opportunity, information is updated in the Products and Revenues tab. The revenue associated with the quote is transferred to the opportunity listed in the quote's Opportunity field. It is important to keep opportunity information updated in this way in order to maintain accurate forecasts and charts.

Clicking Update Opportunity deletes only lines associated with the corresponding quote. For example, a monitor appears in two separate quotes, and Update Opportunity is run for both quotes, so the monitor appears twice in the products for the opportunity. If you go to one of the quote line items, delete the monitor line item, and click Update Opportunity, then only the opportunity product and revenue line associated with that quote line item is deleted.

If you do not see the quote number associated with an opportunity product, then use the Update Opportunity method from the menu for the association to happen.

Note: If you are using multiple price types, then updating an opportunity from a quote only creates revenue records for one time charges. It does not create multiple revenue records for recurring charges. This feature works differently in industry applications, as described in the topic about updating opportunities from quotes in *Siebel Order Management Guide Addendum for Communications* .

CAUTION: Before updating an opportunity from a quote, verify that the associated price list is available for the quote. If the price list is not available, then the product is deleted from the opportunity.

To update an opportunity from a quote

1. Navigate to the Quotes screen, and then the Quotes List view.
2. In the Quotes list, select the desired quote.
3. In the Quotes form, from the menu, select Update Opportunity.

Including Quote Items in a Forecast

When an end user updates an opportunity from a quote, one piece of information that is sent is whether a product is forecastable. The default setting for this functionality in the quote is determined by a setting in Product Administration. By default, in Product Administration, components of products with components and product bundles are marked as forecastable, unless explicitly changed. Simple products are marked as forecastable and cannot be changed in Product Administration. Components of bundles are marked as not forecastable by default, but can be changed to forecastable.

If end users want to make sure that a product in a quote is forecastable, then they can review the setting in the quote, and mark it as forecastable if necessary.

To include a quote item in a forecast

1. Navigate to the Quotes screen, and then the Quote list view.
2. In the Quotes list, drill down on the name of the desired quote.
3. Click the Line Items view tab, select the desired line item.
4. In the Line Item Detail subview, make sure that the Forecastable check box is selected.
5. If it is not already selected, then select the Forecastable check box to include this item in forecasts.
6. When you are finished working on the quote, update the opportunity. For details, see *Updating Opportunities from Quotes*.

Accessing An Existing Quote

When an end user wants to review an existing quote or make changes to it, the end user can access the quote from a number of different places in your Siebel Business Application. The procedures in this documentation primarily describe accessing quotes from the Quotes screen, but end users can also access quotes from an opportunity, from an account, from a contact, or from a service request.

To access an existing quote

1. Navigate to one of the following:
    - Opportunities screen, and then the Opportunities List view
    - Accounts screen, and then the Accounts List view
    - Service Requests screen, and then the Service Request List view
    - Contacts screen, and then the Contacts List view
2. Drill down on opportunity, account, service request, or contact that the quote is associated with.
3. Click the Quotes view tab.
4. In the Quotes list, drill down on the quote name.

   The quote is displayed in the Quotes screen.

The end user can now make additions or changes to the quote, if it is still in progress, or the end user can revise the existing quote, if appropriate.

Revising a Quote

End users can make changes to an initial quote to keep it updated with the appropriate products, price lists, discounts, and so forth. During negotiations with a customer, an end user can add or delete products from a quote and compare features and price.

The Revise feature allows an end user to revise your original quote by creating a new quote record with the same quote number. The versions are distinguished by a revision number. All revisions of a quote appear in the quote list.

Making changes to a quote does not automatically create a new version of the quote. If the end user wants to keep the original quote and have a revision, then the user must complete the steps in the following procedure to create a revised quote.

Note: When the end user revises a quote, the original quote becomes inactive and read-only.

To revise a quote

1. Navigate to the Quotes screen, and then the Quotes List view.
2. In the Quotes list, select the quote you want to revise.
3. Click Revise.

   A new quote, with the same quote number is created.
4. Edit the information in the revised quote as needed.

   In the Quotes screen you see both the original quote and the revised quote. You can verify which one is the revision by looking in the Revision column.

Viewing Change History for Quote Line Items

End users can view change history for line items in quotes. They can view previous and current values for the following line item fields:

- Billing Account
- Service Account
- Service Point
- Asset Status
- Service ID

Note: Change history is relevant when asset-based ordering is enabled. For more information about enabling asset-based ordering, see *Setting Up Asset-Based Ordering*.

Viewing the Change History for a Quote Line Item

To view the change history for a line item in a sales quote, complete the steps in the following procedure.

To view change history for a quote line item

1. Navigate to the Quotes screen, and then the List view.
2. Click the number for the quote.
3. Click the Line Items tab.
4. Select a line item and click the Changes link.

   If the Changes link does not appear, then navigate to the Quotes screen, Line Items, and then the Changes view.

   The Modify History Line Items form appears showing previous and current values for line item fields. After the form, the Line Item Attributes list displays previous and current values for line item attributes.

Adding Line Item Fields to the Change History

You can add fields to the line item change history by completing the steps of the following procedure.

To add a line item field to the change history

1. Start Siebel Tools.
2. In the S_QUOTE_ITEM_OM table, create a column for the previous value you want to display.
3. In the Quote Item business component, create a field based on the column you added.

   You can, for example, give this field the display name Previous.
4. Add the new Previous field to the Quote Item Modify History List Applet and the Quote Detail Line Item Modify Applet.

5. In the SIS OM PMT Service user properties, add a Delta Old Field user property for the new field.
6. Follow the standard process to carry the field from Quotes to Orders when Asset Based Ordering is enabled by performing the following steps:
   a. Add the fields in the business components on which the SIS OM Quote.Line Item and SIS OM Order.Line Item integration components are based. These business components are MACD Quote Item and MACD Order Entry - Line Items, respectively.
   b. Add a new user property mapping these integration component fields in the SIS OM PMT Service business service.

   The new field appears in both the quote line item and order line item change histories.
7. Save the changes and submit the Workspace for delivery.

For more information about adding objects in Siebel Tools, see the chapter on working with objects in *Using Siebel Tools* .

Converting a Quote to an Order

When a customer agrees to purchase the items on a quote, an end user can convert the quote to an order.

To convert a quote to an order

1. Navigate to the Quotes screen, and then the Quotes List view.
2. In the Quotes list, drill down on the name of the desired quote.
3. Review the line items and related details and make sure they contain the correct information.
4. Click the Orders view tab.
5. Click either Sales Order or Service Order, depending on the type of order you want to create from this quote.

   An order is created from the quote, and appears in the Orders list.
6. To work with the order, drill down on the order number.

   The Sales Order screen, Line Items view or the Service Order screen, Line Items view appears.

Converting a Quote to an Agreement

Rather than converting a quote to an order, you can convert a quote to an agreement. There are a number of reasons that you would create an agreement from a quote:

- To create recurring service contracts that identify terms and conditions, including service level agreements and payment terms.
- To create of pricing contracts that identify sets of products and services and associated pricing for future purchases.

For more information about agreements, see *Siebel Field Service Guide* .

To convert a quote to an agreement

1. Navigate to the Quotes screen.

2. In the Quotes list, drill down on the name of the desired quote.
3. Review the line items and related details and make sure they contain the correct information.
4. Click the Agreements view tab.
5. Click AutoContract.

   An agreement is created from the quote, and appears in the Agreements list.
6. To work with the agreement, drill down on the agreement number.

   The Agreements screen, Line Items view appears.

Displaying Charts to Analyze Quotes

You can use the following charts to analyze quotes:

- **Account Analysis.** Shows the number of quotes for each account.
- **Account Discount Analysis.** Shows the average discount for the quotes for each account.
- **Rep Discount Analysis.** Shows the average discount for the quotes for each sales representative.
- **Status Analysis by Rep.** Shows the status of the quotes for each sales representative. Typical statuses are In Progress, Approved, Accepted, and Order Placed.

> **Note:** The charts analyze all quotes displayed in the Quotes list. It does not matter which quote is selected when you click the Charts view tab. You can display the My Quotes view, My Team's Quotes view or All Quotes view before creating a chart. You may also use a query to limit which quotes are displayed in the list and included in the chart.

To display charts analyzing quotes

1. Navigate to the Quotes screen, and then the Charts view.
2. Optionally, use a query in the Quotes list to display only the quotes that you want to be included in the chart.
3. In the Show list, select the type of chart you want to view.
4. To view different types of chart, such as pie charts and horizontal bar charts, select the chart type from the other list and click Go.
5. If you are viewing the Status Analysis by Rep chart, then you can use the By list to display the result either by sales representative or by status.

16 Working with Sales Orders

Working with Sales Orders

This chapter describes how end users work with existing orders. It includes the following topics:

- *About Orders*
- *Accessing an Existing Order*
- *Submitting an Order*
- *Associating an Activity Plan or Activity with an Order*
- *Approving an Order*
- *Checking the Status of Order Shipments*
- *Viewing Shipping Details for Orders and Order Lines*
- *Placing an Order on Hold*
- *Canceling an Order*
- *Revising an Order*
- *Viewing Change History for Sales Order Line Items*

About Orders

An order is a commitment on the part of the customer to purchase products and services at a specific price. Orders can be generated from quotes or created directly by sales people, call center agents and sales administrators. Orders can be created and placed all at once, or developed in stages as the customer clarifies the configuration of the items, requests availability, payment information, shipping information, and other details. When the order is complete, the end user submits it.

After the order is placed, a call center agent can monitor the order, proactively informing the customer of order status and delivery information. In the case of a customer inquiry, an agent can use the Get Status business service to obtain the latest information and determine the status of the order, if the order is being processed by the back office. In the event that post-order service is required, service requests and returns can be created against the order.

Orders are identified as either sales orders or service orders. Sales orders are generally for new products, while service orders are used to support service parts processing, including part locator, repair depot, and so on.

End users work with sales orders and service orders in very similar ways, however some features are specific to each type. This chapter focuses primarily on sales orders. For details about working with service orders, see *Siebel Field Service Guide* .

For information about creating a new order, see *Creating a Quote or Sales Order*.

Accessing an Existing Order

An end user can access an existing order from most of the same places where the end user can create an order.

In addition to accessing the order from the Sales Order screen, the end user can access it from other screens, as described in the following procedure.

To access an existing order

1. Navigate to one of the following screens:
    - Accounts screen, and then the Accounts List view
    - Service Requests screen, and then the Service Request List view
    - Contacts screen, and then the Contacts List view
    - Agreements screen, and then the Agreements List view
2. Drill down on the account, service request, contact or agreement that is associated with the order.
3. Click the Orders view tab.
4. In the Orders list, drill down on the order number of the desired order.

    The order is displayed in the Sales Order of Service Order screen, where the end user can continue to work with it.

Submitting an Order

After the end user has completed the information for an order, the user submits it. Submitting the order sends the order to the manufacturing application or warehouse application for fulfillment.

Submitting the order typically sets the status to Booked.

Before end users submit an order, they may want to check to make sure it has an Active status, current dates, a valid price list, and that any customizable products have been configured appropriately.

To submit an order

1. Navigate to the order that you want to submit.
2. In the Sales Order form, click Submit.

    If your company has configured an approval process, then the order is evaluated by assignment manager rules, and is routed to a manager for approval, if appropriate. If the order meets any approval rules, then the order is validated and submitted to your back office application for fulfillment.

Associating an Activity Plan or Activity with an Order

Individual activities or activity plans (collections of activities) can be associated with line items in your order.

Activities are tasks that must be performed after the order has been finalized and submitted. For example, when a customer purchases an item that needs to be installed, an activity template might list all of the activities that need to occur to support that installation. When that template is added to the line item, the activities from it are assigned to the appropriate people, and will appear in their activity lists.

Typically the administrator sets up activity templates in advance so that end users can add them to items in an order. End users can also add individual activities. For information on setting up activity templates, see *Creating Activity Plan Templates*.

To associate an activity plan with an order

1. Navigate to the Sales Orders screen, Line Items, and then the Activity Plans view.
2. Select the order and line item that requires activities.
3. In the Activity Plans list, add a new record.

   Today's date is assigned as the planned start date for the activities. You can change it, if required.
4. Select the appropriate activity plan template from the Templates list.

   You see the activities associated with the template in the Activities list.
5. Click the Activity view tab.
6. In the Activity list, make any appropriate adjustments to the activities.

   If an employee's name is indicated in the activity, then the activity will display for that person when the person goes to the My Activities view.

To associate an activity with an order

1. Navigate to the order with which you want to associate an activity.
2. Select the line item that requires activities.
3. Click the Activity view tab.
4. Click New to add a new record.

   The activity type defaults to Appointment.
5. Complete the fields to define the activity.

Approving an Order

Some orders will require approval. If your company has configured an approval process as part of its order management process, then an order may be automatically flagged for approval depending on price. A typical approval process would cause the status to be changed to Awaiting Approval and the approver's name to be inserted in the Approved by field when the sales representative clicks Submit.

For example, if an end user has authority to place orders for $1,000, and a customer wants to order merchandise totalling $1,200, then the order needs to be approved before it can be submitted.

The steps in the following procedure show how an end user might take to approve an order when that user is the approver. The exact steps may vary, depending on your company's configuration of this process.

To approve an order

1. Navigate to the Sales Orders screen, and then the Orders List view.
2. Sort the list of orders by status and, if necessary, bring orders with the status Awaiting Approval into view.
3. Drill down on the order number of an order you want to approve.
4. Review the details of the order to make sure it meets your approval.
5. When you are ready to approve the order, in the More Info view tab, select the Approved check box.

Checking the Status of Order Shipments

After an order has been submitted, an end user can monitor it by checking the status of order shipments. The end user makes sure the order is being handled appropriately and to provide customers with information about order status and delivery.

You must have a back office application that tracks shipping of your products, and you must integrate that application with the Siebel Business Application, to allow the end user to see the shipping status of products.

To check the status of order shipments

1. Navigate to the order you want to check.
2. Click the Shipments view tab.

   The Shipment list and the forms under it provide current details about the shipment of the order.

   The read-only Back Office Error Text field may contain notices from the back-office about order errors, such as out-of-stock products or discontinued products.

   The Back Office Error Text field is also available at the Order Header level and in the Order Entry Form applets.

Viewing Shipping Details for Orders and Order Lines

If you have enabled the Order-to-Cash PIP, you can access back-office information about order and line item shipping in Siebel CRM. This information is read-only, and it is entered and modified only by back-office personnel. For more information see *Integrating Order Management with Back-Office Applications*.

To view shipping details for an order and order line

1. Navigate to the Sales Order screen, then the List view.
2. In the Sales Order list, find the order for which you want to view shipping details.
3. Drill down on the link in the Order # field.

4. Click the Shipping view tab.

   The Shipping form displays shipping information for the order sent from the back-office system. The information fields include Pack Instructions and Ship Instructions.
5. Click the Line Items subview tab.

   The Shipping list, which precedes the Line Items form, displays additional shipping information for the order sent from the back-office system. The information fields include Change Reason, Package, Pack Instructions, Ship Date, Ship Instructions, Shipping, and Requested Date.
6. Click the Ship Detail subview tab.
7. In the Shipping list, which precedes the Ship Detail list, select the order line item for which you want to view shipping details.

   The Ship Detail list displays shipping information for the selected order line item sent from the back-office system. The information fields include Tracking Number, Product, Source Inventory Location, Quantity, Ship Date, and Ship Carrier.

Placing an Order on Hold

Users can stop orders temporarily by putting them on hold and selecting a reason for the hold from a configurable list.

For more information about configuring lists of values, see *Siebel Applications Administration Guide* .

To place an order on hold

1. Navigate to the order that you want to put on hold.
2. In the order header select the Hold box.
3. From the Hold Reason list, select a reason for placing the order on hold.

   Note: The Hold Reason list can be configured in the Administration - Data screen.
4. In the Order header summary, click Submit.

Canceling an Order

If an order header status is Booked or Awaiting Shipping, then a customer service representative or sales agent can cancel the order. Orders cannot be canceled when the order header status is Shipped, Fulfilled, or Closed.

Note: Do not cancel an order by setting the order Status field to Canceled.

To cancel an order

1. Navigate to the order that you want to cancel.
2. Revise the order, which creates a new version of the order.
3. Choose a cancellation reason from the Cancel Reason list.

4. Click Submit to submit the order for cancellation.

   The order is set to Cancel Pending. When the order is canceled successfully in the back office system, the order status is set to Canceled in the Sales Order form.

Revising an Order

An end user can edit or revise a customer's original order, which creates a new version of the same order. Revising helps an end user keep track of changes as the order evolves. This feature is useful when adding or deleting products, and can also be used if an order has expired.

When an end user revises an order, the status field is automatically set to Pending.

The Revise feature allows an end user to revise the original order by creating a new order record with the same order number. The versions are distinguished by a revision number. All revisions of an order appear in the order list.

Making changes to a order does not automatically create a new version of the order. If the end user wants to keep the original order and have a revision, then the user must complete the steps in the following procedure to create a revised order.

Note: When an end user revises an order, the original order becomes inactive and read-only.

To revise an order

1. Navigate to the order you want to revise.
2. In the Sales Order form, click Revise.

   A new version of the order is created, with the same order number.
3. Make the required changes to the order—add or remove line items, change attributes, or make additional changes.
4. In the Order header summary, click Submit.

Viewing Change History for Sales Order Line Items

End users can view change history for line items in sales orders. They can view previous and current values for the following line item fields:

- Billing Account
- Service Account
- Service Point
- Asset Status
- Service ID

Note: Change history is relevant when asset-based ordering is enabled. For more information about enabling asset-based ordering, see *Setting Up Asset-Based Ordering*.

You can add fields to the order items change history by the same method you would use for adding fields to the quote items change history. For more information about adding fields to the line items change history display, see *Adding Line Item Fields to the Change History*.

To view the change history for a line item in a sales order, complete the following procedure.

To view change history for a sales order line item

1. Navigate to the Sales Order screen, and then the List view.
2. Click the number for the sales order.
3. Click the Line Items tab.
4. Select a line item and click the Changes link.

   If the Changes link does not appear, then navigate to the Sales Orders screen, Line Items, and then Changes view.

   The Modify History Line Items form appears showing previous and current values for line item fields. After the form, the Line Item Attributes list displays previous and current values for line item attributes.

17 Using Asset-Based Ordering

Using Asset-Based Ordering

This chapter describes how employee end users, such as salespeople and customer service representatives (CSRs), work with quotes and orders using asset-based ordering. It includes the following topics:

- *About Asset-Based Ordering*
- *Basic Concepts of Asset-Based Ordering*
- *Scenario for Asset-Based Ordering*
- *Creating a Quote Using Asset-Based Ordering*
- *Creating a Quote That Modifies an Asset*
- *Ungrouping Quote or Order Line Items*
- *Editing Quote Line Items Using Asset-Based Ordering*
- *Converting Quotes to Orders Using Asset-Based Ordering*
- *Editing Order Line Items Using Asset-Based Ordering*
- *Submitting an Order Using Asset-Based Ordering*
- *Revising a Submitted Order Using Asset-Based Ordering*
- *Creating a Follow-On Order*
- *Modifying the Service Profile*
- *Deactivating an Installed Asset*
- *Suspending and Resuming Service*

The procedures in this chapter use the Account Summary view as their starting point, because it is a central point for viewing information about an account. The Account Summary view is part of the Order Processing module. If you have not licensed the Order Processing module, then employee end users can start from one of the other screens that allow access to quotes and orders, such as the Quotes screen and Orders screen.

About Asset-Based Ordering

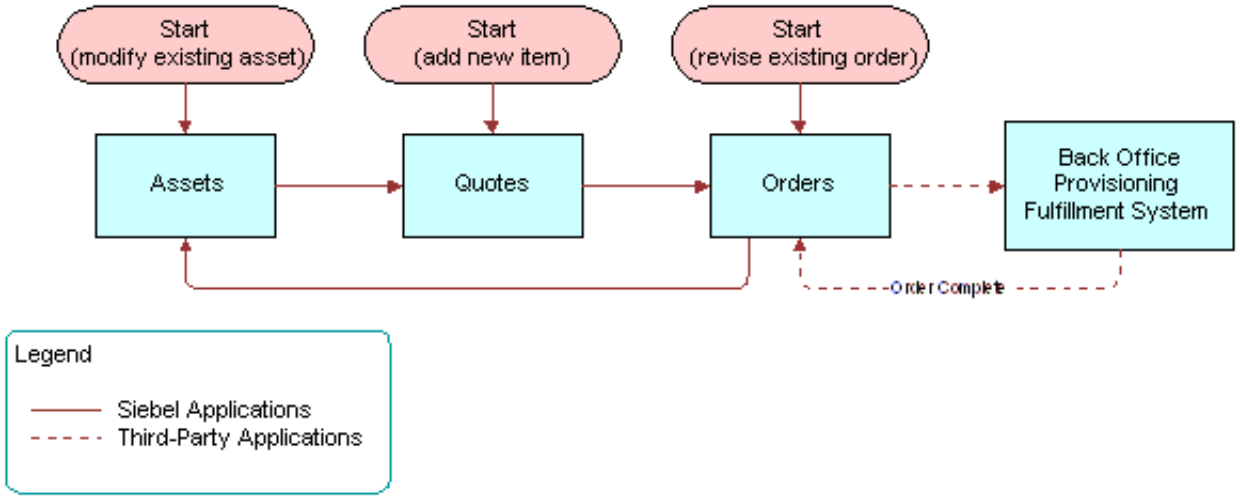
Asset-based ordering allows companies to:

- Generate asset records from orders
- Create quotes for new products and services using existing assets
- Create quotes to modify existing products and services
- Modify in-process orders that have been submitted for fulfillment
- Suspend, resume, or discontinue an existing service
- Have visibility into the asset life cycle during customer interactions

Asset-based ordering is included with Siebel Order Management, but it must be activated separately, as described in *Setting Up Asset-Based Ordering*.

Because asset-based ordering is based on workflows and business services, you can customize it to suit your business model. For more information, see *Workflow Reference for Employee Asset-Based Ordering*.

The following figure shows the cycle of events for asset-based ordering.



Basic Concepts of Asset-Based Ordering

Asset-based ordering gives end users the ability to create, modify, and process orders that contain complex pricing, combined billing types, combination services, or multiple suppliers. For example, a single sales order for a telecommunications customer can contain products and services from multiple suppliers, each provisioned in a different way. Such an order might include service installation, monthly-billed services (such as voice mail and usage-based calling plans), and feature services such as call forwarding. These services have different price types—one time, recurring, and usage-based—that need to be included in the same order.

The basic concepts of asset-based ordering are described in the following topics:

- *Customer Profile in Asset-Based Ordering*
- *About Products and Assets*
- *About Quotes and Orders for Asset-Based Ordering*

Customer Profile in Asset-Based Ordering

The customer's *service profile* is a record of all the active assets for that customer. An *asset* is a product or service currently provided to a customer that has been marked as a trackable asset in the product record, in the Product Administration screen. Assets are sometimes referred to as *installed assets* and are listed in the Installed Assets view.

When you have asset-based ordering turned on in Siebel Order Management, end users can use the Accounts screen, Installed Assets view to create orders using a customer's existing assets. Installed assets can include both products and services.

When these views are used with the automated asset-based order workflows, the amount of time to create quotes, orders, and asset records is reduced because many tasks are automated.

About Products and Assets

The following concepts related to products and assets are important in asset-based ordering.

Service Products

A service product is a product that is not a physical product, although it is often associated with one. For example, a service that is delivered over time, such as telephone or electrical service, is considered a service product. A service product may have physical components. For instance, a residential electric service would include a meter. A product becomes a service product if it is marked as one in the product record.

Integration ID

An *integration ID* is a unique identifier that is assigned to each trackable asset when it becomes a line item in a quote or an order.

The field is named *Asset Integration Id* in Order Line Items and Quote Line Items, but *Integration Id* in Assets.

> **Note:** It is important that this field not be used for other purposes, because using it might affect the functions of the Order Line Items, Quote Line Items, and Assets.

About Quotes and Orders for Asset-Based Ordering

A quote is an offer to a customer for specific products and services at a specific price. Quotes can be generated by employee end users such as call center sales representatives, by customers using Siebel eSales, or by a channel partner. The quote may include products and services (configured to show the customer-selected attributes), any discounts by line item and by account, price for each item, and the service and billing accounts for each item. If you use Multiple Price Types, then they detail one-time, recurring, and usage-based fees that are incurred.

For products with components, all levels in the product hierarchy are treated equally. Every level can have pricing, a service asset, and attributes. This structure allows end users to create one quote that spans multiple service accounts and billing accounts. For example, when placing a quote for phone services that span multiple locations, you can identify agreement names for individual line items, as well as applying appropriate pricing discounts and other rules.

Active Quote

An end user session can have only one active quote at a time. Each item that is added to a quote for a customer is added to the active quote. An active quote is useful, because it avoids creating new quotes when you are working with the same customer. The active quote adds all the quotes items of services or products requested by one customer into one quote.

A quote becomes the active quote when:

- The employee end user creates a new quote by clicking the New, Modify, Deactivate, Suspend or Resume buttons in the Installed Assets or Account Summary view.
- The employee end user navigates to the Quote Details view and clicks the Profile button.
- The customer end user goes to the Shopping Cart.

When an end user makes changes to the service profile of a different account, a different quote is made active. When a new quote is made active, the previously active quote becomes inactive by implication.

When a quote is converted to an order, the quote becomes inactive.

Note: If the employee end user clicks New in the Account Summary or Installed Assets view, and there is already an active quote for the same account, then the end user is taken to the existing active quote. An additional new quote is not created.

Active Order

An order becomes active only when the employee end user creates a new order by clicking the New, Modify, Deactivate, Suspend and Resume buttons in the Installed Assets or Account Summary View.

When an order is submitted, the order becomes inactive.

Quote-to-Sales Order

When a customer approves a quote and it becomes an order, an employee end user such as a CSR (Customer Service Representative) can automatically write the quote information into an order with the click of a button. When the end user clicks the Sales Order button or the Auto Order button, a workflow does the following:

- Generates a new sales order
- Converts all line items in a quote to order line items
- Explodes all quote line items, so that there multiple quantities of an item become individual line items

After a quote is converted to a sales order, the original quote still exists, and can be referenced. You will see the quote number in the order header.

Note: All orders created from quotes are considered *Sales Orders*. These orders may include items that are physical products and items that are service products. Sales Orders must not be confused with *Service Orders*. Service Orders are created as a result of a service request and are managed as part of Siebel Field Service.

Quote and Order Concepts in Asset-Based Ordering

This topic describes quote and order concepts that are specific to asset-based ordering.

Account and Contact Based Ordering

You can base a new order on the assets owned by either an account or a contact.

Orders that are based on accounts are created in the Accounts screen, Account Summary view.

Orders that are based on contacts are created in the Contacts screen, Contact Summary view.

New Quotes and Orders

A new quote or order identifies products and services that are first-time acquisitions for a customer or that are repeat orders such as additional software licenses.

After being processed and fulfilled, new orders result in new assets that become part of a customer's profile. A line item from an order becomes an installed asset if it is marked as a trackable asset in the product record.

Multiple Open Orders

Multiple orders may be in progress for a particular customer or even a particular installed asset at any given time.

Siebel asset-based ordering supports order stacking. Order stacking occurs when order lines are future-dated or lines take a long time to fulfill. If an order to change an asset has not yet been fulfilled and the customer calls to place another order to change the same asset, then the version of the asset changed during the second order includes the changes from the earlier order. It is vital to include changes from the earlier order so the application can ensure that the second change does not break the validation rules for the overall service and to ensure that the customer receives accurate pricing. If a customer has multiple pending orders for a single asset (for example, for the telephone service to a large office complex), then the application supports inserting a new order between the due dates of the two pending orders.

After an order has been placed, the end user can change it in the following ways:

- Create a modify order.
- Create a supplemental order.
- Create a follow-on order.

This feature is handled by two processes, Modify Order and Supplemental Order. Creating a supplemental order does not actually create an additional order; it revises an existing order.

Modify Quotes and Modify Orders

A modify quote or order is created when the customer asks to make a change to an existing product or service.

For example, the local telephone company provides local phone service to a residential customer. After the service has been installed, customers may want to make additions and changes to the services they receive, such as adding call waiting or changing the number to call when forwarding a call.

Each of these changes is a modification to the current asset. When the customer inquires about these changes, the sales representative issues a modify quote for the change. After the customer accepts the modify quote, it becomes a modify order.

When a modify order is placed, it does not modify the asset listed in the customer's profile, because it has not been fulfilled yet. The customer profile is only changed when the order is fulfilled.

Supplemental Order

A supplemental order is created by revising an open order. A supplemental order overrides an order that has already been sent to the fulfillment application, but is still in process and has not yet been fulfilled. Supplemental orders are useful when a customer wants to make further changes to an order while it is in transit—that is, before its status is Complete in the fulfillment application.

The following is an example of using a supplemental order.

A customer currently subscribes to a wireless plan and now wants to change the call-forwarding number associated with the plan. In the Account Summary view, Installed Assets list, the call center representative selects the plan, clicks Modify, and creates a quote for the change. With the customer's approval, the sales representative then converts the quote to an order and submits it. The order is then sent to a fulfillment application. There is now an open order for this item.

An hour later, the customer calls again and wants to back out the change. The call center representative again goes to the Account Summary. In the Orders list, the call center representative drills down into the open order and revises

the order. The representative selects the appropriate line item, makes a change during the configuration session that returns the call-forwarding number to the previous number, and clicks Done to add the change. The representative then submits the supplemental order. The supplemental order replaces the original order in the fulfillment application.

The order number of a supplemental order remains the same as the number of the original order, with a different version. Where supplemental orders exist for an original order, a user will see records of multiple order instances with the same order number. The most recent order instance is displayed at the start of the order list.

The relationships between original and supplemental orders are as follows:

- A revised order refers to the original order by its Order Number, which is the same as that of the original order.
- All that changes in the original order is the value in the Active field, which is set to N. This change makes the order and its child records read-only.
- Each line item of a revised order refers to the corresponding original order line item by its Revised Line Item Id.
- In the first order line item of a product's order revision history, Id = Revised Line Item Id.
- In each subsequent revision, the Revised Line Item Id is copied to the new version of the order line item.

Note: Some fulfillment applications do not support supplemental orders. Others only support a limited set of allowed changes. To support the widest possible set of fulfillment applications, your Siebel Business Application supports these types of changes to an order: Add, Delete, Update, and No Change.

Follow-On Order

A *follow-on order* is a future-dated change to an asset before the asset is fulfilled. It can be created at any time.

Follow-on orders allow you to make a change after an order has reached the point of no return, so it is not possible to create a supplemental order. You can create a follow-on order for a customer, which allows the customer to place a new order for the same service using the original order as a starting point.

Note: Contact-based ABO does not support Follow-On Orders.

Action Codes

The following action codes for line items on quotes and orders help to keep track of changes made through modify orders and supplemental orders:

- **Add.** Instruction to add a new product to the customer assets.
- **Update.** Instruction to update either an attribute value or delta field value for an existing product (either asset or ordered)
- **Delete.** Instruction to delete or deactivate or disconnect an existing product (either asset or ordered).
- **- .** Do not change an existing product.
- **Suspend.** Instruction to update the status of an asset to *Suspended*.
- **Resume.** Instruction to update the status of an asset from *Suspended* to *Active*.

These action codes appear next to the line items on quotes and orders to show which items are affected, and which will not.

Action codes are extensible using the SIS OM PMT user properties.

Auto Asset Button

You might see an Auto Asset button on the Orders screen. This button is used for demonstration purposes, to show how order line items can be converted into assets that display in the Installed Assets view of the customer's service profile.

In actual deployments, a workflow is used to create assets automatically from completed order line items, so this button is not used.

Ungrouping or Exploding Quote Line Items

When a quote is converted to an order, line items with a quantity greater than one can be *ungrouped* to become multiple items, each with a quantity of one. This automatic ungrouping is referred to as *exploding* the items. Each line item becomes an individual asset when the order is fulfilled.

When this feature is turned on, you can automate the ungrouping of multiple quantities of products that will become assets, so that each can be given a unique integration ID.

> **Note:** Ungrouping or exploding items is only required when the items will become trackable assets, and therefore require integration IDs. Items that will not become assets may remain as an order for a multiple quantity, without being ungrouped.

The Ungroup feature can simplify the process of creating bulk orders, as shown in the following example:

- The customer wants to order ten local lines, with similar configurations: five will have voice mail, five will have call waiting. The customer wants each of the ten lines to have the same four additional features.
- An employee end user can start by creating one line item—with the four features that are desired on all of them—and the quantity two. Then, the end user ungroups that item, so that there are two items.
- Next, the end user takes one of those two items, adds voice mail to it, and changes the quantity to five. Then, the user takes the second line item, adds call waiting to it, and changes the quantity of it to five. Finally, the user ungroups both line items, so that there are ten individual line items.
- Now each line item has an integration ID and a field for Service ID. Depending on the implementation of the Siebel Business Applications with the back end applications, the end user might click a button to populate the field with phone numbers from the asset management application, or a phone number might be assigned as part of the provisioning process.

Scenario for Asset-Based Ordering

This topic gives one example of how asset-based ordering might be used. You might use asset-based ordering differently, depending on your business model.

During a typical day in a customer service organization, a group of CSRs responds to customer phone calls to generate quotes and sales orders.

If the caller is new, then the CSR creates an account and adds the caller to the database. If the caller is an existing contact, then the CSR searches the database to display the contact's record.

If the conversation leads to a potential sale, then the CSR will create a quote and fax, email, or mail it to the caller. The caller can use the quote to get approval to place the order.

The information collected during the call is permanently stored, associated with the contact's record.

When the caller contacts the CSR to place the order, the CSR reviews the quote and converts the quote to an order. Any changes to the product and pricing definitions since the quote was saved are displayed in the quote the CSR views.

The CSR enters any additional information required for the order and submits the order. A new Order record is created. After the order has been fulfilled, trackable items become assets, and appear in the Account Summary view for this customer.

At a later date, the customer may contact the CSR to add or delete items from the order. The order may be pending or in another status, such as complete. In either case, the CSR can generate a quote and order to change the initial order.

Creating a Quote Using Asset-Based Ordering

End users can use the following procedure to create a quote for an existing customer purchasing a new service or product.

You can create a quote that is based on the assets of either an account or of a contact. This procedure shows how to create a quote for an account. Use a similar procedure to create a quote for a contact, but begin by navigating to the Contact screen, Contacts List view.

When you use this procedure, the account type determines whether a quote or order is created. If the account type is *Residential*, then quotes are not created and all modifications of the service profile create an order directly. For all other account types, a quote is created.

For contact-based ABO, the user creates orders directly and does not create quotes.

> **Note:** The following procedure shows how to create a quote for an asset owned by an account. You can also create a quote for an asset owned by a contact using a similar procedure, but navigating to the Contacts screen, and then the Contact Summary view.

To create a quote

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Installed Asset list, click New.
   The Quote form and Line Item list appear.

   > **Note:** If there is already an active quote for this account, then it is shown. If there is not already an active quote, then a blank new quote appears.

   In the Quote form, the Status field displays In Progress and the Quote # field displays a unique number generated by the application. The Quote Name field also displays this unique number, which you may change to a quote name.
5. In the Line Items list, click Add Items.
6. In the Pick Products dialog box, select the items that the customer wants, and click OK.
   The Action field for the line items you added displays Add. For products and services that are customizable, the Customize button is available.
7. To customize an item, select it in the Line Items list and click Customize.

A Configurator session starts and displays selection pages.

8. Make the selections that the customer wants, and click Done.

The Quote Line Items list appears with the quote.

Some fields in the Line Items list, and the Totals form after it, are described in the following table:

Field	Comments
Item Net Price	The price of the line item after discounts from pricing rules, volume discounts, or manual discounts. (Read-only)
Extended Price	Quantity times the start price.
Net Discount%	The percentage of the start price that the discount equals. If a discount was entered in either Discount Amount or Discount Price, then this percentage is calculated using the discount entered and the start price. (Read-only)
Extended Net Price	Quantity times the unit net price. (Read-only)
Discount Amount	The amount to be discounted for each unit, as a dollar figure.
Discount%	The discount amount relative to the unit price, shown as a percentage.
Discount Price	The total amount to be charged for the item, for each unit.
Non-Discount Extended Price	The extended start price.
Sub-Total	The sum of all items in a product with components with a one-time price type.
MRC Subtotal	Monthly recurring charge subtotal. The sum of all items in a product with components that are charged by the month.
Action	Action required for processing a modify order. Choices are add, update, delete, suspend, resume, or - (dash). A dash means the item was not changed, though the associated child line items may change.
Agreement Name	Agreement under which the quote line item was made.
Adjustment MRC	Total adjustments for monthly recurring charges.

Field	Comments
Adjustment NRC	Total adjustments for non-recurring charges.
Item Total MRC	Total charges for monthly recurring charges.
Item Total NRC	Total charges for non-recurring charges.
Net Total MRC	Net total for monthly recurring charges.
Net Total NRC	Net total for non-recurring charges.
Total Item Discount MRC	Total discounts for monthly recurring charges.
Total Item Discount NRC	Total discounts for non-recurring charges.

Creating a Quote That Modifies an Asset

Employee end users can use the following procedure to create a quote that is based on an asset. For example, a customer has five phone lines installed as part of a small business service product. The customer calls and wants to add more phone lines. You would use the existing installed asset as the basis for a new quote that adds more phone lines.

When you use this procedure, the account type determines whether a quote or order is created. If the account type is *Residential*, then quotes are not created and all modifications of the service profile create an order directly. For all other account types, a quote is created.

Note: The following procedure shows you how to create a modify quote for an asset owned by an account. In contact-based ABO, the user does not create modify quotes but only modify orders.

To modify an installed asset

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Installed Asset list, select the installed asset you want to modify.
5. Click Modify.

   Siebel Configurator displays the current configuration of the item, adjusted for any open orders.
6. You can multi-select customizable products from Installed Assets or Billing Items applets and Click Modify to open the view to configure multiple customizable products explorer view. If the asset is bundle promotion, Modify opens the view to configure multiple customizable products.

7. In the Configurator selection pages, modify the configuration by adding or deleting items or by changing item attribute values.
8. When you are finished modifying the configuration, click Done in the selection pages.

   The Quote form appears and the changes you made are shown in the Line Items list.
9. In the Line Items list, verify that items display correctly and have the correct action codes.

   The configuration displayed in the quote is a composite of the asset as it is currently installed and adjustments to the asset requested in open orders.
10. Click Profile to return to the Account Summary view.

Ungrouping Quote or Order Line Items

If the product's Auto Explode flag is set, then when a quote is turned into an order, line items with a quantity of more than one are *ungrouped* to become separate instances, each with a quantity of one.

Ungrouping only applies to asset-based ordering and to assets for which the Auto Explode flag is set. If the end user enters quantities greater than one, then the quote-to-order process automatically ungroups them.

Ungrouping is needed to give each asset separate a Service ID, so it can be tracked and serviced in the future. It also gives each asset a separate integration ID, which is used by the back-end provisioning application for fulfillment.

The ungroup feature allows you to configure once for multiple items and then splitting them into individual items that can have their own variant configurations, telephone numbers, or Billing/ Service accounts, as described in *Ungrouping or Exploding Quote Line Items*.

This procedure shows how to ungroup line items in a quote for an asset owned by an account. For contact-based ABO, ungroup is carried out in the order itself, not in the quote.

> **Note:** This procedure shows how to ungroup quote line items. Use a similar procedure to ungroup order line items, but drill down on the Orders list rather than the Quotes list.

To ungroup quote line items

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Account List, drill down on the account name.
3. Click the Account Summary view tab.
4. In the Quotes list, drill down on the desired quote number.

   The Quote view appears.
5. Click the Line Items view tab.

   The line items in the quote appear.
6. Select the line item you want to ungroup.

   This line item must have a quantity greater than one, with items that need to be configured differently. The line item must have an Action Code of Add; you cannot ungroup existing Assets.
7. From the Line Items menu, select Ungroup.

   The quote ungroups the item and displays it as multiple line items, with a quantity of one for each item.

Editing Quote Line Items Using Asset-Based Ordering

End users may need to edit line items when creating a new quote, or when returning to an existing quote.

Use the following procedure to do one of the following:

- Edit line item details for a simple product.
- Configure or reconfigure a product with components in a quote an end user is creating in the Quotes screen. If a customer asset exists, then Configurator selection pages display the asset adjusted for any open or pending orders.
- Edit the configuration of a product with components in a quote created by modifying an asset or deactivating an installed asset.

Note: The following procedure shows how to edit quote line items for an asset owned by an account. In contact-based ABO, the user does not edit quote line items, only order line items.

To edit quote line items

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Quotes list, click the desired Quote Number.

   The Quote form appears, displaying details of the quote.
5. In the Line Items list, select the line item that you want to edit and change the values of the fields directly in the line item.

   The action code of the line item will be changed to Update.

   To retain manual discounts, click the Keep Discount check box in the Line Item Detail form.
6. To customize an item, select a customizable product in the Line Items list.

   The Customize button becomes active.
7. Click Customize.

   A Configurator session starts and displays selection pages, which the current configuration of the customer asset adjusted for any open orders.
8. Modify the configuration of the customizable product or service as needed and click Done.

   The revised configuration displays in Line Items.

Converting Quotes to Orders Using Asset-Based Ordering

After a customer agrees to the details of a quote, the employee end user must complete the steps in the following procedure to convert a quote into an order.

> **Note:** The following procedure shows how to convert quotes to orders for an asset owned by an account. It is not applicable to contact-based ABO, where the user creates orders directly rather than creating quotes and converting them to orders.

To convert a quote to an order

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Quotes list, drill down on the desired Quote Number.

   The Quote form appears, displaying details of the quote.
5. Click the Orders view tab.
6. In the Orders list, click Sales Order.

   The quote is converted to an order and appears in the Orders list.

The following Business Component fields are transferred from the quote to the order when the user click Sales Order, but some of the fields are not displayed in the UI:

- **To Order.** Account Id, Discount, Discount Amount, Discount Amount MRC, Discount Reason, Contact Id, Currency Code, Price List Id, Order Type, Quote Id, Agreement Id.
- **To Order Item.** Base Price, Asset Id, Service Id, Discount Amount, Discount Amount MRC, Discount Percent, Discount Percent MRC, Discount Source, Volume Discount Id, Volume Discount Item Id, Unit Price, MRC CxTotal, NRC CxTotal, Due Date, Agreement Id, Quantity Requested, Product Id, Action Code, List Price, Asset Integration Id.
- **To Order Item Attribute.** Name, Required, Sequence, Unit of Measure, Validation, Description, Action Code, Data Type, Value, Attribute Id, Display Name, Hidden, LOV Type, Read Only.

Editing Order Line Items Using Asset-Based Ordering

Employee end users can edit the configuration of customizable products in pending orders by using the following procedure. A customer asset does not need not be associated with the order.

If the order an end user is changing is a modification to a customer asset, then all open orders are applied to the asset before it is displayed in the Configurator selection pages.

Note: The following procedure shows how to edit order line items for an asset owned by an account. You can also edit order line items for an asset owned by a contact using a similar procedure, but navigating to the Contacts screen, and then the Contact Summary view.

To edit order line items

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Orders view tab.
4. In the Orders list, drill down on the desired Order #.

   The Sales Order form and Line Items list appear, displaying details of the order.
5. In the Line Items list, select the line item you want to modify.
6. In the order's Status field, make sure the status is Open or Pending.

   If the status is Complete, then the order has been fulfilled. If the customer wants to make a change to a completed order, then you must create a new order or revise an existing order. To create a new order, which you must do when modifying an installed customer asset, see *Modifying the Service Profile*. To revise an existing order, see *Revising a Submitted Order Using Asset-Based Ordering*.
7. Select the line item that you want to edit and change the values of the fields directly in the line item.

   The action code of the line item will is changed Update.
8. To change the configuration of an item, select the line item you want to edit and click Customize.

   A Configurator session starts and displays selection pages.
9. Modify the configuration of the product or service as needed, and click Done.

   The revised configuration appears in the Line Items list. The Revision field displays the version of the order, which is incremented when the order is revised to create a supplemental order.

Submitting an Order Using Asset-Based Ordering

Employee end users who are using asset-based ordering can use the following procedure to submit an order.

Note: The following procedure shows how to submit an order for an asset owned by an account. You can also submit an order for an asset owned by a contact using a similar procedure, but navigating to the Contacts screen, and then the Contact Summary view.

To submit an order

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Orders list, drill down on the desired Order #.

   The Sales Order form and Line Items list appear displaying details of the order.
5. In the Sales Order form, click Submit.

The Status changes to Open for the order and for any line items that were previously pending.

Revising a Submitted Order Using Asset-Based Ordering

Employee end users may need to revise an order that has been submitted. It must be revised before it can be changed and resubmitted. This revision is done using a supplemental order. The following procedure describes how end users can revise open orders.

Note: The following procedure shows how to revise a submitted order for an asset owned by an account. You can also revise a submitted order for an asset owned by a contact using a similar procedure, but navigating to the Contacts screen, and then the Contact Summary view.

To revise a submitted order

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Orders list, drill down on the desired order number of the open order.

   The Sales Order form and Line Items list appear, displaying details of the order.
5. In the Sales Order form, click Revise.
6. In the Line Items list, select the line item you want to modify.
7. To change the configuration of a line item, select the item and click Customize.

   A Configurator session starts and displays selection pages. The selection pages display the current configuration of the customer asset adjusted for any open orders.
8. Modify the configuration of the customizable product or service as needed and click Done.
9. In the Sales Order form, click Submit.

   The Status changes to Open for the order and any line items that were previously pending.

The integrator is responsible for extending the Submit Order Process workflow to do both of the following:

- Compare the before and after versions of the order.

Interact with the service fulfillment application to cancel, update, roll back, or accelerate the order line items.

Original Action	Status	New Action	Effect
- (Dash)	All	- (Dash)	No effect
- (Dash)	All	Update	Updates installed asset
- (Dash)	All	Delete	Deletes installed asset

Original Action	Status	New Action	Effect
- (Dash)	All	Add	No effect
Add	Complete	- (Dash)	No effect
Add	Complete	Update	Updates installed asset
Add	Complete	Delete	Deletes installed asset
Add	Complete	Add	No effect
Add	Complete	- (Dash)	No effect
Add	Complete	Update	No effect
Add	Complete	Delete	No effect
Add	Complete	Add	Adds installed asset
Update	All	- (Dash)	No effect
Update	All	Update	Updates installed asset
Update	All	Delete	Deletes installed asset
Update	All	Add	No effect
Delete	Complete	- (Dash)	No effect
Delete	Complete	Update	No effect
Delete	Complete	Delete	No effect
Delete	Complete	Add	Adds installed asset with same configuration
Delete	Complete	- (Dash)	No effect
Delete	Complete	Update	Updates installed asset
Delete	Complete	Delete	Deletes installed asset
Delete	Complete	Add	No effect

Original Action	Status	New Action	Effect

Creating a Follow-On Order

An order passes the point of no return when the company's business policies dictate that a customer can no longer change the order because it would be too expensive. For example, if the service is fully installed and is now being tested, then it is at the point of no return.

When an order has reached the point of no return, so it is not possible to create a supplemental order, you can create a follow-on order for a customer, which allows the customer to place a new order for the same service using the original order as a starting point.

A follow-on order is a future-dated change to an asset before the asset is fulfilled, which can be created at any time. This procedure shows one common use of follow-on orders.

To create a follow-on order

1. Navigate to the Orders screen.
2. In the Orders list, query for the order for which you want to create a follow on order.
3. Drill down on the Order # link to display the Order screen, Line Items view.
4. Select the line item for the service that you want to change and click Modify.

   When you click Modify, the application applies all open orders against the asset, in chronological order, before presenting the asset to the user.
5. If the service is customizable, then the Configurator view appears. Make the changes to the service and click Done.

   Depending on the type of Account (Residential or Business) either the Quote Detail view or the Order Line Items view will be displayed with the follow-on order.

Modifying the Service Profile

With asset-based ordering, employee end users use the Account Summary or Contact Summary view to access the service profile and to work with quotes and orders. End users can also access an account's service profile in the Billing Item and Installed Asset views.

Note: There is an Accounts view labeled Service Profile, but it is used primarily in Field Service. It does not display the customer's service profile that is used in asset-based ordering.

There are several ways to modify the service profile of a customer:

- Create a quote for a new item. See *Creating a Quote Using Asset-Based Ordering*.
- Create a quote that modifies an existing item. See *Creating a Quote That Modifies an Asset*.

- Create a quote to deactivate an item. See *Deactivating an Installed Asset*.

If a quote is generated by any of these methods, then the service profile is updated when the quote and subsequent order are fulfilled. This update is typically done in one of two ways:

- From an external order management application. For more information on how this integration is done, see *Workflow to Apply an Order to the Service Profile*.
- From an external application that maintains service profiles. In this case, Siebel EIM sends a replication of the customer's profile to the Siebel Business Application. For more information about EIM, see *Siebel Enterprise Integration Manager Administration Guide* .

Without configuration, an Auto-Asset button can be used to update the service profile from a completed Order. However, when one of these two methods is implemented, the integrator usually removes the Auto-Asset button. If the Auto-Asset button has not been removed, then the end user can use the following procedure.

To modify the service profile using Auto-Asset

1. Navigate to the Sales Orders screen, and then the List view.
2. In the Orders list, drill down on the desired order number of an order that has been submitted.
3. In the Line Items list, set the status of the line items to Complete.
4. Click Auto-Asset.

   Note: Auto-Asset must be used once for each root level product in the Order.

5. Navigate to the Account Summary view for the Service Account associated with the line items.

   The completed order line items is reflected in the service profile.

Deactivating an Installed Asset

With asset-based ordering, employee end users can use the following procedure to create a quote that discontinues an existing customer service.

When you use this procedure, the account type determines whether a quote or order is created. If the account type is *Residential*, then quotes are not created and all modifications of the service profile create an order directly. For all other account types, a quote is created.

Note: The following procedure shows how to deactivate an installed asset owned by an account. You can also deactivate an installed asset owned by a contact using a similar procedure, but navigating to the Contacts screen, and then the Contact Summary view.

To deactivate a installed asset

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the appropriate account.
3. Click the Account Summary view tab.
4. In the Installed Assets list or the Billing Items list, select the installed asset you want to deactivate, and click Deactivate.

   The Quotes form appears. The Line Items list shows the items that will be deactivated.

5. Verify that the installed assets you want to deactivate display an action code of Delete.
6. Convert the quote to an order and submit it. For more information, see *Converting Quotes to Orders Using Asset-Based Ordering*.

Suspending and Resuming Service

When you create a suspend order, you stop service. The customer is still the owner of the service, but no service is being delivered. When you create a resume order, you recommence a service that has been suspended.

You may use suspend and resume orders for a number of reasons:

- Customers who are going on vacation may want to suspend their newspaper delivery service while they are away.
- A chain store may have one branch in a remote location that closes for the winter, so it wants to suspend both its phone and internet service to that site for the winter.
- A phone company may have a policy saying that, if customers have not paid their bills for three months and do not respond to calls from collection agents, then the phone company suspends their service. The phone company resumes service after the bills are paid.
- Customers may report that they have lost their cell phones, and the phone company suspends service to prevent fraud. When the phones are found, the company resumes the service.

Note: The following procedures show how to suspend and resume service for an asset owned by an account. You can also suspend and resume service for an asset owned by a contact using similar procedures, but navigating to the Contacts screen, and then the Contact Summary view.

To enter a suspend order

1. Navigate to the Accounts screen, and then the Accounts List view.
2. In the Accounts list, click the name of the account that has the service.
3. Click the Account Summary Assets view tab.
4. In the Installed Assets list, if necessary, do a query to find the assets to be suspended.

   For example, if you want to suspend service to a site, then query on the Point of Service field to find all assets for that site.
5. In the Installed Assets list, select one or more services with the status of Active.
6. From the Installed Assets menu, select Suspend.

Use the following procedure to enter a resume order.

To enter a resume order

1. Navigate to the Installed Assets view.
2. In the Installed Assets list, select a service that has the status of Suspended.
3. From the Installed Assets menu, select Resume.

18 Workflow and Method Reference for Order Management Interface

Workflow and Method Reference for Order Management Interface

Some features of the order management interface are based on Siebel workflows. You can modify these workflows to suit your own business model using Siebel Business Process Designer. For more information, see *Siebel Business Process Framework: Workflow Guide* .

This chapter explains the Siebel workflows used by the order management interface, and the methods called by these workflows. This chapter does not include workflows for asset-based ordering, which are covered in *Workflow Reference for Employee Asset-Based Ordering*.

This chapter covers the following topics:

- *Order Management Interface Workflow Reference*
- *Order Management Interface Methods Reference*

Order Management Interface Workflow Reference

The order management interface uses the following workflows:

- Workflows for creating a quote or order:
  - *Account - New Order Workflow*
  - *Account - New Quote Workflow*
  - *Contact - New Order Workflow*
  - *Contact - New Quote Workflow*
  - *Goto_Quote Workflow*
  - *Goto_Order Workflow*
- Workflows for validating a quote, order, or agreement:
  - *Verify Header (Quote) Workflow*
  - *ISS Validation (Agreement) Workflow*
  - *ISS Validation (Order) Workflow*
  - *ISS Validation (Quote) Workflow*
- Workflows for approving a quote, order, or agreement:
  - *ISS Approval (Quote) Workflow*

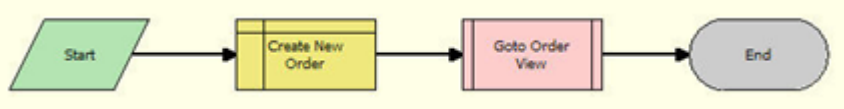
- *ISS Approval (Order) Workflow*
- *ISS Approval (Agreement) Workflow*
- *ISS Post Approval Workflow (Quote) Workflow*
- *ISS Post Approval Workflow (Order) Workflow*
- *ISS Post Approval Workflow (Agreement) Workflow*

Note: The functionality of these workflows may depend on the exact setting of the arguments for the methods called. For detailed information about the argument settings for the methods called by workflow processes, refer to the workflow process definition in *Siebel Tools Online Help* .

Account - New Order Workflow

Account - New Order Workflow, shown in the following figure, creates a new Quote for the current Account. It then navigates to Order Entry - Detail Orders View.

It is triggered by the Accounts Screen, Account List applet menu.



Workflow Description

This workflow does the following:

1. **Create New Order.** Creates an order
2. **Goto Order View.** Calls a Sub-process called Goto_Order, which navigates to Order Entry - Detail Orders View

Account - New Quote Workflow

Account - New Quote Workflow, shown in the following figure. creates a new Quote for the current Account. It then navigates to Quote Item Detail View.

It is triggered from the Accounts screen, Account List applet menu.

Workflow Description

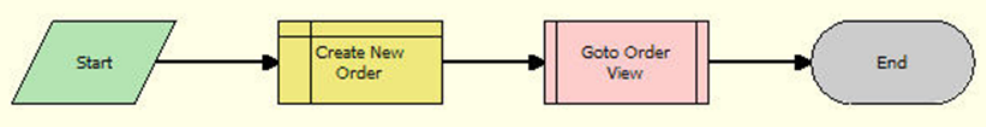
This workflow does the following:

1. **Create New Quote.** Creates a quote
2. **Goto Order View.** Calls a sub-process called Goto_Quote, which uses Interactive workflow step to navigate to Quote Item Detail View.

Contact - New Order Workflow

Contact - New Order Workflow, shown in the following figure, creates a new Order for the current Contact. It then navigates to the Order Entry screen, Detail Orders View.

It is triggered from the Contacts screen, Contact List applet menu.



Workflow Description

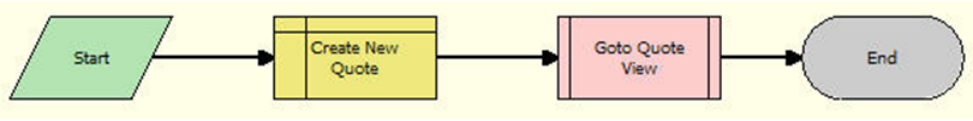
This workflow does the following:

1. **Create New Order.** Creates an order for a contact.
2. **Goto Order View.** Calls a Sub-process called Goto_Order, which navigates to Order Entry - Detail Orders View

Contact - New Quote Workflow

Account - New Quote Workflow, shown in the following figure, creates a new Quote for the current Account. It then navigates to the Quote Item Detail View.

It is triggered from the Contacts Screen, Contact List applet menu.



Workflow Description

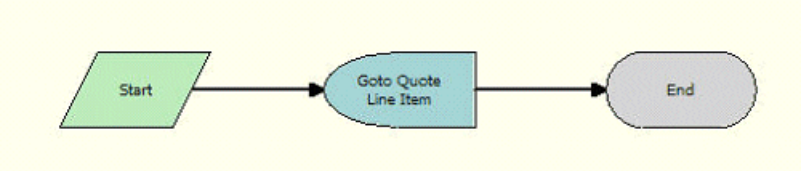
This workflow does the following:

1. **Create New Quote.** Creates a quote
2. **Goto Quote View.** Calls a sub-process called Goto_Quote, which uses interactive workflow step to navigate to Quote Item Detail View.

Goto_Quote Workflow

Goto_Quote workflow, shown in the following figure, navigates to Quote Item Detail View

It is triggered by the Account - New Quote and Contact - New Quote workflows.



Workflow Description

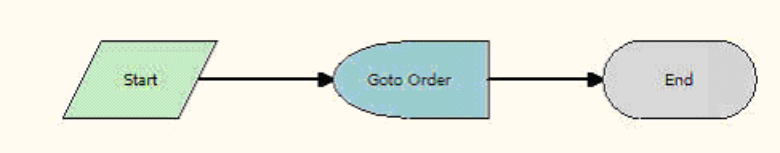
This workflow does the following:

This sub-process uses an interactive workflow step to navigate to Quote Item Detail View.

Goto_Order Workflow

Goto_Order workflow, shown in the following figure, navigates to Order Entry - Detail Orders View.

It is triggered by the Account - New Order and Contact - New Order workflows.



Workflow Description

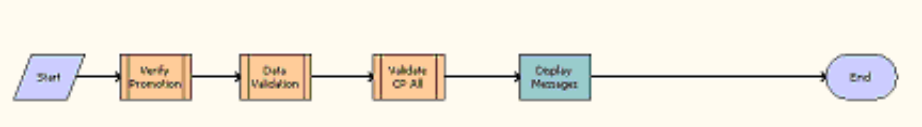
This workflow does the following:

This sub-process uses an interactive workflow step to navigate to Order Entry - Detail Orders View.

Verify Header (Quote) Workflow

Verify Header (Quote) workflow, shown in the following figure, is used to verify a quote, verify a promotion, and verify line items of a quote. It displays promotion violations, data validation rule violations, and complex product violations.

It is triggered when the user clicks the Verify button in a quote header or selects Verify from the menu of the quote line item applet.



Workflow Description

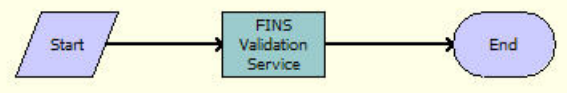
This workflow does the following:

1. **Verify Promotion.** Calls the ISS Promotion Verify SubProcess workflow to verify promotion status.
2. **Data Validation.** Calls the ISS Validation (Quote) workflow to verify Data Validation Rules.
3. **Validate CP All.** Calls the VORD Validate Complex Product All (Quote) workflow to validate all complex products under the Quote.
4. **Display Messages.** Displays all messages generated by the earlier steps.

ISS Validation (Agreement) Workflow

ISS Validation (Agreement) workflow, shown in the following figure, validates the Agreement according to the Rule Set definition for the agreement in Administration - Data Validation.

It is triggered by the VerifyItem signal, which is called by the Agreement screen, Verify applet menu.



Workflow Description

This workflow does the following:

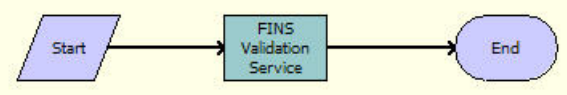
This workflow calls the Validate method of the Data Validation Manager Business Service. This workflow validates the Agreement according to the Rule Sets that are defined in Administration - Data Validation view. For more information, see *Validate Method*.

Name	Business Object	Data Type	Comments
Error Code	Service Agreement	String	The Error Code that is returned if an error occurs.
Error Message	Service Agreement	String	The Error Message that is returned if an error occurs.
Object Id	Service Agreement	String	The Row ID of the current object.
Process Instance Id	Service Agreement	String	The Row ID of the current workflow process.
Return Code	Service Agreement	String	The Return Code that is defined in the Rule Set.
Return Message	Service Agreement	String	The Return Message that is defined in the Rule Set.
Siebel Operation Object Id	Service Agreement	String	The Row ID of the Siebel Operation Object.
Store Messages	Service Agreement	String	Variable to determine if the validation messages must be stored in the validation tables.
Validation Ruleset Name	Service Agreement	String	The name of the Rule Set that is defined in the Data Validation Administration screen.

ISS Validation (Order) Workflow

ISS Validation (Order) workflow, shown in the following figure, validates the Order according to the Rule Set definition in Administration - Data Validation view for orders.

It is triggered by the VerifyItem signal, which is called by the Verify applet menu item of the Order Header menu.



Workflow Description

This workflow does the following:

This workflow calls the Validate method of the Data Validation Manager Business Service. This workflow validates the Order according to the Rule Sets that are defined in Administration - Data Validation. For more information, see *Validate Method*.

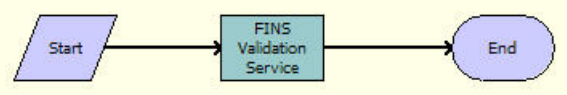
Process Properties

Name	Business Object	Data Type	Comments
Error Code	Order Entry	String	The Error Code that is returned if an error occurs.
Error Message	Order Entry	String	The Error Message that is returned if an error occurs.
Object Id	Order Entry	String	The Row ID of the current object.
Process Instance Id	Order Entry	String	The Row ID of the current workflow process.
Return Code	Order Entry	String	The Return Code that is defined in the Rule Set.
Return Message	Order Entry	String	The Return Message that is defined in the Rule Set.
Siebel Operation Object Id	Order Entry	String	The Row ID of the Siebel Operation Object.
Store Messages	Order Entry	String	Variable to determine if the validation messages must be stored in the validation tables.
Validation Ruleset Name	Order Entry	String	The name of the Rule Set that is defined in the Data Validation Administration screen.

ISS Validation (Quote) Workflow

ISS Validation (Quote) workflow, shown in the following figure, validates the Quote according to the Rule Set definition for quotes in Administration - Data Validation.

It is triggered from VerifyItem signal which is called by the Verify Applet Menu item on the Quote Header.



Workflow Description

This workflow does the following:

This workflow calls the Validate method of the Data Validation Manager Business Service. This workflow validates the Quote according to the Rule Sets that are defined in Administration - Data Validation. For more information, see *Validate Method*.

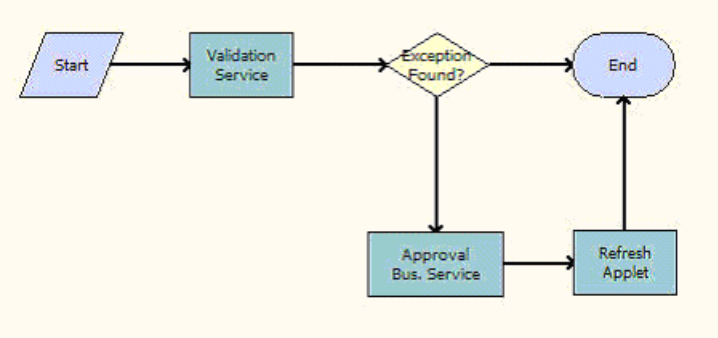
Process Properties

Name	Business Object	Data Type	Comments
Error Code	Quote	String	The Error Code that is returned if an error occurs.
Error Message	Quote	String	The Error Message that is returned if an error occurs.
Object Id	Quote	String	The Row ID of the current object.
Process Instance Id	Quote	String	The Row ID of the current workflow process.
Return Code	Quote	String	The Return Code that is defined in the Rule Set.
Return Message	Quote	String	The Return Message that is defined in the Rule Set.
Siebel Operation Object Id	Quote	String	The Row ID of the Siebel Operation Object.
Store Messages	Quote	String	Variable to determine if the validation messages must be stored in the validation tables.
Validation Ruleset Name	Quote	String	The name of the Rule Set that is defined in the Data Validation Administration screen.

ISS Approval (Quote) Workflow

ISS Approval (Quote) workflow, shown in the following figure, validates approval rules and creates approval items if necessary.

It is triggered when the user selects the Generate Approvals menu item on the Quote Header applet menu. This menu item raises the ApproveItem Signal, which calls this workflow.



Workflow Description

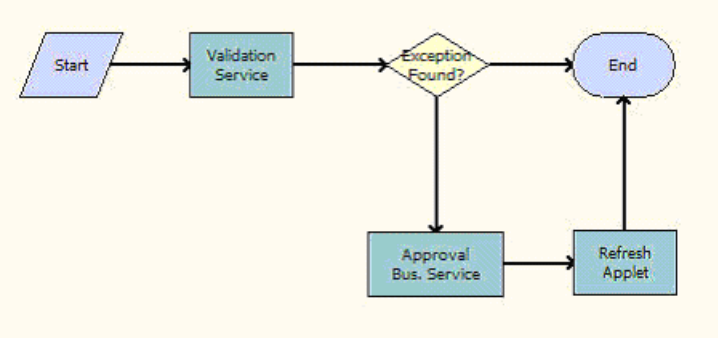
This workflow does the following:

1. **Validation Service.** Calls the Validate method of the Data Validation Manager Business Service, which validates the Quote according to the Rule Sets that are defined. For more information, see *Validate Method*.
2. **Exception Found.** If the validation service returns a null, then the workflow ends.
3. **Approval Business Service.** If it returns a non-null value, then the workflow calls CreateNewApprovalTasks method, which creates a new approval task for the quote. For more information, see *CreateNewApprovalTasks Method*.
4. **Refresh Applet.** Calls the Refresh Business Component method of the SIS OM PMT Service Business Service, which refreshes the Quote, so the approval items appear in the approvals view.

ISS Approval (Order) Workflow

ISS Approval (Order) workflow, shown in the following figure, validates approval rules and creates approval items if necessary.

It is triggered when the user selects the Generate Approvals menu item on the Order Header applet menu. This menu item raises the ApproveItem Signal, which calls this workflow.



Workflow Description

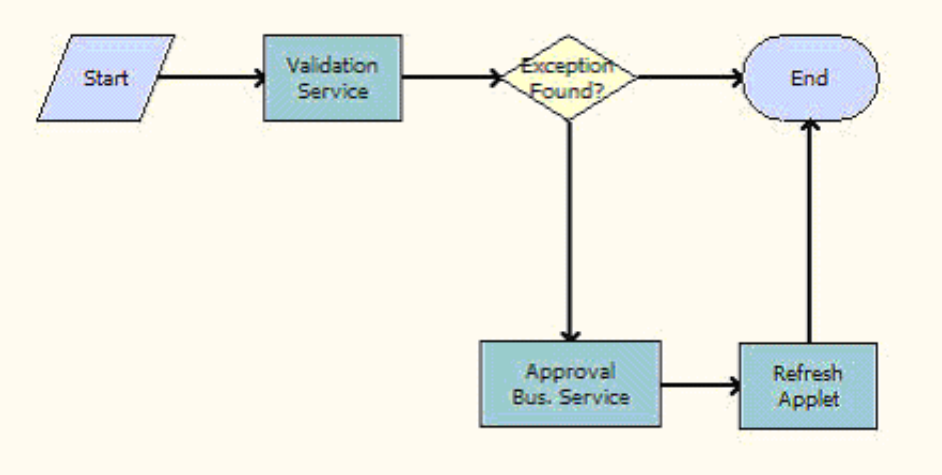
This workflow does the following:

1. **Validate.** Calls the Validate method of the Data Validation Manager Business Service, which validates the Order according to the Rule Sets that are defined. For more information, see *Validate Method*.
2. **Exception Found.** If the validation service returns a null, then the workflow ends.
3. **Approval Bus Service.** If it returns a non-null value, then the workflow calls the CreateNewApprovalTasks method of the ISS Approval Bus Service, which creates a new approval task for the Order. For more information, see *CreateNewApprovalTasks Method*.
4. **Refresh Applet.** Calls the Refresh Business Component method of the SIS OM PMT Service Business Service, which refreshes the Order, so the approval items appear in the approvals view.

ISS Approval (Agreement) Workflow

ISS Approval (Agreement) workflow, shown in the following figure, validates approval rules and create approval items if necessary.

It is triggered when the user selects the Generate Approvals menu item on the Agreement Header applet menu. This menu item raises the ApproveItem Signal, which calls this workflow.



Workflow Description

This workflow does the following:

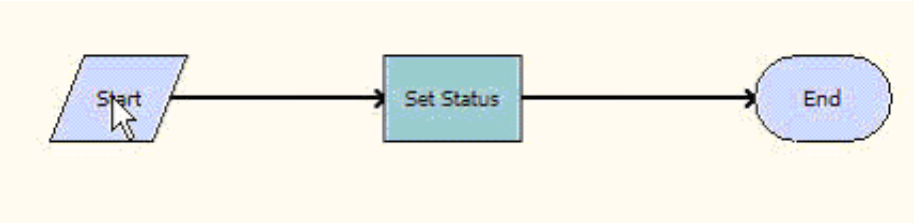
Validation Service. Calls the Validate method of the Data Validation Manager Business Service. For more information, see *Validate Method*.

1. **Exception Found.** If the validation service returns a null, then the workflow ends.
2. **Approval Bus Service.** If it returns a non-null value, then the workflow calls the CreateNewApprovalTasks method of the ISS Approval Bus Service. For more information, see *CreateNewApprovalTasks Method*.
3. **Refresh Applet.** Calls the Refresh Business Component method of the SIS OM PMT Service Business Service, which refreshes the Agreement, so the approval items appear in the Approvals view.

ISS Post Approval Workflow (Quote) Workflow

ISS Post Approval Workflow (Quote) workflow, shown in the following figure, checks whether any additional approvals are needed.

It is triggered when a user approves an Quote to see if additional approvals are needed.



Workflow Description

This workflow does the following:

Calls the SetApprovalDecision method. This will set the status of the current approval item and check to see if additional approval is needed.

Process Properties

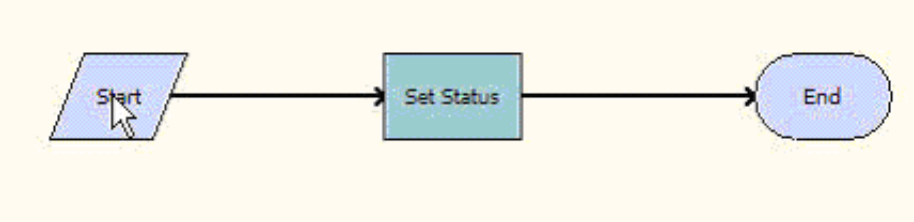
Name	Business Object	Data Type	Comments
ActionLIC	Quote	String	The Language Independent Code for the Action Field
Approval Level Name	Quote	String	The name of the Approval Level that is defined in the Approval Administration screen.
Error Code	Quote	String	The Error Code that is returned if an error occurs.
Error Message	Quote	String	The Error Message that is returned if an error occurs.
InboxItemId	Quote	String	The Row ID of the current Universal Inbox Item.
InboxTypeName	Quote	String	The Universal Inbox Type.

Name	Business Object	Data Type	Comments
Object Id	Quote	String	Obsolete.
ObjectId	Quote	String	The Row ID of the current object.
OwnerInfold	Quote	String	The Row ID of the approver.
Process Instance Id	Quote	String	The Row ID of the current workflow process.
Returned Status	Quote	String	The Status of the Approval Item.
SendEmail	Quote	String	Variable to determine if email is sent to the person who needs to approve the current item.
SeqNum	Quote	String	The Sequence Number for the current approval level.
Siebel Operation Object Id	Quote	String	The Row ID of the Siebel Operation Object.
StageId	Quote	String	The Row ID of the current Approval Stage.

ISS Post Approval Workflow (Order) Workflow

ISS Post Approval Workflow (Order) workflow, shown in the following figure, checks whether any additional approvals are needed.

It is triggered when a user approves an Order to see if additional approvals are needed.



Workflow Description

This workflow does the following:

Calls the SetApprovalDecision method of the ISS Approval Bus Service Business Service. This will set the status of the current approval item and check to see if additional approval is needed.

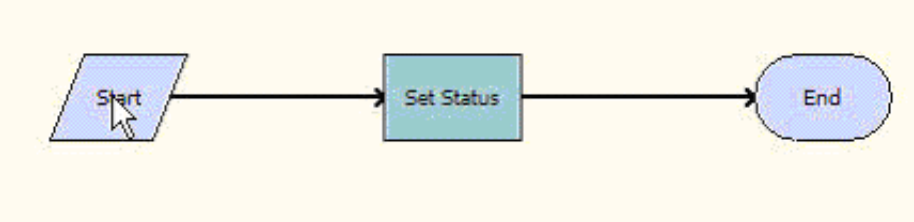
Process Properties

Name	Business Object	Data Type	Comments
ActionLIC	Order Entry	String	The Language Independent Code for the Action Field
Approval Level Name	Order Entry	String	The name of the Approval Level that is defined in the Approval Administration screen.
Error Code	Order Entry	String	The Error Code that is returned if an error occurs.
Error Message	Order Entry	String	The Error Message that is returned if an error occurs.
InboxItemId	Order Entry	String	The Row ID of the current Universal Inbox Item.
InboxTypeName	Order Entry	String	The Universal Inbox Type.
Object Id	Order Entry	String	Obsolete.
ObjectId	Order Entry	String	The Row ID of the current object.
OwnerInfoId	Order Entry	String	The Row ID of the approver.
Process Instance Id	Order Entry	String	The Row ID of the current workflow process.
Returned Status	Order Entry	String	The Status of the Approval Item.
SendEmail	Order Entry	String	Variable to determine if email is sent to the person who needs to approve the current item.
SeqNum	Order Entry	String	The Sequence Number for the current approval level.
Siebel Operation Object Id	Order Entry	String	The Row ID of the Siebel Operation Object.
StageId	Order Entry	String	The Row ID of the current Approval Stage.

ISS Post Approval Workflow (Agreement) Workflow

ISS Post Approval Workflow (Agreement) workflow, shown in the following figure, checks whether any additional approvals are needed.

It is triggered when a user approves an Agreement to see if additional approvals are needed.



Workflow Description

This workflow does the following:

Calls the SetApprovalDecision method of the ISS Approval Bus Service Business Service. This will set the status of the current approval item and check to see if additional approval is needed.

Process Properties

Name	Business Object	Data Type	Comments
ActionLIC	Service Agreement	String	The Language Independent Code for the Action Field
Approval Level Name	Service Agreement	String	The name of the Approval Level that is defined in the Approval Administration screen.
Error Code	Service Agreement	String	The Error Code that is returned if an error occurs.
Error Message	Service Agreement	String	The Error Message that is returned if an error occurs.
InboxItemId	Service Agreement	String	The Row ID of the current Universal Inbox Item.
InboxTypeName	Service Agreement	String	The Universal Inbox Type.
Object Id	Service Agreement	String	Obsolete.
ObjectId	Service Agreement	String	The Row ID of the current object.
OwnerInfoId	Service Agreement	String	The Row ID of the approver.

Name	Business Object	Data Type	Comments
Process Instance Id	Service Agreement	String	The Row ID of the current workflow process.
Returned Status	Service Agreement	String	The Status of the Approval Item.
SendEmail	Service Agreement	String	Variable to determine if email is sent to the person who needs to approve the current item.
SeqNum	Service Agreement	String	The Sequence Number for the current approval level.
Siebel Operation Object Id	Service Agreement	String	The Row ID of the Siebel Operation Object.
StageId	Service Agreement	String	The Row ID of the current Approval Stage.

Order Management Interface Methods Reference

The following methods are called by the order management interface workflows:

- *Validate Method*
- *SetApprovalDecision Method*
- *CreateNewApprovalTasks Method*
- *GetApprovalStatus Method*
- *RefreshBusinessComponent Method*

Validate Method

This method validates a quote, order, or agreement according to the Rule Sets that are defined in Administration - Data Validation.

Arguments

Input Argument	Type	Property Name	Comments
Rule Set Name	Process Property	Validation Ruleset Name	None
Store Messages	Process Property	Store Messages	This flag determines whether the validation messages are stored in the table.

Output Argument	Type	Property Name	Comments
Return Code	Output Argument	Return Code	Returns the error code. Depending on the code, the workflow determines whether approval is needed.
Return Message	Output Argument	Return Message	Returns the error message from the validation engine

SetApprovalDecision Method

This method sets the status of the current approval item and checks to see if additional approval is needed. This method is part of the ISS Approval Bus Service Business Service.

Arguments

Input Argument	Type	Property Name	Comments
ActionLIC	Process Property	ActionLIC	Approval status. Values are Approved, Rejected.
InboxItemId	Process Property	InboxItemId	ID of the current Inbox Item to be approved or rejected.
InboxTypeName	Process Property	InboxTypeName	Inbox Type defined in the Administration inbox screen; for example, Quote, Sales Order, Agreement, and so on.
ObjectId	Process Property	ObjectId	Row ID of the item to be approved; for example, the Quote ID, Order ID, and so on.
OwnerInfold	Process Property	OwnerInfold	Login ID of the person who is approving or rejecting the Inbox Item.
SendEmail	Process Property	SendEmail	Determines whether to send an email to the approver. Values are TRUE, FALSE.
SeqNum	Process Property	SeqNum	Sequence Number of the current Approval Stage.
StageId	Process Property	StageId	ID of the current Approval Stage.

CreateNewApprovalTasks Method

This method creates a new approval task for a quote, order, or agreement. This method is part of ISS Approval Business Service.

Arguments

Input Argument	Type	Property Name	Comments
Approval Level Context	Process Property	Approval Level Context	Added to accommodate Self-Service applications. Contains the Account ID.
Approval Item ID	Process Property	Object Id	Row ID of the item that needs to be approved; for example, Quote ID, Order ID, and so on.
Approval Level Name	Process Property	Approval Level Name	Name of the Approval Item that is defined in the Administration - Application screen, Approval Admin view.
Inbox Type	Process Property	Inbox Type	Inbox Type that is defined in the Administration Inbox screen; for example, Quote, Sales Order, Agreement, and so on.
Line Item #	Process Property	Line Item #	Not currently used.
Line Item Name	Process Property	Line Item Name	Not currently used.
Requesting Buscomp	Process Property	Requesting Buscomp	Business component that triggers the approval procedure. Used for dynamic drilldown (Quote, Order Entry - Orders, Service Agreement, and so on).
Return Message	Process Property	Return Message	Return Message defined in the Rule Set in the Administration - Data Validation screen, Rule Set view.
SendEmail	Process Property	SendEmail	Determines whether to send an email to the approver. Values are TRUE, FALSE.

GetApprovalStatus Method

This method returns the approval status for the inbox item. The status is one of the following: Approved, Declined, or In Progress. You must pass in the Approval Level Name, Approval Item ID, Inbox Type, and Object Id. This method is part of ISS Approval Business Service.

Input Arguments

The following table describes the GetApprovalStatus method input arguments.

Argument	Type	Property Name
Approval Item ID	Process Property	Approval Identifier
Approval Level Name	Process Property	Approval Item Name
Inbox Type Name	Process Property	Inbox Type Name

Output Arguments

The following table describes the GetApprovalStatus method output arguments.

Argument	Type	Property Name
Approval Item Status	Process Property	Approval Status

RefreshBusinessComponent Method

This method refreshes the display of a quote, order, or agreement. This method is part of the SIS OM PMT Service Business Service.

Arguments

Input Argument	Type	Value
Business Component Name	Literal	The name of the business component (such as Approval History - Quote, Approval History - Order, or Approval History - Agreement)
Business Object Name	Literal	The name of the business component (such as Quote, Order Entry (Sales), or Service Agreement).

19 Workflow Reference for Employee Asset-Based Ordering

Workflow Reference for Employee Asset-Based Ordering

This chapter explains the Siebel workflows used for employee asset-based ordering.

In this chapter, the workflows are grouped in topics that correspond to the business processes that they automate. Each topic describes a workflow process and may describe one or more subprocesses that are called by the main process, grouped together to shows how the process and its subprocesses are related to each other.

This chapter includes the following topics:

- *About Workflows for Employee Asset-Based Ordering*
- *Workflow to Ungroup Quote Line Items*
- *Workflow to Ungroup Order Line Items*
- *Workflow to Edit Quote Line Items*
- *Workflow to Convert a Quote to an Order*
- *Workflow to Edit Order Line Items*
- *Workflow to Submit an Order*
- *Workflow to Apply an Order to the Service Profile*
- *Workflows to Modify the Service Profile*
- *Workflows to Redisplay the Service Profile*
- *Workflows that Allow Users to Modify a Service Item*
- *Workflows to Disconnect a Service Item*
- *Workflows That Allow Users to Suspend or Resume Service Items*
- *Workflows for Contact-Based Asset-Based Ordering*
- *Workflows for Quotes and Orders Assets Catalog Views*
- *Workflows that Support the Automatic Configuration of Customizable Products*

About Workflows for Employee Asset-Based Ordering

Workflows are a central feature of asset-based ordering. They automate the business processes associated with managing accounts, quotes, orders, and assets.

You can modify these workflows to suit your own business model using Siebel Business Process Designer. For more information, see *Siebel Business Process Framework: Workflow Guide* .

Many of these processes and subprocesses use business service methods that are specific to asset-based ordering. The descriptions of workflows include tables listing the names of any methods they call. For information about these methods, see *Siebel Order Management Infrastructure Guide* .

Note: Some workflows in the following topics may not be relevant to your application services.

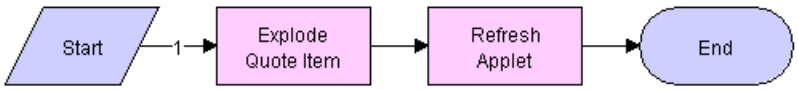
Workflow to Ungroup Quote Line Items

This workflow ungroups the line items in a quote so that all items have a quantity of 1. It consists of the workflow process *SIS OM Ungroup Quote*

SIS OM Ungroup Quote

This workflow decomposes each line item with a quantity greater than 1 into multiple line items, each with a quantity of 1. It ungroups the root product and every child product. Subsequently, a user can specify different attributes for each line item. Ungroup can act only on the root product level; it cannot ungroup individual child products without ungrouping the root and all its children. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Ungroup button in the Order Entry - Line Items view (Sales).



Workflow Description

This workflow does the following:

1. Ungroups (explodes) the quote line items.

   If the quantity of any line item is greater than 1, then the workflow creates multiple instances of the line item, each with a quantity of 1.

   Note: Items with the Auto Explode flag set to N are not exploded.

2. Refreshes the Quote Item list of the Order Entry - Line Items view (Sales).

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Explode Quote Line Item	Ungroup
Refresh Applet	Refresh Business Component

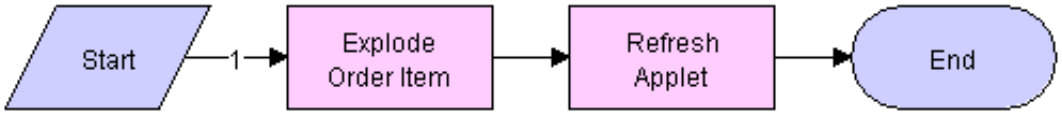
Workflow to Ungroup Order Line Items

This workflow ungroups the line items in a quote so that all items have a quantity of 1. It consists of the workflow process *SIS OM Ungroup Order*.

SIS OM Ungroup Order

This workflow decomposes each line item with a quantity greater than 1 into multiple line items, each with a quantity of 1. It ungroups the root product and every child product. Subsequently, a user can specify different attributes for each line item. Ungroup can act only on the root product level; it cannot ungroup individual child products without ungrouping the root and all its children. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Ungroup button in the Quote Detail view.



Workflow Description

This workflow does the following:

1. **Explode Order Item.** Ungroups (explodes) the order line items.

   If the quantity of any line item is greater than 1, then the workflow creates multiple instances of the line item, each with a quantity of 1.

   > **Note:** Items with the Auto Explode flag set to N are not exploded.

2. **Refresh Applet.** Refreshes the Order Entry - Line Item view's Order Item list.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Explode Order Line Item	Ungroup
Refresh Applet	Refresh Business Component

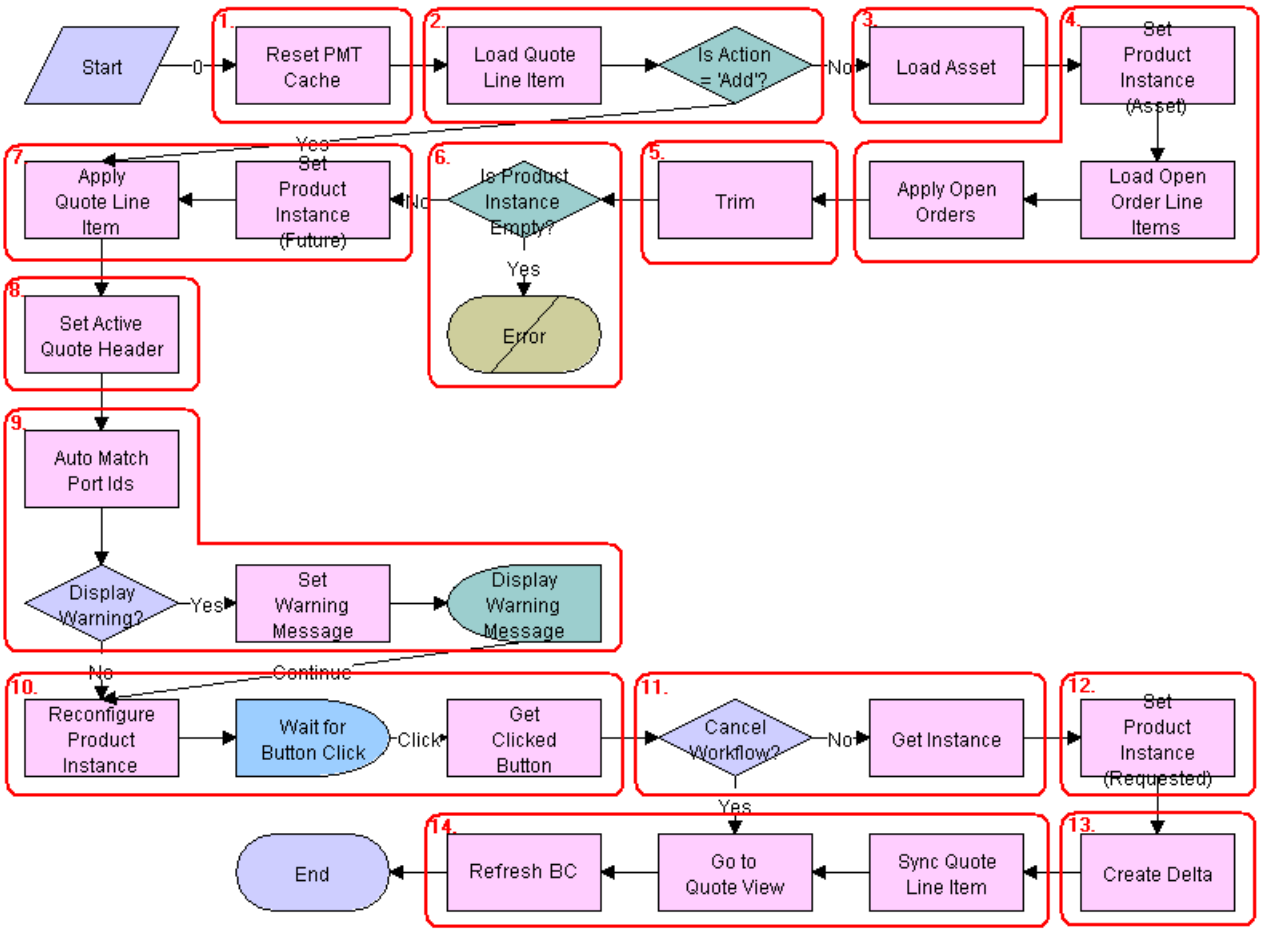
Workflow to Edit Quote Line Items

This workflow allows a user to edit the line items of quotes that result from creating or modifying service items. It consists of the workflow process *SIS OM Edit Delta Quote Line Item*.

SIS OM Edit Delta Quote Line Item

This workflow applies active and unprocessed orders to the asset associated with the selected line item and displays the asset in the Product Configurator view. The user makes changes to define the newly requested state and clicks the Done button. The workflow writes the delta of the current and requested states of the asset to the current quote as 1 or more line items. This workflow is shown in the following figure.

A user initiates this workflow by clicking Customize in the Quote Detail view.



Workflow Description

This workflow does the following:

1. Clears business service cache of existing product instances.

2. Checks the action code of the high-level component in the order line item. If the action code is Add, then goes to Step 7.
3. If the action code is not Add, reads the asset associated with the selected quote line item from the database. Components with a status of Inactive are excluded.
4. Retrieves all prior active and unprocessed orders that relate to the asset and applies them to generate the requested future state of the asset.
5. Removes all asset components that do not have the 'Track As Asset' flag set.
6. If one of the active and unprocessed orders disconnects the high-level component, or there are asset components with the 'Track As Asset' flag set, then the resulting asset is empty. The workflow displays the following error message: A request has already been made to disconnect this service item. The workflow stops.
7. Applies the current quote line item to the asset.
8. Stores the quote header so that line items can be added to it.
9. Self-heals the asset such that the asset structure matches the product definition. If any changes are made to the asset, then a warning message appears.
10. Displays the currently requested state of the asset in the Product Configurator view. The user makes the required changes and clicks the Done or Cancel button.
11. If the user clicks the Cancel button, then the workflow is terminated. If the user clicks the Done button, then the newly requested state is retrieved from the Configurator.
12. If the action code of the root item is not Add, caches the previously requested state of the asset to be used as an input to Delta.
13. Generates the delta of the following as 1 or more quote line items:
    - The requested state of the asset before the current order line item was applied.
    - The state of the asset after the user has reconfigured it.
14. Writes the quote line item to the database and refreshes the view.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance (Asset)	Set Product Instance
Load Open Order Line Items	Find Orders
Apply Open Orders	Apply
Trim	Trim
Set Product Instance (Future)	Set Product Instance
Apply Changes Quote Line Item	Apply
Set Active Quote Header	Set Output Header

Workflow Step	Business Service Method Called
Auto Match Port Ids	AutoMatch
Reconfigure Product Instance	Reconfigure Product Instance
Get Clicked Button	Get Configure Button Click Information
Get Instance	Get Product Instance
Set Product Instance (Requested)	Set Product Instance
Create Delta	Delta
Sync Quote Line Item	Synchronize
Go To Quote View	GotoView

Workflow to Convert a Quote to an Order

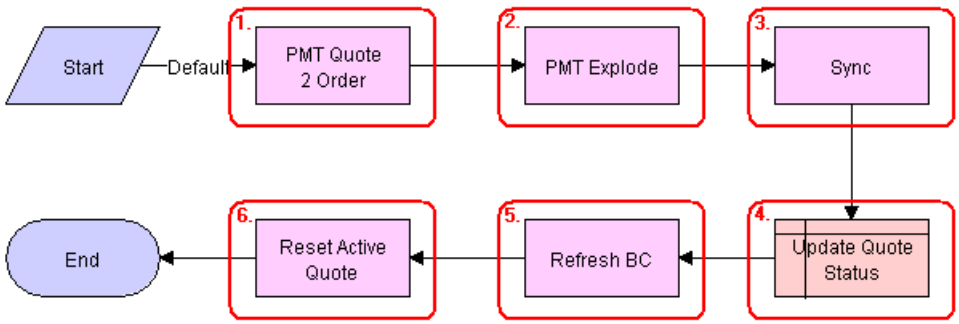
This workflow allows a user to convert a sales quote to a sales order. It consists of the workflow processes:

- *SIS OM Quote To Order PMT Version*
- *CUT Quote To Order PMT Enhancement*
  - *CUT Quote To Order PMT Enhancement - Sub-process*

SIS OM Quote To Order PMT Version

This workflow creates an order that is associated with the current quote. It copies the line items from the quote to the order and sets the quote's status to Order Placed. This workflow is shown in the following figure.

A user initiates this workflow by choosing the Sales Order button or menu option in the Quote Orders view.



Workflow Description

This workflow does the following:

1. Converts a quote to an order. The new order has all the line items and attributes of the quote.
2. Explodes line items with a quantity greater than 1 into multiple copies of the line item, each with quantity 1.
3. Writes the order to the database.
4. Set the status of the quote to Order Placed.
5. Refreshes the Quote Orders View.
6. Resets the Active Quote so that the current quote is no longer the active quote.

Associated Business Service Methods

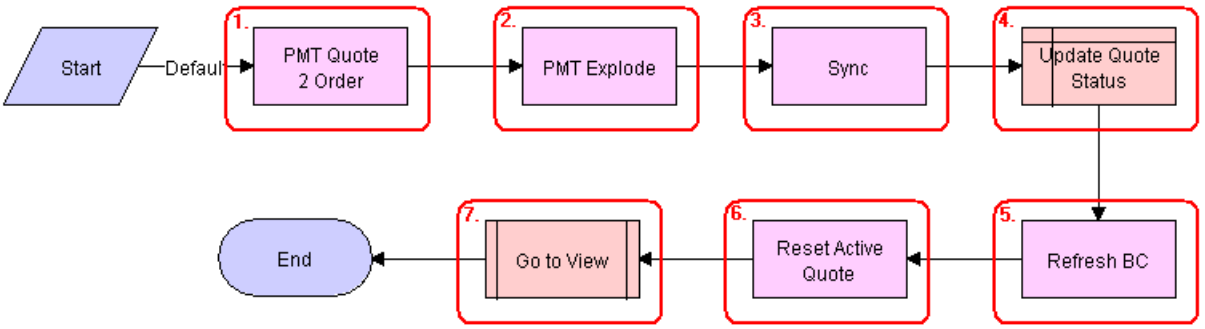
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
PMT Quote 2 Order	Convert Product Instance
PMT Explode	Explode
Sync	Synchronize
Refresh BC	Refresh Business Component
Reset Active Quote	Set Profile Attribute

CUT Quote To Order PMT Enhancement

This workflow creates an order that is associated with the current quote. It copies the line items from the quote to the order and sets the quote's status to Order Placed. It then displays the new order in the Order Entry - Line Items view. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Auto Order button in the Quote Detail view.



Workflow Description

This workflow does the following:

1. Converts a quote to an order. The new order has all the line items and attributes of the quote.
2. Explodes line items with a quantity greater than 1 into multiple copies of the line item, each with quantity 1.
3. Writes the order to the database.
4. Set the status of the quote to Order Placed.
5. Refreshes the Quote Orders business component.
6. Resets the Active Quote such that the current quote is no longer the active quote.
7. Displays the Order Entry - Line Items view.

Associated Business Service Methods

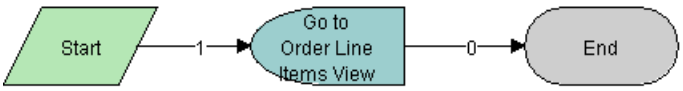
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
PMT Quote 2 Order	Convert Product Instance
PMT Explode	Explode
Sync	Synchronize
Refresh BC	Refresh Business Component
Reset Active Quote	Set Profile Attribute

CUT Quote To Order PMT Enhancement - Sub-process

This workflow displays the Order Entry - Line Items view.

The CUT Quote To Order PMT Enhancement workflow initiates this workflow. The workflow is shown in the following figure.



Workflow to Edit Order Line Items

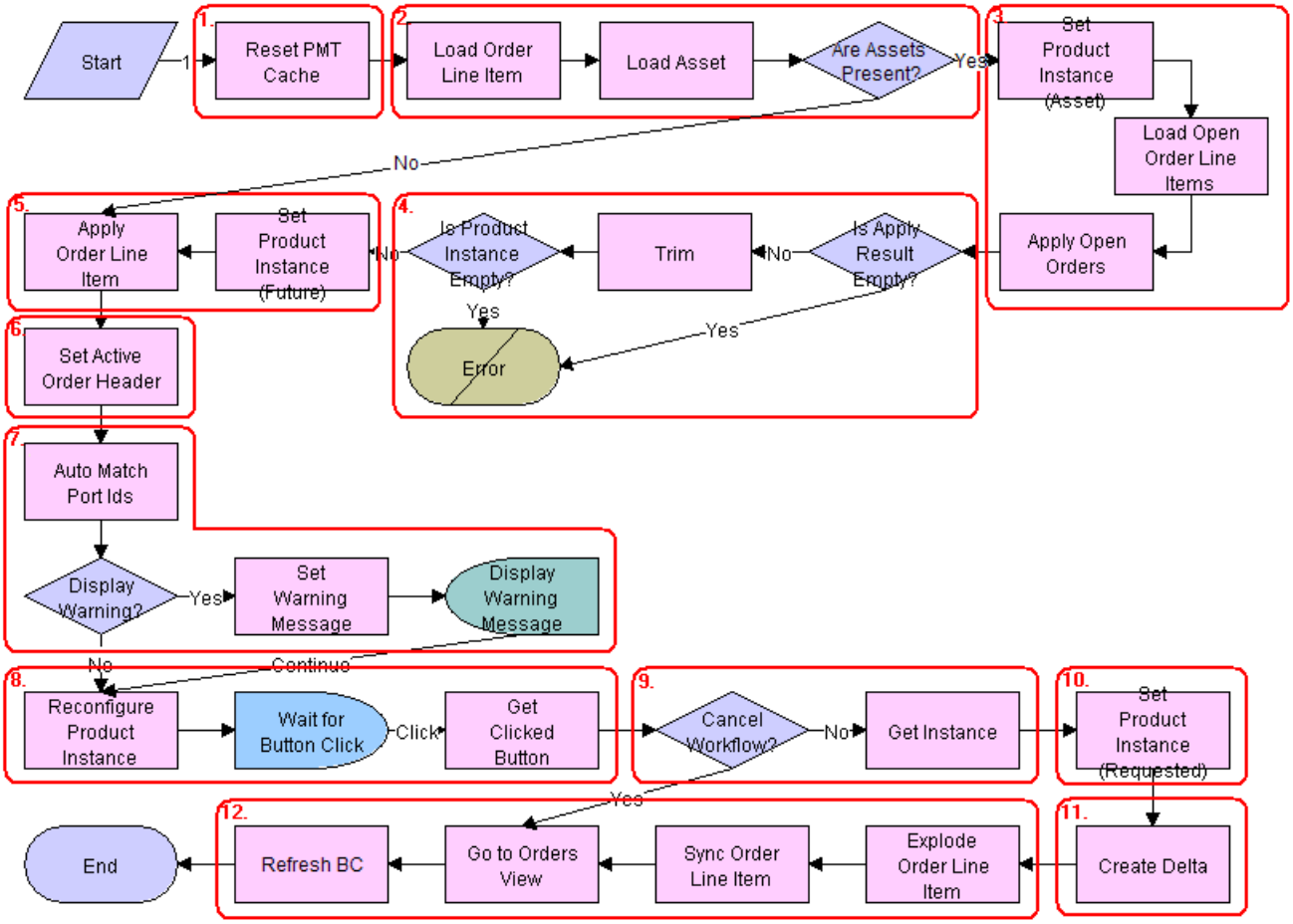
This workflow allows a user to edit order line items. It consists of the workflow process:

- *SIS OM Edit Service Order Line Item*

SIS OM Edit Service Order Line Item

This workflow applies active and unprocessed orders to the asset associated with the selected line item and displays the asset in the Product Configurator view. The user makes changes to define the newly requested state and clicks the Done button. The workflow then writes the delta of the current and requested states of the asset to the current order as 1 or more line items. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Customize button in the Order Detail view.



Workflow Description

This workflow does the following:

1. Clears the business service cache of existing product instances.
2. Loads the asset associated with the current order line item, if it exists. Checks whether the asset has been successfully loaded. If not, go to *SIS OM Edit Service Order Line Item*.
3. Retrieves all prior active and unprocessed orders related to the asset and applies them to generate the requested future state of the asset.
4. Removes the asset components that do not have the 'Track As Asset' flag set. If none of the components has the 'Track As Asset' flag set or one of the active and unprocessed orders disconnects the high-level component, then the resulting asset is empty. The workflow displays the following error message: A request has already been made to disconnect this service item. The workflow stops.
5. Applies the current order line item to the asset.

6. Stores the order header so that line items can be added to it.
7. Self-heals the asset such that the asset structure matches the product definition. If any changes are made to the asset, then a warning message is displayed.
8. Displays the currently requested state of the asset in the Product Configurator view. The user makes the required changes and clicks the Done or Cancel button.
9. If the user clicks the Cancel button, then the workflow is terminated. If the user clicks the Done button, then the newly requested state is retrieved from the Configurator.
10. Caches the newly requested state of the asset to be used as an input to Delta.
11. Generates the delta of the following as one or more order line items:
    - The requested state of the asset before the current order line item was applied.
    - The state of the asset after the user has reconfigured it.
12. Explodes (ungroups) any order line item that has a quantity of more than 1 into multiple line items, each with a quantity of 1. Writes the order line item to the database and refreshes the view.

> **Note:** This step does not control the quantity of the line items, but only works with the quantity it has been given. The user specifies order line item quantities in Configurator.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance (Asset)	Set Product Instance
Load Open Order Line Items	Find Orders
Apply Open Orders	Apply
Trim	Trim
Set Product Instance (Future)	Set Product Instance
Apply Order Line Item	Apply
Set Active Order Header	Set Output Header
Auto Match Port Ids	AutoMatch
Reconfigure Product Instance	Reconfigure Product Instance
Get Clicked Button	Get Configure Button Click Information

Workflow Step	Business Service Method Called
Get Instance	Get Product Instance
Set Product Instance (Requested)	Set Product Instance
Create Delta	Delta
Explode Order Line Items	Explode
Sync Order Line Items	Synchronize
Go to Orders View	GotoView

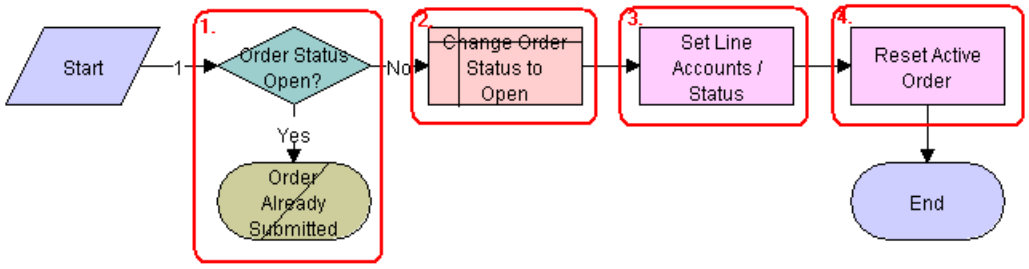
Workflow to Submit an Order

This workflow allows a user to submit an order, which signals the provisioning application that the order is ready to be provisioned. It consists of the workflow process: *SIS OM Submit Order Process*.

SIS OM Submit Order Process

This workflow checks that account information exists. It sets the status of the order and the line items to Open. Because only orders that do not have a status of Open can be submitted, this prevents the order from being submitted again. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Submit button on the order header form in the Order Entry views. The Submit button is hard coded to call this workflow.



Workflow Description

This workflow does the following:

1. Checks that the order status is not Open.
2. If the order has a status of Open, then it cannot be submitted, because it has already been submitted.
3. Sets the status of the order to Open.
4. Sets the status of each Pending line item to Open.

5. Resets the Active Order such that the current order is no longer the active order.

Workflow to Apply an Order to the Service Profile

This workflow applies a completed order line item to the current account's service profile:

- *SIS OM Apply Completed Service Order Line Item to Service Profile*

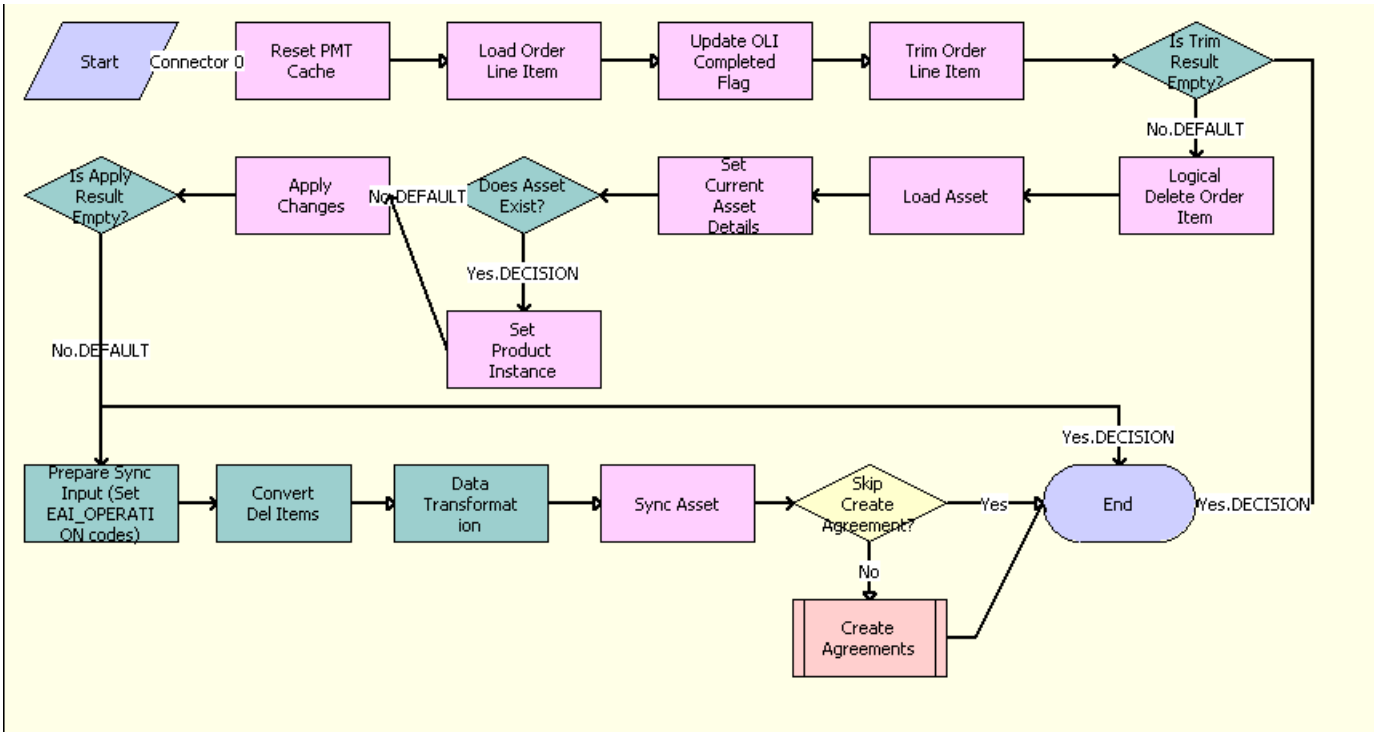
SIS OM Apply Completed Service Order Line Item to Service Profile

This workflow creates assets from completed order line items or modifies a current asset. It therefore maintains the account's service profile, which consists of its associated assets. If there are multiple accounts in an account hierarchy, then service items are applied to the service account. This workflow is shown in the following figure.

You must develop an order synchronization workflow that updates the status of order line items in your Siebel Business Application when your company's external order management application completes all or part of an order. After it updates all the order items, the order synchronization workflow must call this workflow to apply the completed changes to the customer's service profile, which is stored as a series of assets. You can disable this workflow if an external application maintains service profiles.

By default, a user initiates this workflow by clicking the Auto-Asset button in the Order Entry - Order Line Items view.

> **Note:** The Auto-Asset button is designed for testing and demonstration purposes, not for actual end use. It is recommended that you configure your Siebel Business Application to call this workflow as a subprocess of the order synchronization workflow mentioned previously.



Workflow Description

This workflow does the following:

1. **Reset PMT Cache.** Clears business service cache of existing product instances.
2. **Load Order Line Item.** Loads the selected order line item from the database.
3. **Update OLI Completed Flag.** Updates the Order Item Processed flag. The workflow updates the Order Item Processed flag of the topmost order line item. The workflow sets this flag to Y if all line items have a status of Complete, Failed, or -. Other workflows use this flag to determine whether a line item remains Open and must be applied to the current service profile to generate a requested future state.

   > **Note:** To set the flag for the other status values, add these status values to this step's I/O Argument *Complete Statuses*. The default value is: Complete, Rejected.

4. **Trim Order Line Item.** Applies the Trim method. Trim eliminates any incomplete or failed changes from the order line item.

   Trim keeps the line items whose products are tracked as assets and meet one of the following conditions:

   - whose status is **Complete**
   - whose action code is **Existing**

   This causes the service profile to reflect the true state of the assets.

   > **Note:** To keep the line items with the other status values, add these status values to this step's I/O Argument *KeepSpec*. The default value is: (([Status] = LookupValue('FS_ORDER_STATUS', 'Complete')) OR ([Action Code] = LookupValue('DELTA_ACTION_CODE', 'Existing'))) AND ([Convert To Asset Flag] = 'Y').

5. **Is Trim Result Empty?** Checks whether line items exist. If Trim has eliminated all line items, then the workflow stops. Otherwise, the workflow continues.

6. **Logical Delete Order Item.** Converts Delete actions into Modified actions. The Logical Delete step converts Delete actions into Modified actions and sets the status to Inactive. This step supports the maintenance of a history of disconnected services.
7. **Load Asset.** Loads the service profile from the database. If the order line item is a change to an existing item, then the workflow loads the corresponding service profile, or asset, from the database.
8. **Set Current Asset Details.** Uses the SetOldAssetDetails method of the ISS Promotion Agreement Manager business service to store Row Id of the current asset, the Integration Id of the current asset, the Row Id of the promotion associated with the current asset, the Promotion Instance Id of the current asset, and so on, in internal storage, so these IDs can be used by the GetPromotionDetails method.
9. **Does Asset Exist?** Checks whether there are any existing assets, that is, whetherthe number of assets is greater than 0.
10. **Set Product Instance.** Calls the Set Product Instance method of the Set Product Instance business service, which loads the product instance of the asset if any asset exists. The next step, Apply Changes, applies to the product instance of the asset that is loaded in this step.
11. **Apply Changes.** Applies changes related the order line item to the asset. If the asset does not already exist, then the workflow creates a new asset.
12. **Is Apply Result Empty?** Checks whether the asset is empty. If, as a result of applying the changes, the asset is empty, then the workflow stops. Otherwise, the workflow writes the asset to the database.
13. **Prepare Sync Input (Set EAI_Operation Codes).** Determines the operation that will be used, Insert, Update or Delete. This operation is entered in the Operation field of the Siebel Message. Based on the value of this field, the Sync Asset step determines which operation to perform.
14. **Convert Del Items.** Uses the ConvertDelItemsFieldToChildren method of the SIS OM PMT Service business service to convert the details of deleted products from Line Item fields to Integration Component MACD Order User Deleted Item structure, which will be used to update the deleted item details in the asset table.
15. **Data Transformation.** Uses the Execute method of the EAI Data Transformation Engine business service to transfer the asset hierarchy from Header - Line - XA to Line -XA. The header and line items business components are based on the same table, and this reduced hierarchy reduces the time taken to synchronize the data in the table.
16. **Sync Asset.** Uses the Synchronize method of the EAI Siebel Adapter business service to synchronize the updated asset structure to the Asset tables.
17. **Skip Create Agreement.** Determines whether or not a new agreement is created based on the input value of Skip Agreement.
18. **Create Agreements.** Calls ISS Promotion Agreement Management Sub Process workflow, which creates a new agreement or updates an existing agreement.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Update OLI Completed Flag	Update Order Line Item Completed Flag
Trim Order Line Item	Trim
Logical Delete Order Line Item	Logical Delete
Apply Changes	Apply

Workflow Step	Business Service Method Called
Set Product Instance	Set Product Instance
Sync Asset	Synchronize

Workflows to Modify the Service Profile

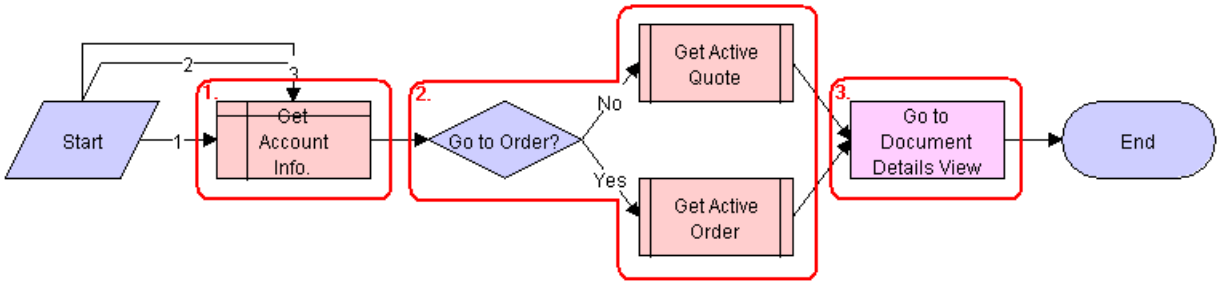
This group of workflows allows a user to modify a service profile by requesting new service items. It consists of the workflow processes:

- *SIS OM New Products & Services Process*
- *SIS OM New Products & Services Process - VORD*

SIS OM New Products & Services Process

This workflow displays the active document, if there is one, or a newly created document in the Quote Detail or Order Entry - Line Items view. The user can then add line items, add or delete an asset, or change the attributes of the asset. This workflow is shown in the following figure.

A user initiates this workflow by clicking the New button in the Account screen's Service Items view, or Account Summary view. This workflow is also initiated as a subprocess of the SIS OM New Products & Services Process - VORD.



Workflow Description

This workflow does the following:

1. Gets the account information.
2. If the account is of type 'Residential', then the workflow retrieves the active order. Otherwise it retrieves the active quote.

   The workflow retrieves the active document (quote or order) from the session. If there is no active document, then a new document is created and set as the active document. If there is an active document, then the subprocess checks that the document is associated with the same customer account as the asset being modified, and that the document has the correct status. See *SIS OM Active Quote Sub-Process* and *SIS OM Active Order Sub-Process*. If the quote does not meet these criteria, then the workflow creates a new document and sets it as the active document.

3. Displays the document in the Quote Details or Order Entry - Line items view, as appropriate.

Associated Business Service Methods

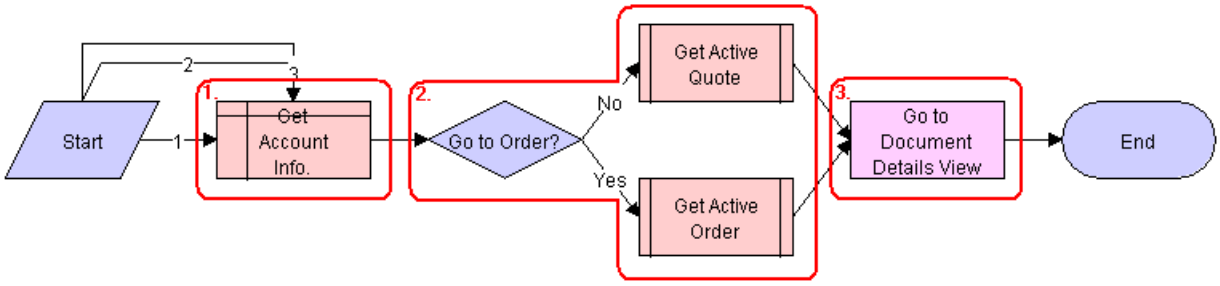
The following table shows the step in this workflow that calls business service methods.

Workflow Step	Business Service Method Called
Go to Document Details View	GotoView

SIS OM New Products & Services Process - VORD

This workflow displays the active document, if there is one, or a newly created document in the Quote Detail or Order Entry - Line Items view. The user can then add line items, add or delete an asset, or change the attributes of the asset. This workflow is shown in the following figure.

A user initiates this workflow by clicking the New button in the Networks screen's Nodes, Connections or All Elements view.



Workflow Description

This workflow does the following:

- Gets the account information associated with the current network and initiates the SIS OM New Products & Services workflow to display the active document.

Workflows to Redisplay the Service Profile

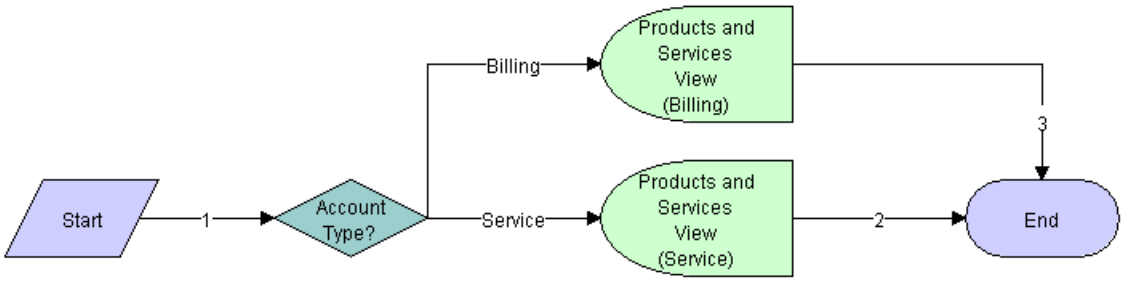
The workflows that navigate to the current account's service profile, which displays the account's active assets, are:

- *SIS OM Profile Process*
- *SIS OM Profile Process - Order*

SIS OM Profile Process

This workflow makes the current quote the active quote. It then displays the Billing Items or Service Items view for the account associated with the current quote. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Profile button in the Quote Details view.



Workflow Description

This workflow does the following:

1. Displays an error message and stops if the quote header does not specify an account.
2. Sets the current quote as the active quote.
3. Displays the Service Items view.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset Active Quote	Set Profile Attribute
Set Active Quote	Set Profile Attribute
Go to Account Profile View	GotoView

SIS OM Profile Process - Order

This workflow is exactly the same as the *SIS OM Profile Process* workflow, except that it makes the current order the active order, and it displays the Service Items View for the account associated with that order.

Workflows that Allow Users to Modify a Service Item

This group of workflows allows a user to modify a service item. It consists of the workflow process and subprocesses:

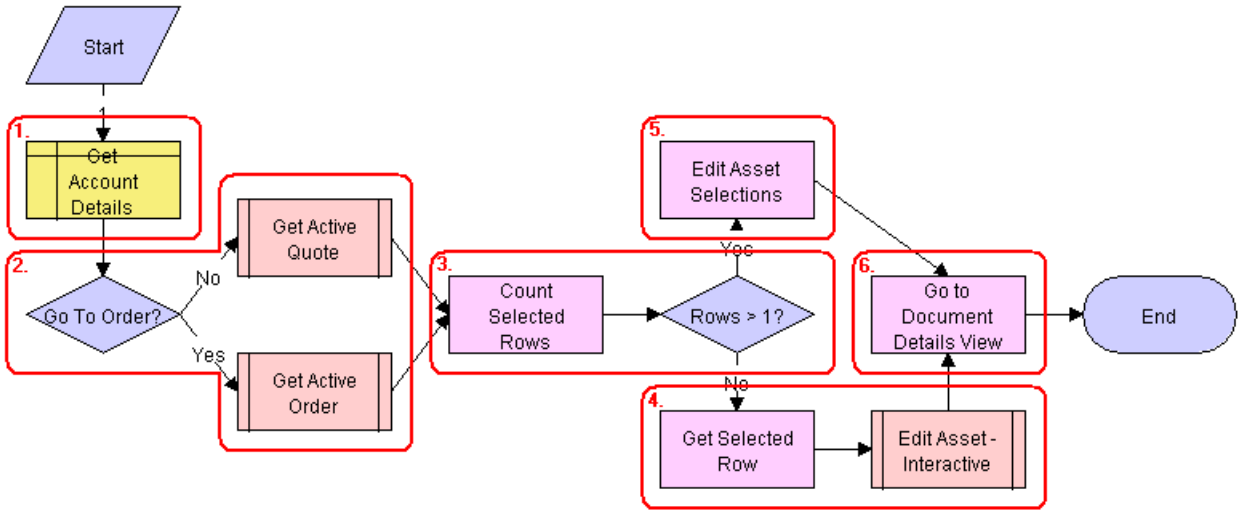
- *SIS OM Modify Products & Services Process*
  - *SIS OM Active Quote Sub-Process*
  - *SIS OM Active Order Sub-Process*
  - *SIS OM Edit Complex Asset Workflow*

SIS OM Modify Products & Services Process

This workflow determines whether there is an active quote for the current account. If there is no active quote, then it creates a new quote. It generates a quote line item to update the selected asset. If the product is a product with components, then the workflow displays the service item in Configurator with all related open orders applied to it. Otherwise, it displays the Quote Detail view. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Modify button in the following views:

- Accounts screen, Service Items view
- Accounts screen, Account Summary view
- Networks screen, Nodes view
- Networks screen, Connections view
- Networks screen, All Elements view
- Orders screen, Order Entry - Line Items view



Workflow Description

This workflow does the following:

1. Gets the details of the current account.

2. Retrieves the active document (quote or order) for the session.

   If the account is of type 'Residential' then the active order is retrieved. Otherwise the active quote is retrieved. See *SIS OM Active Quote Sub-Process* and *SIS OM Active Order Sub-Process*.
3. Gets the number of assets that the user has selected to modify.
4. If the user has selected only one asset, then the currently requested state of the asset is retrieved and displayed in the Product Configurator view. See *SIS OM Edit Complex Asset Workflow*.
5. If the user has selected more than one asset, then the currently requested state of each asset is retrieved and added to the active document. See *SIS OM Edit Complex Asset Workflow*.
6. Displays active document.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

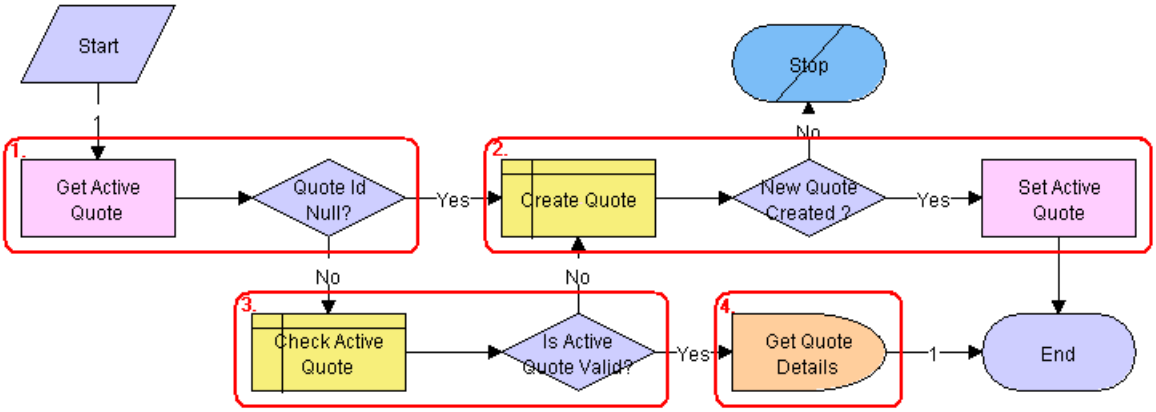
Workflow Step	Business Service Method Called
Count Selected Rows	Get Selected Row Count
Get Selected Row	Get First Selected Row Value
Edit Asset Selections	Iterate Process For Selected Rows
Go to Document Details View	GotoView

SIS OM Active Quote Sub-Process

This workflow determines whether the user has an active quote for the current account. The workflow has two process properties that are set by the calling workflow: the Row ID of the customer account associated with the asset and the Row ID of the active quote. This workflow is shown in the following figure.

The following workflows initiate this workflow as a subprocess:

- SIS OM New Products & Services Process
- SIS OM New Products & Services Process - VORD
- SIS OM Modify Products & Services Process
- SIS OM Disconnect Products & Services Process
- SIS OM Suspend / Resume Products & Services Process



Workflow Description

This workflow does the following:

1. Retrieves the active quote for the session and checks that there is a value.
2. If there is no active quote, then a new quote is created and set as the active quote.
3. If there is an active quote, determines whether the quote is associated with the same customer account as the asset being modified, and whether the quote has a status of In Progress. If the quote is not valid, then a new quote is created and set as the active quote.
4. If the new quote is valid, retrieves the price list Id and the currency code associated with the quote and returns them to the calling workflow.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

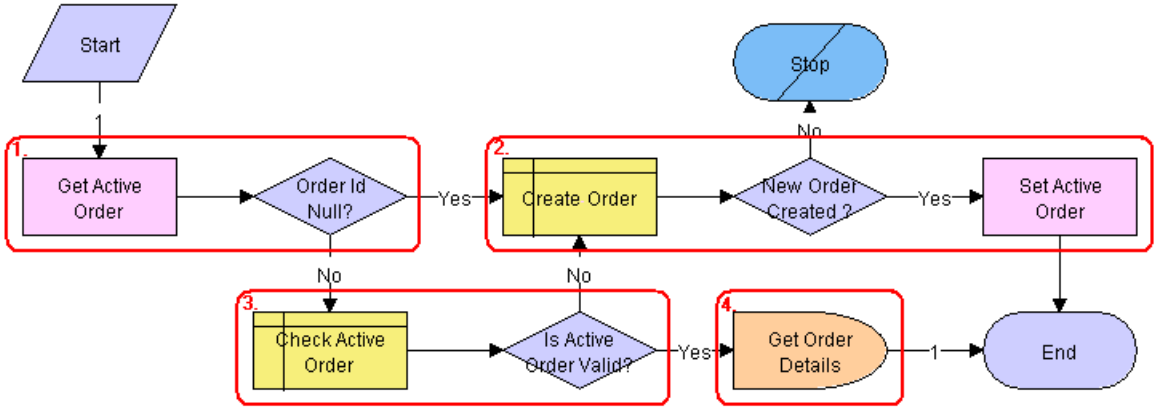
Workflow Step	Business Service Method Called
Count Selected Rows	Get Selected Row Count
Get Selected Row	Get First Selected Row Value
Get Active Quote	Get Profile Attribute
Set Active Quote	Set Profile Attribute

SIS OM Active Order Sub-Process

This workflow determines whether the user has an active order for the current account. The workflow has two process properties that are set by the calling workflow: the Row ID of the customer account associated with the asset and the Row ID of the active order. This workflow is shown in the following figure.

The following workflows initiate this workflow as a subprocess:

- SIS OM New Products & Services Process
- SIS OM New Products & Services Process - VORD
- SIS OM Modify Products & Services Process
- SIS OM Disconnect Products & Services Process
- SIS OM Suspend / Resume Products & Services Process



Workflow Description

This workflow does the following:

1. Retrieves the active order for the session and checks that there is a value.
2. If there is no active order, then a new order is created and set as the active order.
3. If there is an active order, determines whether the order is associated with the same customer account as the asset being modified, and whether the quote has a status of Open. If the quote is not valid, then a new order is created and set as the active order.
4. If the new order is valid, retrieves the price list Id and the currency code associated with the order and returns them to the calling workflow.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

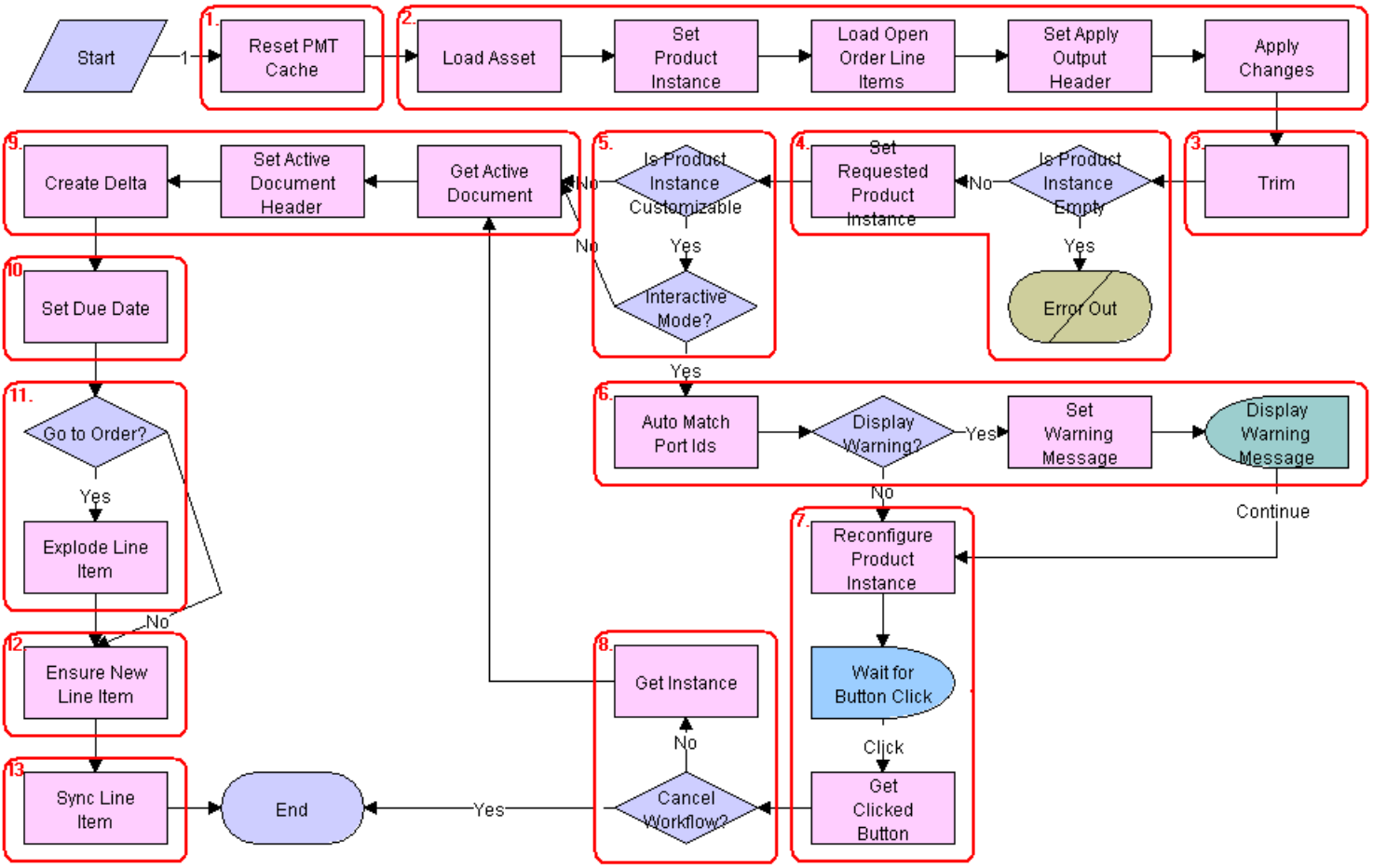
Workflow Step	Business Service Method Called
Get Active Quote	Get Profile Attribute
Set Active Quote	Set Profile Attribute

SIS OM Edit Complex Asset Workflow

This workflow applies the open and pending orders related to the selected asset and displays the requested state in the Product Configurator view. After the user makes the changes that specify the newly requested state, the workflow

writes the delta of the two requested states as quote line items and adds them to the active quote. This workflow is shown in the following figure.

The SIS OM Modify Products & Services Process initiates this workflow.



Workflow Description

This workflow does the following:

1. Clears the business service cache of existing product instances.
2. Reads the asset from the database.

   Only components that do not have a status of Inactive are included. The changes specified by all the open and pending orders related to the asset are applied to it to re-create the state of the service product as the customer has requested.
3. Removes components from the asset that do not have the 'Track As Asset' flag set.
4. Determines whether an open or pending order disconnects the root component. If one of the open or pending orders disconnects the root component, or there are no components of the asset with the 'Track As Asset' flag set, then the workflow returns an empty asset and displays an error message.
5. Tests for a customizable asset and interactive mode.

   If the asset is customizable and the workflow is in interactive mode, then the asset is displayed in the Product Configurator view, where the user can make changes to it. If the asset is not customizable or the workflow is not in interactive mode, then it goes directly to Step 9.

6. Self-heals the asset such that the asset structure matches the product definition. If any changes are made to the asset, then a warning message appears.
7. The future requested state of the asset is displayed in the Configurator view.
8. If the user clicks the Cancel button, then the workflow is terminated. If the user clicks the Done button, then the newly requested state is retrieved from the Configurator.
9. The workflow retrieves the active document. The delta of the previously requested state of the asset and the newly requested state of the asset are added as line items.
10. Sets the due date for all the line items.
11. If the active document is an order, then the line items with a quantity greater than 1 are exploded into multiple copies of the line item, each with quantity 1.
12. Generates a unique asset integration Id for all line items with an action code of Add.
13. Saves the document to the database and refreshes the view.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance	Set Product Instance
Load Open Order Line Items	Find Orders
Set Apply Output Header	Set Output Header
Apply Changes	Apply
Trim	Trim
Set Requested Product Instance	Set Product Instance
Auto Match Port Ids	AutoMatch
Reconfigure Product Instance	Reconfigure Product Instance
Get Clicked Button	Get Cfg Button Click Information
Get Instance	Get Product Instance
Set Active Document Header	Set Output Header
Create Delta	Delta
Set Due Date	Set Field Value

Workflow Step	Business Service Method Called
Explode Line Item	Explode
Ensure New Line Item	Set Field Value
Sync Line Item	Synchronize

Workflows to Disconnect a Service Item

This group of workflows allows a user to disconnect a service. It consists of the workflow process and subprocess:

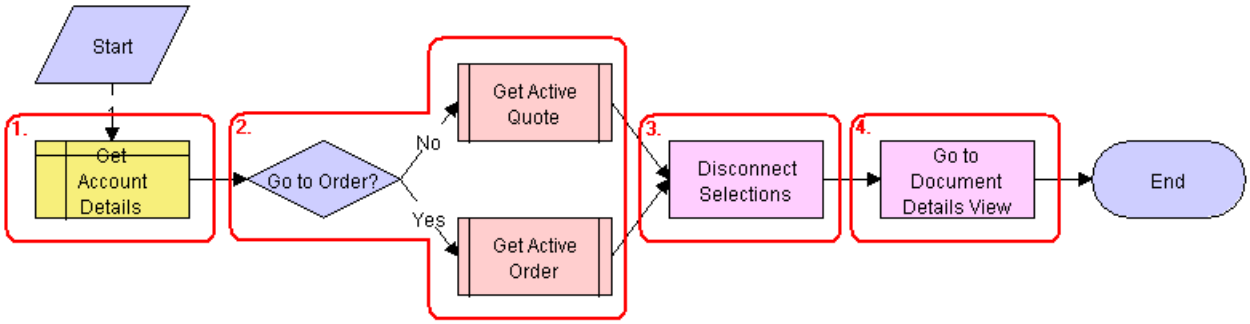
- *SIS OM Disconnect Products & Services Process*
  - *SIS OM Disconnect Asset Sub-process*

SIS OM Disconnect Products & Services Process

This workflow identifies an active document (quote or order) for the current account. Then it generates a line item to disconnect the selected asset and displays the document detail view. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Disconnect button in the following views:

- Accounts screen, Service Items view
- Accounts screen, Account Summary view
- Networks screen, Nodes view
- Networks screen, Connections view
- Networks screen, All Elements view



Workflow Description

This workflow does the following:

1. Gets the details of the current account.
2. Retrieves the active document (quote or order) for the session.

If the account is of type 'Residential', then the active order is retrieved. Otherwise, the active quote is retrieved. See *SIS OM Active Quote Sub-Process* and *SIS OM Active Order Sub-Process*.

3. Allows the user to disconnect the asset.

   The workflow gets the asset's current state and then creates quote line items to disconnect the asset in that state. See *SIS OM Disconnect Asset Sub-process*.

   > **Note:** When deleting a network node, the process will automatically delete all associated network connections. If this is not desirable behavior, then change the value of the Delete Connection argument to N.

4. Displays active document.

Associated Business Service Methods

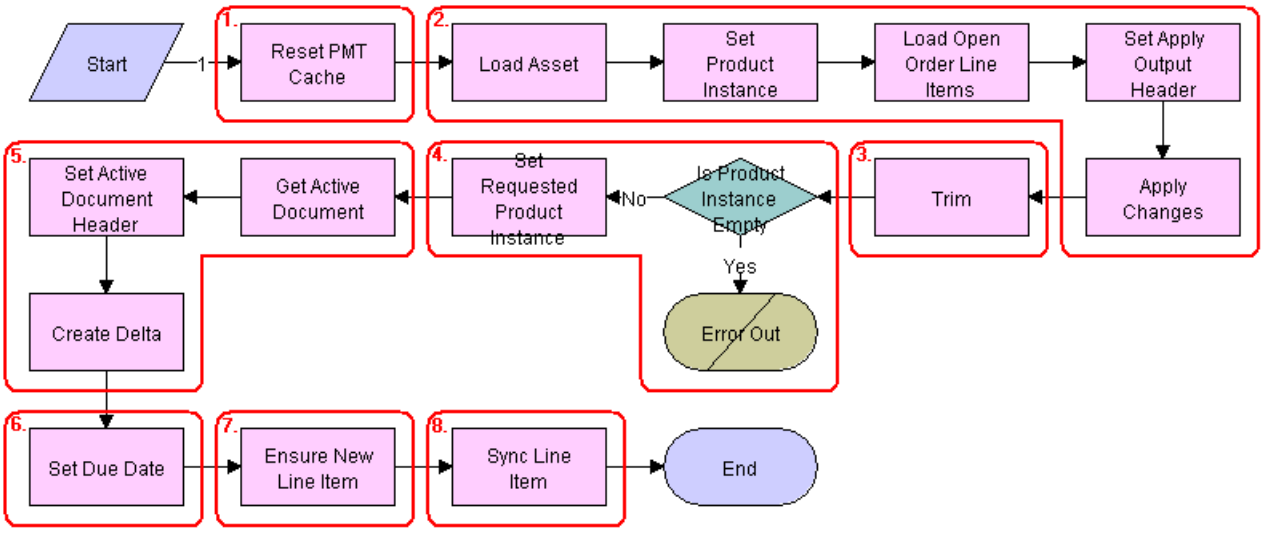
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Disconnect Selections	Iterate Process For Selected Rows
Go to Document Details View	GotoView

SIS OM Disconnect Asset Sub-process

This workflow applies the open and pending orders related to the selected asset to get its requested state. Then it creates quote line items to disconnect the selected asset in that state. It adds these line items to the active quote. This workflow is shown in the following figure.

The SIS OM Disconnect Products & Services Process initiates this workflow.



Workflow Description

This workflow does the following:

1. Clears the business service cache of existing product instances.
2. Reads the asset from the database.

   The workflow includes only components that do not have a status of Inactive. The workflow looks for all the orders with a status of Open or Pending that are related to the selected asset. The workflow applies the changes specified by those orders. This re-creates the state of the service product as the user has requested it.
3. Removes any components from the asset that do not have the 'Track As Asset' flag set.
4. Displays an error message if the asset is empty.

   If one of the open or pending orders disconnects the root component, or there are no components that have the 'Track As Asset' flag set, then the result is an empty asset. In this case, the workflow displays an error message.
5. Retrieves the active document. The workflow generates the delta of the requested state of the asset and the future empty state of the asset. This creates line items to disconnect the asset. The workflow adds these line items to the active document.
6. Sets the due date for the line items.
7. Generates a new asset integration ID for any line item with an action code of Add.
8. Saves the document to the database.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance	Set Product Instance
Load Open Order Line Items	Find Orders
Set Apply Output Header	Set Output Header
Apply Changes	Apply
Trim	Trim
Set Requested Product Instance	Set Product Instance
Set Active Document Header	Set Output Header
Create Delta	Delta
Set Due Date	Set Field Value

Workflow Step	Business Service Method Called
Ensure New Line Item	Set Field Value
Sync Line Item	Synchronize

Workflows That Allow Users to Suspend or Resume Service Items

This group of workflows allows a user to suspend or resume a service. It consists of the workflow process and subprocesses:

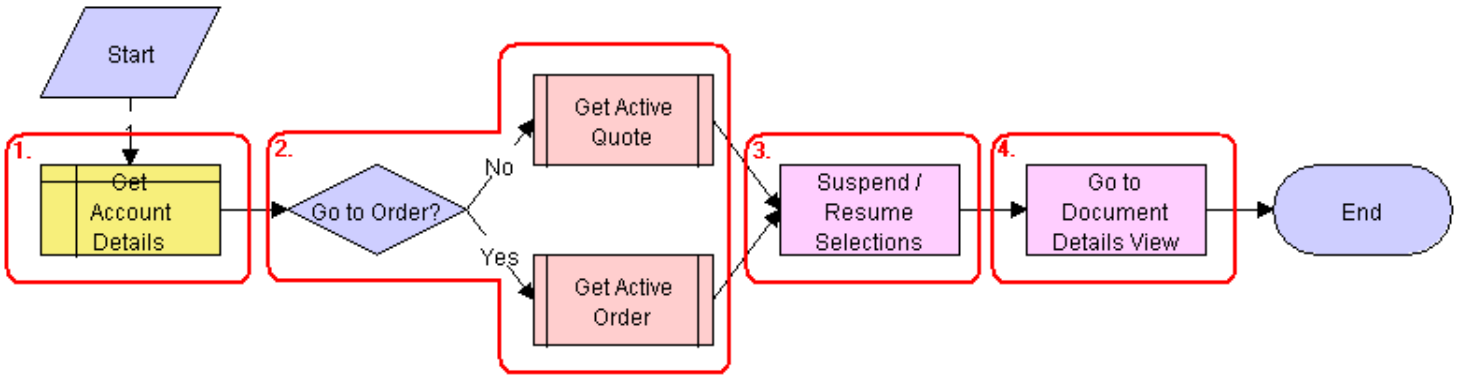
- *SIS OM Suspend / Resume Products & Services Process*
- *SIS OM Suspend / Resume Asset Sub-process*

SIS OM Suspend / Resume Products & Services Process

This workflow identifies an active document for the current account. Then it generates a line item to suspend or resume the selected asset and displays the document detail view. This workflow is shown in the following figure.

A user initiates this workflow by clicking the Suspend or Resume button in the following views:

- Accounts screen, Service Items view
- Accounts screen, Account Summary view
- Networks screen, Nodes view
- Networks screen, Connections view
- Networks screen, All Elements view



Workflow Description

This workflow does the following:

1. Gets the details of the current account.

2. Retrieves the active document (quote or order) for the session.

   If the account is of type 'Residential', then the active order is retrieved. Otherwise, the active quote is retrieved. See *SIS OM Active Quote Sub-Process* and *SIS OM Active Order Sub-Process*.
3. Allows the user to suspend or resume the asset.

   The workflow gets the asset's current state and then creates quote line items to suspend or resume the asset in that state. See *SIS OM Suspend / Resume Asset Sub-process*.
4. Displays active document.

Associated Business Service Methods

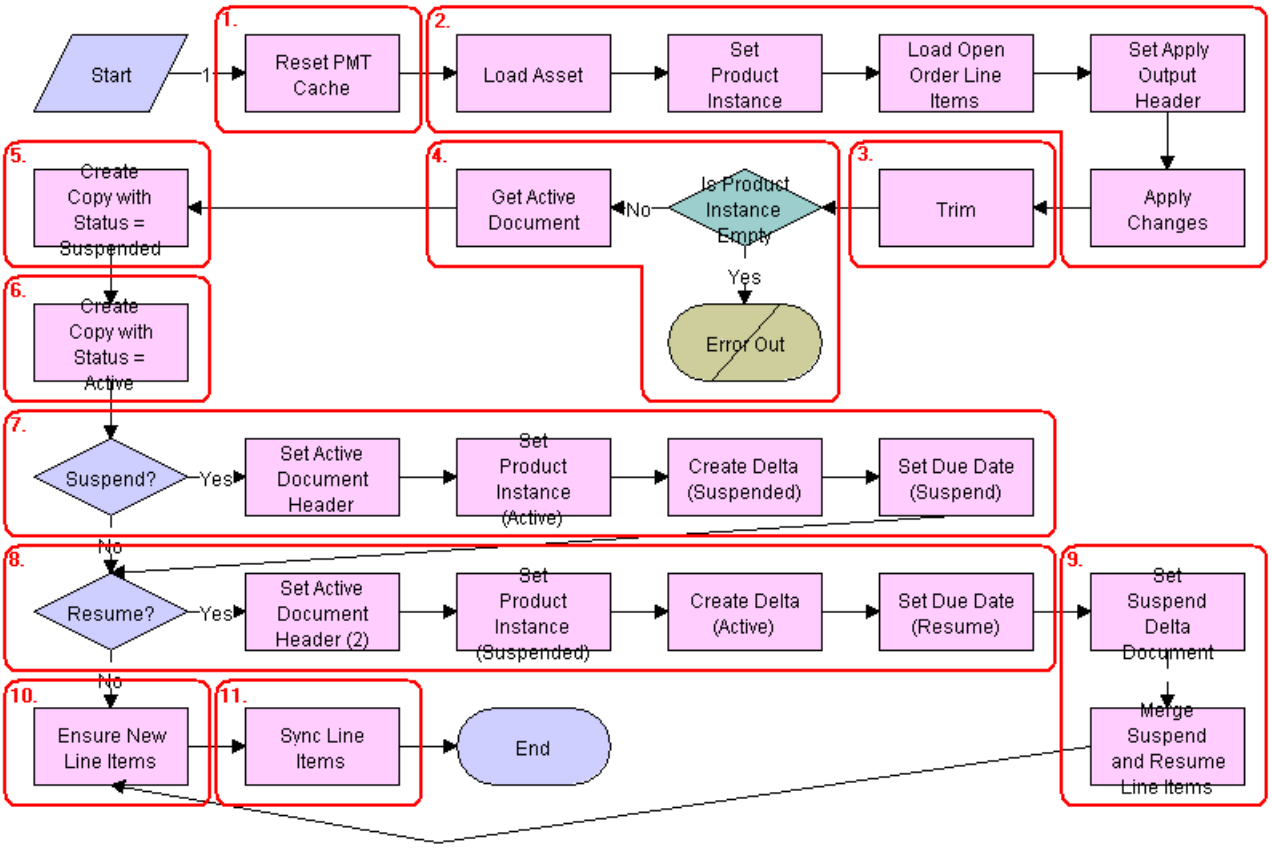
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Suspend / Resume Selections	Iterate Process For Selected Rows
Go to Document Details View	GotoView

SIS OM Suspend / Resume Asset Sub-process

This workflow applies the open and pending orders related to the selected asset to get its requested state. Then it creates line items to suspend or resume the selected asset in that state. It adds these line items to the active document. This workflow is shown in the following figure.

The SIS OM Suspend / Resume Products & Services Process initiates this workflow.



Workflow Description

This workflow does the following:

1. Clears the business service cache of existing product instances.
2. Reads the asset from the database.

   The workflow includes only components that do not have a status of Inactive. The workflow looks for all the orders with a status of Open or Pending that are related to the selected asset. The workflow applies the changes specified by those orders. This re-creates the state of the service product as the user has requested it.
3. Removes any components from the asset that do not have the 'Track As Asset' flag set.
4. Displays an error message if the asset is empty.

   If one of the open or pending orders disconnects the root component, or there are no components that have the 'Track As Asset' flag set, then the result is an empty asset. In this case, the workflow displays an error message.
5. Copies the asset and sets the status of each component to 'Suspended'.
6. Copies the asset and sets the status of each component to Active.
7. If the Suspend button was clicked, then the workflow generates the delta of the requested state of the asset and the future suspended state of the asset. This creates line items to suspend the asset. Sets the due date for each line item.
8. If the resume button was clicked, then the workflow generates the delta of the requested state of the asset and the future active state of the asset. This creates line items to resume the asset, and it sets the due date for each line item.
9. Merges the line items of the suspend and resume requests into a single document.
10. Generates a new asset integration Id for any line item with an action code of Add.

11. Saves the document to the database.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance	Set Product Instance
Load Open Order Line Items	Find Orders
Set Apply Output Header	Set Output Header
Apply Changes	Apply
Trim	Trim
Create Copy with Status = Suspended	Set Field Value
Create Copy with Status = Active	Set Field Value
Set Active Document Header	Set Output Header
Set Requested Product Instance (Active)	Set Product Instance
Create Delta (Suspended)	Delta
Set Due Date (Suspend)	Set Field Value
Set Active Document Header (2)	Set Output Header
Set Requested Product Instance (Suspended)	Set Product Instance
Create Delta (Active)	Delta
Set Due Date (Resume)	Set Field Value
Set Suspend Delta Document	Set Product Instance
Merge Suspend and Resume Line Items	Merge

Workflow Step	Business Service Method Called
Ensure New Line Item	Set Field Value
Sync Line Item	Synchronize

Workflows for Contact-Based Asset-Based Ordering

In addition to basing orders on accounts, users can base orders on contacts when they use asset-based ordering. In addition to the service profile (which lists assets) being associated with the account and a number of the ordering processes being initiated from the Account Summary View, the service profile can be associated with a contact, and the same ordering processes can be initiated from Contact Summary View, with the order placed for a contact rather than for an account.

The following workflows support contact-based asset-based ordering:

- SIS OM Active Order Sub-Process - Contact
- SIS OM Active Quote Sub-Process - Contact
- SIS OM Disconnect Asset Sub-process - Contact
- SIS OM Disconnect Products & Services Process - Contact
- SIS OM Edit Complex Asset Workflow - Contact
- SIS OM Modify Products & Services Process - Contact
- SIS OM New Products & Services Process - Contact
- SIS OM Suspend / Resume Asset Sub-Process - Contact
- SIS OM Suspend / Resume Products & Services Process - Contact

These workflows work in the same way as the equivalent workflows used for account-based asset-based ordering, which are covered earlier in this chapter.

Workflows for Quotes and Orders Assets Catalog Views

End users can use the Quote - Assets Catalog View and Orders - Assets Catalog View to display the assets associated with the Account specified in the quote or order header. The user can use these views to add a new line item to the quote or order that modifies, deactivates, suspends or resumes an asset. The following workflows are used by these views:

- SIS OM Disconnect Products & Services Process - Quote & Order
- SIS OM Modify Products & Service Process - Quote & Order
- SIS OM Suspend / Resume Products & Services Process - Quote & Order

These workflows work in the same way as the equivalent workflows used for account-based asset-based ordering, which are covered earlier in this chapter.

Note: These views apply only to account-based ABO assets. They do not apply to contact-based ABO.

Workflows that Support the Automatic Configuration of Customizable Products

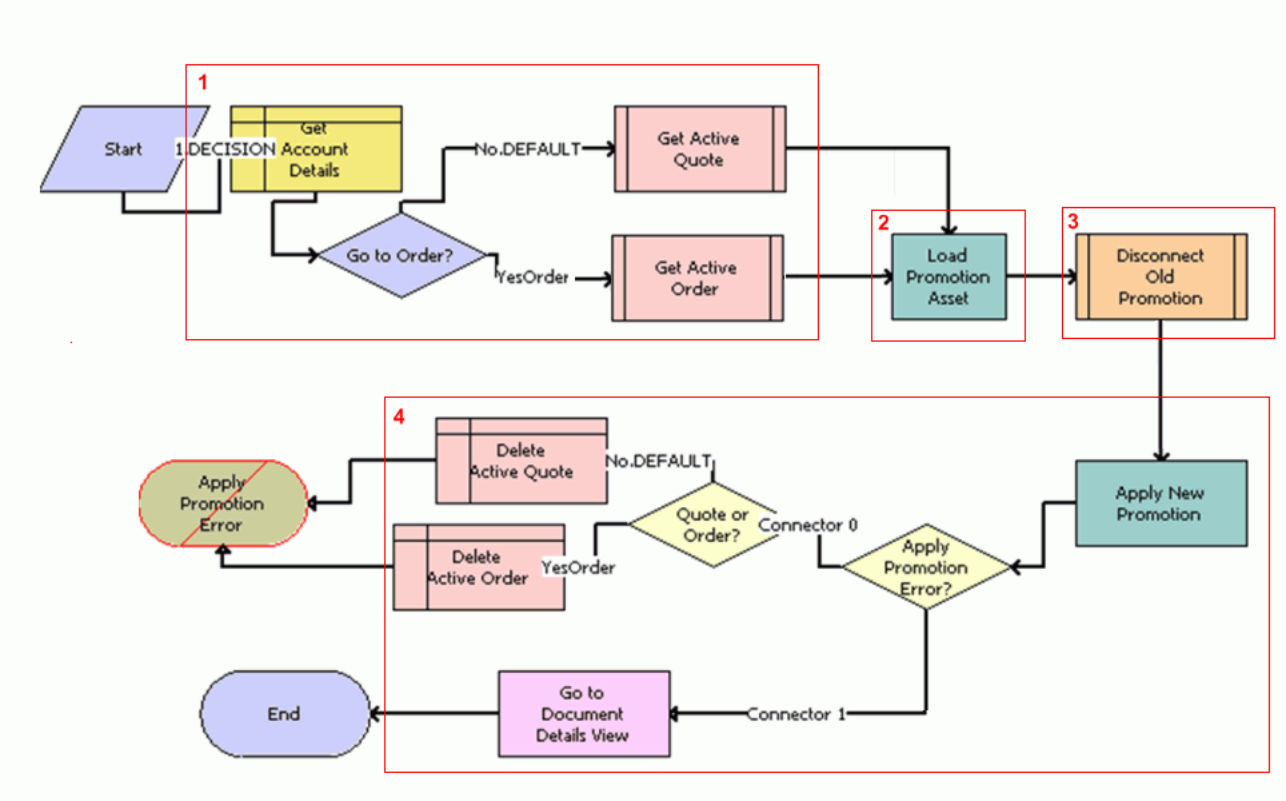
This group of workflows supports the automatic configuration of customizable products that are based on a promotion definition. In the automatic configuration, the bundled promotions contain customizable products that are added with the default configuration when you add a promotion line item to a quote or an order. This group of workflows consists of the following process and subprocesses:

- *ISS Promotion Upgrade Process*
- *Headless Configuration Web Service*
- *SIS OM Get Future Asset Subprocess*
- *SIS OM Sync Delta Subprocess*

For more information about setting up the automatic configuration of customizable products that are based on a promotion definition, see *Process of Enabling the Automatic Configuration of Customizable Products*.

ISS Promotion Upgrade Process

This workflow, shown in the following figure, upgrades one promotion to another promotion. ISS Promotion Upgrade Process is initiated by a user when the user upgrades an existing promotion.



Workflow Description

This workflow does the following:

1. **Get Account Details.** This step gets the account details, and:
   a. If linked to an order, gets the active order.
   b. If linked to a quote (which is the default), gets the active quote.
2. **Load Promotion Asset.** This step loads the promotion asset.
3. **Disconnect Old Promotion.** This step disconnects the old promotion.
4. **Apply New Promotion.** This step applies the new promotion, and:
   a. If errors are generated on applying the promotion, then it deletes the active document (order or quote) obtained in Step 1.
   b. Otherwise, it goes to the document details view.

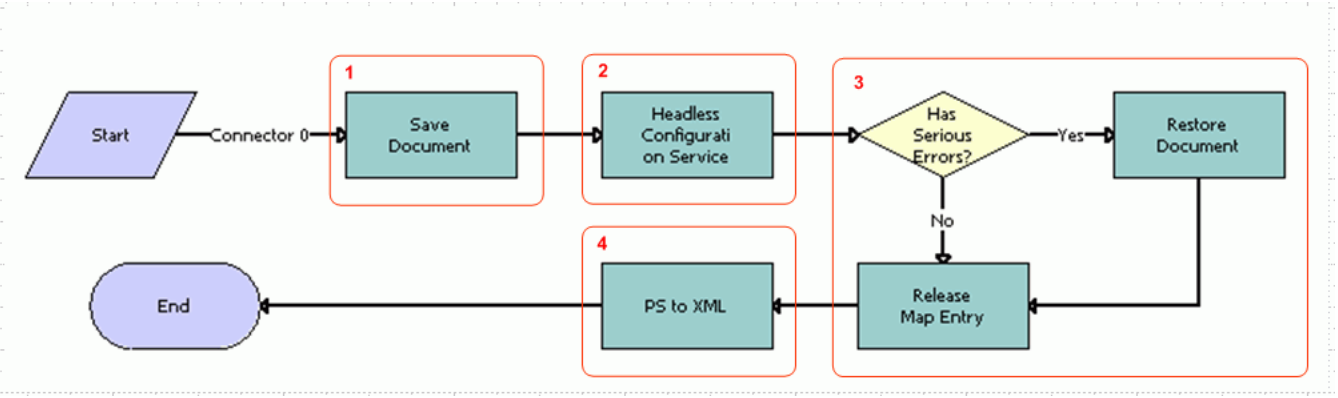
Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Load Promotion Asset	LoadPromAsset
Apply New Promotion	ApplyPromotion

Headless Configuration Web Service

The Headless Configuration Web Service is a wrapper workflow for using Promotion AutoMatch functionality from Web services. This workflow, shown in the following figure, calls the Headless Configuration Service.



Workflow Description

This workflow does the following:

1. **Save Document.** This step saves the current document.
2. **Headless Configuration Service.** This step calls the Headless Configuration Service.
3. **Has Serious Errors?** This step determines whether errors have been generated:
    a. If errors were generated, then the document is restored to its original format.
    b. If no errors were generated, then the map entry is released.
4. **PS to XML.** This step converts output property set to XML.

Associated Business Service Methods

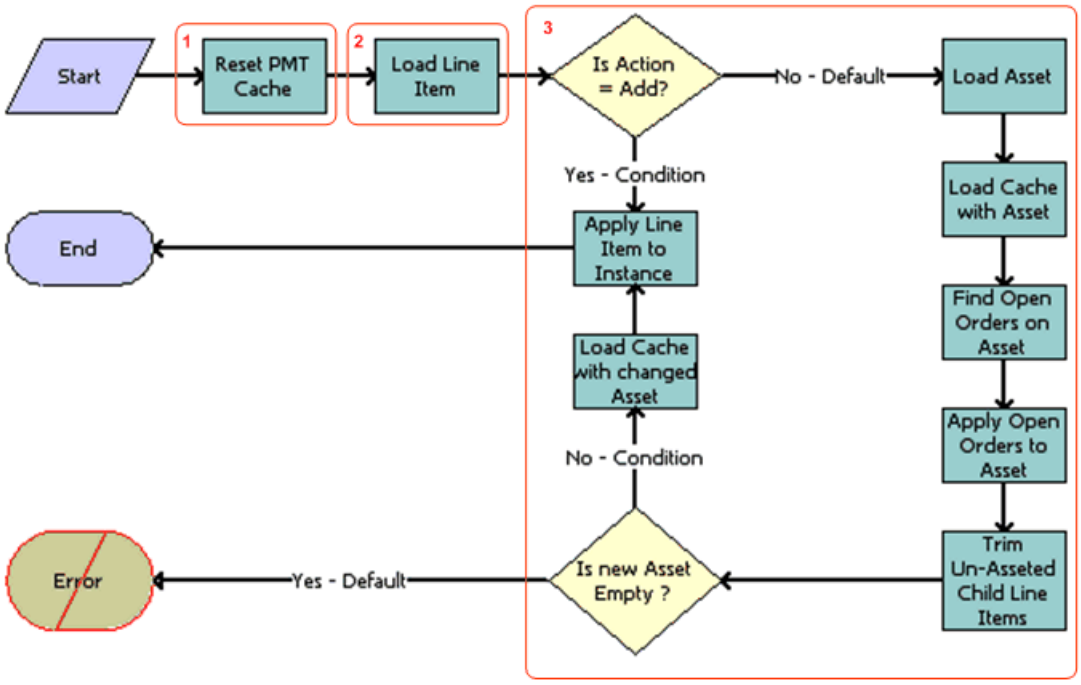
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Save Document	Save Document
Headless Configuration Service	Apply Promotion Defaults
Restore Document	Restore Document
Release Map Entry	Release Map Entry
PS to XML	PropSetToXML

SIS OM Get Future Asset Subprocess

This workflow, shown in the following figure, is called by the *Headless Configuration Service* to get the future asset and the original asset. The future asset is the result of applying the current line item and any open orders to the original asset. This workflow returns the asset in its future state, the asset in its current state, and the document header with the root line item.

This workflow is called by the HdlsCfgSvcLoadABO signal.



Workflow Description

This workflow does the following:

1. **Reset PMT Cache.** This step resets the cache to prevent an incomplete workflow from entering the PMT cache.
2. **Load Line Item.** This step loads the line item.

3. **Is Action equal to Add?** This step checks the action code:
    a. If action code is Yes (that is, Add), then it applies the line item to the instance.
    b. If action code is No (which is the default), then:
        - It loads the asset.
        - It loads the cache with the asset.
        - It finds open orders on the asset.
        - It applies open orders to the asset.
        - It deletes child line items (that are not part of the asset).
        - It checks if the new asset is empty. If the new asset is empty (which is the default), then an error is generated. If the new asset is not empty, then loads the cache with the changed item and applies the line item to the instance.

Associated Business Service Methods

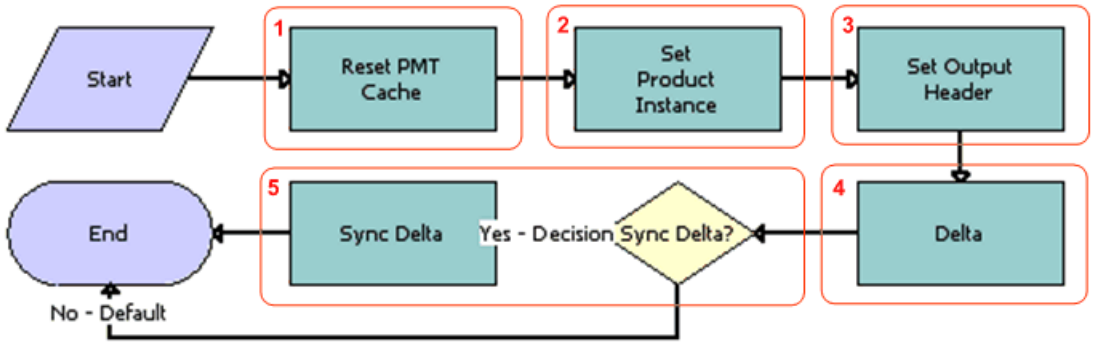
The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Load Line Item	Query
Load Asset	Query
Load Cache with Asset	Set Product Instance
Find Open Orders on Asset	Find Orders
Apply Open Orders to Asset	Apply
Trim Un-Asseted Child Line Items	Trim
Load Cache with Changed Asset	Set Product Instance
Apply Line Item to Instance	Apply

SIS OM Sync Delta Subprocess

This workflow, shown in the following figure, is called by the *Headless Configuration Service* to synchronize the changes to the customizable product line item. The asset based ordering difference between the batch validated asset and the original asset is calculated, and that is applied to the document.

This workflow is called by the HdlsCfgSvcSyncABO signal.



Workflow Description

This workflow does the following:

1. **Reset PMT Cache.** This step resets the cache to prevent an incomplete workflow from entering the PMT cache.
2. **Set Product Instance.** This step sets the product instance.
3. **Set Output Header.** This step sets the output header.
4. **Delta.** This step calls the Delta method of the SIS OM PMT Service to compare the modified asset against the original asset.
5. **Sync Delta?** This step checks if the decision is to synchronize the change:
   - If the decision to synchronize is Yes, then synchronizes the change.

     This decision is controlled by the Sync input argument of the Headless Configuration Service business service.
   - If the decision to synchronize is No (which is the default), then the process ends.

Associated Business Service Methods

The following table shows the steps in this workflow that call business service methods.

Workflow Step	Business Service Method Called
Reset PMT Cache	Reset
Set Product Instance	Set Product Instance
Set Output Header	Set Output Header

Workflow Step	Business Service Method Called
Delta	Delta
Sync Delta	Synchronize

20 Product Data Service and Import Export API Reference

Product Data Service and Import Export API Reference

This chapter covers the Product Data Service API and the ISS Authoring Import Export Business Service. It includes the following topics:

- *About the Product Data Service API*
- *Product Data Services API Architecture*
- *Product Data Services API Methods Reference*
- *PDS PublishCatalog Test Sample Workflow*
- *ISS Authoring Import Export Business Service Reference*

About the Product Data Service API

The Product Data Service API allows you to use the catalog, product selection, and product search capabilities of the Siebel Business Application without using the Siebel Business Application user interface.

These APIs give you complete flexibility to build the user interface. You can use your own tool to design the user interface. Your user interface is completely independent of the Siebel Business Application user interface, but you can use these APIs to call all the functionality that the Siebel Business Application uses to work with products, including eligibility and catalog search.

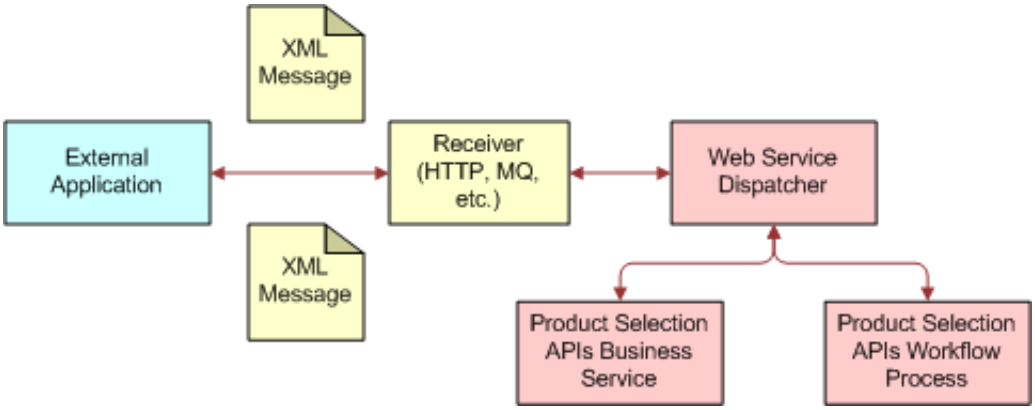
Product Data Services API Architecture

The Product Data Service API is provided as Siebel business services and workflows inside the Siebel Business Application. The Siebel Web Services Framework can consume a WSDL document and create a proxy business service to publish Siebel business services and workflows as Web Services. Once published as Web Services, all the external applications are able to access these product selection APIs through supported protocols (HTTP, MQ, and so on) by exchanging XML messages.

In the Siebel Business Application, the Data Services API includes:

- **Data Services API Methods.** These methods allow you to access Siebel Data, such as Catalog, Categories, Products, Product Details, Favorites, Quote, and Order. They are based on the core Data Sync service (CSSEAIDataSyncService).
- **Workflows.** Workflows use these methods to add an asset to a quote or order, modify or disconnect asset from quote or order, and get projected assets without any use of the user interface.

This architecture is shown in the following figure.



Product Data Services API Methods Reference

The Product Data Service APIs include the following methods:

Catalog API:

- *GetCatalogs Method*
- *GetCatalogCategories Method*
- *GetCategoryProducts Method*
- *GetProductDetails Method*
- *PublishCatalog Method*
- *GetFavorites Method*
- *GetFavoriteItems Method*
- Product Details API
  - *GetRelatedProducts Method*
  - *GetRelatedPromotions Method*
  - *GetProductAttributeDomain Method*
  - *GetProductChildren Method*
- Search API
  - *GetSearchOptions Method*
  - *GetSearchParameters Method*
  - *ExecuteSearch Method*
- Add Favorite API
  - *AddFavoriteToQuote, AddFavoriteToOrder, AddFavoriteItemtoQuote, AddFavoriteItemtoOrder, SaveQuoteAsFavorite, and SaveOrderAsFavorite Methods*
- Quote / Order API
  - *DeleteQuote Method*

- *InsertQuote Method*
- *UpdateQuote Method*
- *DeleteQuoteItem Method*
- *InsertQuoteItem Method*
- *UpdateQuoteItem Method*
- *DeleteOrder Method*
- *InsertOrder Method*
- *UpdateOrder Method*
- *DeleteOrderItem Method*
- *InsertOrderItem Method*
- *UpdateOrderItem Method*
- *GetActiveQuote Method*
- *SetActiveQuote Method*
- *GetActiveOrder Method*
- *SetActiveOrder Method*

GetCatalogs Method

Gets a list of catalogs for a given user name and context information. External.

Syntax

Product Data Service.InvokeMethod(“GetCatalogs", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Catalog Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Catalog Interface)

Properties of Input Arguments

Property	Type	Value	Description
SiebelMessage	Integration Object	PDS Catalog Interface	None.

Example of Input Arguments

```
PropertySet@0012C868 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AA6D8F0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A9D00C0 p#0 c#1 type="ListOfPDS Catalog Interface" vt=0 value=""
 {
 c[0] PropertySet@0A9CF930 p#1 c#0 type="Catalog" vt=0 value=""
 {
 p["Name"] = "PDSCatalog";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AAA9240 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1JIU";
 p["IntObjectName"] = "PDS Catalog Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A76FAC8 p#0 c#1 type="ListOfPDS Catalog Interface" vt=0 value=""
 {
 c[0] PropertySet@0A76AFC0 p#11 c#0 type="Catalog" vt=0 value=""
 {
 p["Effective End Date"] = "";
 p["Id"] = "42-54R7R";
 p["Sequence Number"] = "";
 p["ThumbnImageFileName"] = "";
 p["Private Flag"] = "N";
 p["Version"] = "1";
 p["Effective Start Date"] = "07/21/2006 17:11:21";
 p["Catalog Type"] = "Buying";
 p["Active"] = "Y";
 p["Name"] = "PDSCatalog";
 p["Description"] = "";
 }
 }
 }
}
```

Error Conditions

None

GetCatalogCategories Method

Gets catalog categories for a given catalog. External

Syntax

Product Data Service.InvokeMethod(“GetCatalogCategories", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Catalog Category Interface")
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Catalog Category Interface)

Example of Input Arguments

```
PropertySet@0012C868 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0A782558 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Category Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AA70F08 p#0 c#1 type="ListOfPDS Catalog Category Interface"
vt=0 value=""
 {
 c[0] PropertySet@0AA73710 p#1 c#0 type="Catalog" vt=0 value=""
 {
 p["Name"] = "PDSCatalog";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AAC66C0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1JIV";
 p["IntObjectName"] = "PDS Catalog Category Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AB22E28 p#0 c#1 type="ListOfPDS Catalog Category Interface"
vt=0 value=""
 {
 c[0] PropertySet@0AAD8E68 p#11 c#1 type="Catalog" vt=0 value=""
 {
 p["Effective End Date"] = "";
 p["Id"] = "42-54R7R";
 p["Sequence Number"] = "";
 p["ThumbnImageFileName"] = "";
 p["Private Flag"] = "N";
 p["Version"] = "1";
 p["Effective Start Date"] = "07/21/2006 17:11:21";
 p["Catalog Type"] = "Buying";
 p["Active"] = "Y";
```

```
p["Name"] = "PDSCatalog";
p["Description"] = "";
c[0] PropertySet@0AAC66F0 p#0 c#2 type="ListOfCategory" vt=0 value=""
{
c[0] PropertySet@0AACFF00 p#20 c#0 type="Category" vt=0 value=""
{
p["Effective End Date"] = "";
p["Active Flag"] = "Y";
p["Id"] = "42-54R7S";
p["No Training Flag"] = "N";
p["Sequence Number"] = "";
p["ThumbnImageFileName"] = "";
p["Private Flag"] = "N";
p["Catalog Id"] = "42-54R7R";
p["Parent Category Id"] = "";
p["Effective Start Date"] = "07/21/2006 17:12:35";
p["Response Threshold"] = "";
p["Display Name"] = "PDSCatalogCategory1.1";
p["Usage"] = "";
p["Parent Category Name"] = "";
p["Count"] = "3";
p["Root Category Flag"] = "Y";
p["Root Category Id"] = "";
p["Name"] = "PDSCatalogCategory1.1";
p["Display Template"] = "";
p["Description"] = "";
}
c[1] PropertySet@0AACAF30 p#20 c#0 type="Category" vt=0 value=""
{
p["Effective End Date"] = "";
p["Active Flag"] = "Y";
p["Id"] = "42-54R7T";
p["No Training Flag"] = "N";
p["Sequence Number"] = "";
p["ThumbnImageFileName"] = "";
p["Private Flag"] = "N";
p["Catalog Id"] = "42-54R7R";
p["Parent Category Id"] = "";
p["Effective Start Date"] = "07/21/2006 17:13:28";
p["Response Threshold"] = "";
p["Display Name"] = "PDSCatalogCategory1.2";
p["Usage"] = "";
p["Parent Category Name"] = "";
p["Count"] = "5";
p["Root Category Flag"] = "Y";
p["Root Category Id"] = "";
p["Name"] = "PDSCatalogCategory1.2";
p["Display Template"] = "";
p["Description"] = "";
}
}
}
}
}
}
```

Error Conditions

Error ID	Error Message Text	User Action/Resolution
IDS_ERR_ANIESALES_MISSING_SERVICE_NAME	The '%1' user property is not defined in business service '%2'.	Define user prop Data Sync Service (GetCatalogCategories) in Product Selection Service.

Error ID	Error Message Text	User Action/Resolution
SSAOMErrCannotCreateService	Unable to create the Business Service '%1	Check to see if the business service defined in user prop Data Sync Service (GetCatalogCategories) in Product Selection Service is defined correctly.

GetCategoryProducts Method

Gets a list of products for given catalog, category and context. External

Syntax

Product Data Service.InvokeMethod(“GetCategoryProducts", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Category Product Interface) String: EligibilityMode String: PricingMode String: ContextMode (Optional) Hierarchy: SiebelMessage (Integration object PDS Catalog Context)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Category Product Interface)

Example of Input Arguments

```
PropertySet@0012CCC4 p#3 c#2 type="" vt=0 value=""
{
 p["EligibilityMode"] = "1";
 p["ContextMode"] = "SiebelContext";
 p["PricingMode"] = "Y";
 c[0] PropertySet@0AA66E90 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Catalog Context";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A7A84C0 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0AB68BA8 p#0 c#1 type="Context" vt=0 value=""
 {
```

```
 c[0] PropertySet@0A9DA050 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0AB23A70 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0AA997F0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Category Product Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A77C290 p#0 c#2 type="ListOfPDS Category Product Interface"
vt=0 value=""
 {
 c[0] PropertySet@0A993EE0 p#1 c#0 type="Catalog Category" vt=0 value=""
 {
 p["Name"] = "PDSCatalogCategory1.2";
 }
 c[1] PropertySet@0AA89B50 p#1 c#0 type="Catalog Category" vt=0 value=""
 {
 p["Name"] = "PDSCatalogCategory1.1";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CC58 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@1078D580 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-WAQ";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = " PSS Category Product Interface ";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0BD55380 p#0 c#1 type="ListOfPSS Category Product Interface"
vt=0 value=""
 {
 c[0] PropertySet@0BD82770 p#4 c#1 type="Catalog Category" vt=0 value=""
 {
 p["Catalog Name"] = "ERM Catalog";
 p["Version"] = "1";
 p["Description"] = "ERM Category";
 p["Name"] = "ERM Category";
 p["Description - Translation"] = "ERM Category";
 p["[Field]"] = "[Value]";
 c[0] PropertySet@0BF21F30 p#0 c#1 type="ListOfInternal Product" vt=0
value=""
 {
 c[0] PropertySet@0BD2D7A0 p#85 c#2 type="Internal Product" vt=0 value=""
 {
 p["Name"] = "xxx";
 p["[Field]"] = "[Value]";
 }
 c[…] PropertySet@0BD2D7A0 p#85 c#2 type="Internal Product" vt=0 value=""
 {
```

```
p["Name"] = "xxx";
p["[Field]"] = "[Value]";
}
}
}
}
}
}
```

Error Conditions

Error ID	Error Message Text	User Action/Resolution
IDS_ERR_ANIESALES_MISSING_SERVICE_NAME	The '%1' user property is not defined in business service '%2'.	Need to define user prop Data Sync Service (GetCategoryProducts) in Product Selection Service.
SSAOMErrCannotCreateService	Unable to create the Business Service '%1	Check to see if the business service defined in user prop Data Sync Service (GetCategoryProducts) in Product Selection Service is defined correctly.

GetProductDetails Method

Gets a list of product details for given context. External.

Syntax

Product Data Service.InvokeMethod(“GetProductDetails", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Product Details Interface) String: ContextMode (Optional) String: EligibilityMode String: PricingMode Hierarchy: SiebelMessage (Integration object PDS Catalog Context)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Details Interface)

Example of Input Arguments

```
PropertySet@0012C6BC p#3 c#2 type="" vt=0 value=""
{
 p["EligibilityMode"] = "";
 p["ContextMode"] = "SiebelContext";
 p["PricingMode"] = "Y";
 c[0] PropertySet@0A9DB3E0 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Context";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A775E38 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0B23B2C8 p#0 c#1 type="Context" vt=0 value=""
 {
 c[0] PropertySet@0AAAC308 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0AC022C8 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0ABC6868 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Product Details Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AAB6410 p#0 c#1 type="ListOfPDS Product Details Interface"
vt=0 value=""
 {
 c[0] PropertySet@0B639560 p#1 c#0 type="Product" vt=0 value=""
 {
 p["Id"] = "99-28W7U";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AD67060 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1JLM";
 p["IntObjectName"] = "PDS Product Details Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A9A1C88 p#0 c#1 type="ListOfPDS Product Details Interface"
vt=0 value=""
 {
 c[0] PropertySet@0AD72870 p#23 c#1 type="Product" vt=3 value="99-28W7U"
 {
 p["Class Id"] = "99-28VEN";
 p["Vendor Integration Id"] = "";
 p["Reference Price"] = "";
```

```
 p["Product Def Type Code"] = "Customizable";
 p["Eligibility Reason"] = "";
 p["Id"] = "99-28W7U";
 p["Product Id"] = "99-28W7U";
 p["Pre Pick"] = "N";
 p["Part #"] = "GPAP";
 p["Eligibility Status"] = "";
 p["Integration Id"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Parent Item Id"] = "";
 p["Net Price"] = "0";
 p["List Price"] = "0";
 p["Effective To"] = "";
 p["Product Line Id"] = "No Match Row Id";
 p["Product Configuration Model Id"] = "99-28W7U";
 p["Effective From"] = "";
 p["List Price Type Code"] = "STANDARD";
 p["Name"] = "CDA_gp_Audio Package";
 p["Quantity"] = "1";
 p["Price List Id"] = "1-31C5";
 c[0] PropertySet@0AD5A848 p#0 c#1 type="ListOfProduct Attribute" vt=0
value=""
 {
 c[0] PropertySet@0AD58DA8 p#5 c#0 type="Product Attribute" vt=0 value=""
 {
 p["Data Type"] = "Text";
 p["Value"] = "Alpine";
 p["Default Value"] = "";
 p["Display Name"] = "Brand";
 p["Name"] = "Brand";
 }
 }
 }
 }
 }
}
```

Error Conditions

Error ID	Error Message Text	User Action/Resolution
IDS_ERR_ANIESALES_MISSING_SERVICE_NAME	The '%1' user property is not defined in business service '%2'.	Define user prop Data Sync Service (GetProductDetails) in Product Selection Service.
SSAOMErrCannotCreateService	Unable to create the Business Service '%1	Check to see if the business service defined in user prop Data Sync Service (GetProductDetails) in Product Selection Service is defined correctly.

PublishCatalog Method

Gets a list of catalogs details including all Categories, all available products under each of them and product details for a given catalog and context. External

Syntax

Product Data Service.InvokeMethod(“PublishCatalog", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: EligibilityMode String: ContextMode String: PricingMode String: IncludePriceWaterfall: Y/N (includes price waterfall or not) Hierarchy: SiebelMessage (Integration object PDS Catalog Context) Hierarchy: SiebelMessage (Integration object PDS Publish Catalog)
Output Arguments	Out	Hierarchy: SiebelMessage

Properties of Input Arguments

Property	Type	Value	Description
EligibilityMode	String	0,1,2	0:No EligibilityCheck 1:Check Eligibility without Filtering non-eligible Products 2:Check Eligbility by filtering non-eligible products
ContextMode	String	SiebelContext, ExternalContext	SiebelContext: Using existing Quote/Order Information in Siebel Database. In this case, the API will construct the Context. ExternalContext: External Call will provide the context information, API will pass them directly to the Pricing and Eligibility Process.
PricingMode	String	Y/N	None
Context	Hierarchy	See the following examples.	None
SiebelMessage	Integration Object	PDS Publish Catalog Interface	None

Example of Input Arguments

```
PropertySet@0012C73C p#4 c#2 type="" vt=0 value=""
{
 p["EligibilityMode"] = "";
 p["ContextMode"] = "SiebelContext";
 p["PricingMode"] = "Y";
 p["IncludePriceWaterfall"] = "Y";
 c[0] PropertySet@0A7ADFF8 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Context";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A93C710 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0A7DBE20 p#0 c#1 type="Context" vt=0 value=""
 {
 c[0] PropertySet@0A711D30 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0AB2C880 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0A92AB50 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Publish Catalog Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AB32600 p#0 c#1 type="ListOfPDS Publish Catalog Interface"
vt=0 value=""
 {
 c[0] PropertySet@0A7E4F60 p#1 c#0 type="Catalog" vt=0 value=""
 {
 p["Name"] = "PDSCatalog";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0B58ECD8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1NM2";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Publish Catalog Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0B86A9C8 p#0 c#1 type="ListOfPDS Publish Catalog Interface"
vt=0 value=""
 {
 c[0] PropertySet@0AA33768 p#11 c#1 type="Catalog" vt=0 value=""
 {
 p["Effective End Date"] = "";
 p["Id"] = "42-54R7R";
```

```
 p["Sequence Number"] = "";
 p["Private Flag"] = "N";
 p["ThumbnImageFileName"] = "";
 p["Effective Start Date"] = "07/21/2006 17:11:21";
 p["Version"] = "1";
 p["Catalog Type"] = "Buying";
 p["Active"] = "Y";
 p["Description"] = "";
 p["Name"] = "PDSCatalog";
 c[0] PropertySet@0B7B8348 p#0 c#2 type="ListOfCategory" vt=0 value=""
 {
 c[0] PropertySet@0C18AFF8 p#20 c#1 type="Category" vt=0 value=""
 {
 p["Effective End Date"] = "";
 p["Id"] = "42-54R7S";
 p["Active Flag"] = "Y";
 p["No Training Flag"] = "N";
 p["Sequence Number"] = "";
 p["Catalog Id"] = "42-54R7R";
 p["Private Flag"] = "N";
 p["ThumbnImageFileName"] = "";
 p["Effective Start Date"] = "07/21/2006 17:12:35";
 p["Parent Category Id"] = "";
 p["Display Name"] = "PDSCatalogCategory1.1";
 p["Response Threshold"] = "";
 p["Usage"] = "";
 p["Count"] = "3";
 p["Parent Category Name"] = "";
 p["Root Category Flag"] = "Y";
 p["Description"] = "";
 p["Display Template"] = "";
 p["Name"] = "PDSCatalogCategory1.1";
 p["Root Category Id"] = "";
 c[0] PropertySet@0B7ADAE0 p#0 c#3 type="ListOfProduct" vt=0 value=""
 {
 c[0] PropertySet@0A785780 p#19 c#2 type="Product" vt=3
value="99-28W7U"
 {
 p["Class Id"] = "99-28VEN";
 p["Product Def Type Code"] = "Customizable";
 p["Id"] = "99-28W7U";
 p["Eligibility Reason"] = "";
 p["Pre Pick"] = "N";
 p["Defective Product State"] = "";
 p["Product Id"] = "99-28W7U";
 p["Cost List Item Currency Code"] = "";
 p["Eligibility Status"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Parent Item Id"] = "";
 p["Net Price"] = "0";
 p["Effective To"] = "";
 p["List Price"] = "0";
 p["Product Configuration Model Id"] = "99-28W7U";
 p["Product Line Id"] = "No Match Row Id";
 p["Effective From"] = "";
 p["Quantity"] = "";
 p["Name"] = "CDA_gp_Audio Package";
 c[0] PropertySet@0B817CF8 p#0 c#1 type="ListOfISS Product
Attribute VBC" vt=0
value=""
 {
 c[0] PropertySet@0B7F9FD8 p#5 c#0 type="ISS Product
Attribute VBC" vt=0
value=""
 {
 p["Data Type"] = "Text";
```

```
 p["Default Value"] = "Alpine";
 p["Attribute Name"] = "CDA_gp_BrandText";
 p["Display Name"] = "Alpine";
 p["Name"] = "Brand";
 }
 }
 c[1] PropertySet@0B7FF410 p#0 c#0 type="ListOfNet Price
Waterfall" vt=0 value=""
 }
 }
 c[1] PropertySet@0B79A670 p#20 c#1 type="Category" vt=0 value=""
 {
 p["Effective End Date"] = "";
 p["Id"] = "42-54R7T";
 p["Active Flag"] = "Y";
 p["No Training Flag"] = "N";
 p["Sequence Number"] = "";
 p["Catalog Id"] = "42-54R7R";
 p["Private Flag"] = "N";
 p["ThumbnImageFileName"] = "";
 p["Effective Start Date"] = "07/21/2006 17:13:28";
 p["Parent Category Id"] = "";
 p["Display Name"] = "PDSCatalogCategory1.2";
 p["Response Threshold"] = "";
 p["Usage"] = "";
 p["Count"] = "5";
 p["Parent Category Name"] = "";
 p["Root Category Flag"] = "Y";
 p["Description"] = "";
 p["Display Template"] = "";
 p["Name"] = "PDSCatalogCategory1.2";
 p["Root Category Id"] = "";
 c[0] PropertySet@09464A70 p#0 c#5 type="ListOfProduct" vt=0 value=""
 {
 c[0] PropertySet@0B7946E8 p#19 c#2 type="Product" vt=3
value="99-28W7F"
 {
 p["Class Id"] = "";
 p["Product Def Type Code"] = "None";
 p["Id"] = "99-28W7F";
 p["Eligibility Reason"] = "";
 p["Pre Pick"] = "N";
 p["Defective Product State"] = "";
 p["Product Id"] = "99-28W7F";
 p["Cost List Item Currency Code"] = "";
 p["Eligibility Status"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Parent Item Id"] = "";
 p["Net Price"] = "1";
 p["Effective To"] = "";
 p["List Price"] = "1";
 p["Product Configuration Model Id"] = "99-28W7F";
 p["Product Line Id"] = "No Match Row Id";
 p["Effective From"] = "";
 p["Quantity"] = "";
 p["Name"] = "CDA_gp_Key";
 c[0] PropertySet@0AEBB400 p#0 c#0 type="ListOfISS Product
Attribute VBC" vt=0
value=""
 c[1] PropertySet@0554B198 p#0 c#0 type="ListOfNet Price
Waterfall" vt=0 value=""
 }
 }
 }
 }
 }
```

```
    }
   }
  }
```

Error Conditions

None.

GetFavorites Method

Get a list quote or order favorites using search specification. External

Syntax

Product Data Service.InvokeMethod(“GetFavorites", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Template Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Template Interface)

Error Conditions

Error ID	Error Message Text	User Action/Resolution
IDS_ERR_ANIESALES_MISSING_SERVICE_NAME	The '%1' user property is not defined in business service '%2'.	Define user prop Data Sync Service (GetFavorites) in Product Selection Service.
SSAOMErrCannotCreateService	Unable to create the Business Service '%1	Check to see if the business service defined in user prop Data Sync Service (GetFavorites) in Product Selection Service is defined correctly.

GetFavoriteItems Method

Get a list of favorite items for a given favorite. External

Syntax

Product Data Service.InvokeMethod(“GetFavoriteItems", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Template Item Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Template Item Interface)

Error Conditions

Error ID	Error Message Text	User Action/Resolution
IDS_ERR_ANIESALES_MISSING_SERVICE_NAME	The '%1' user property is not defined in business service '%2'.	Define user prop Data Sync Service (GetFavoriteItems) in Product Selection Service.
SSAOMErrCannotCreateService	Unable to create the Business Service '%1	Check to see if the business service defined in user prop Data Sync Service (GetFavoriteItems) in Product Selection Service is defined correctly.

GetRelatedProducts Method

Gets related products. External.

Syntax

Product Data Service.InvokeMethod(“GetRelatedProducts", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Related Products Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Related Products Interface)

Example of Input Arguments

```
PropertySet@0012C5B0 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@1088CCE8 p#4 c#1 type="SearchSpec" vt=0 value=""
{
 p["SearchSpc"] = "xxx"
 }
}
```

Example of Output Arguments

```
PropertySet@0012C5B0 p#0 c#1 type="" vt=0 value=""
{
 c[1] PropertySet@1088CCE8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = " PSS Related Products";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@1088CCE8 p#4 c#1 type="Internal Product" vt=0 value=""
{
c[0] PropertySet@108797E8 p#0 c#1 type="ListOfPDS Related Products" vt=0 value=""
 {
 c[0] PropertySet@1088CCE8 p#4 c#1 type=" Component Product" vt=0 value=""
{
 }
 c[1] PropertySet@1088CCE8 p#4 c#1 type=" Component Product" vt=0 value=""
{
 }
 }
}
}
 }
```

Error Conditions

None.

GetRelatedPromotions Method

Gets related promotions. External.

Syntax

Product Data Service.InvokeMethod(“GetRelatedPromotions", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Product Details Interface) String: ContextMode (Optional) String: EligibilityMode

Argument	Type	Description
		String: PricingMode Hierarchy: SiebelMessage (Integration object PDS Catalog Context)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Related Promotions Interface)

Example of Input Arguments

```
PropertySet@0012CCC4 p#3 c#2 type="" vt=0 value=""
{
 p["EligibilityMode"] = "";
 p["ContextMode"] = "SiebelContext";
 p["PricingMode"] = "Y";
 c[0] PropertySet@0AA62270 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Catalog Context";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A7BE0A8 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0AB2D9E0 p#0 c#1 type="Context" vt=0 value=""
 {
 c[0] PropertySet@0A9AD8D0 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0A995050 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0A7C6A88 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Related Promotions Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AAFB870 p#0 c#1 type="ListOfPDS Related Promotions Interface"
vt=0 value=""
 {
 c[0] PropertySet@0B747730 p#1 c#1 type="Product" vt=0 value=""
 {
 p["Name"] = "InMotion 130XL";
 c[0] PropertySet@0AA473D0 p#0 c#1 type="ListOfRelated Promotion" vt=0
value=""
 {
 c[0] PropertySet@0AA29BF8 p#1 c#0 type="Related Promotion" vt=0 value=""
 {
 p["Name"] = "*";
 }
 }
 }
```

```
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AAEBC20 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1NRM";
 p["IntObjectName"] = "PDS Related Promotions Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0ABAEE98 p#0 c#1 type="ListOfPDS Related Promotions Interface"
vt=0 value=""
 {
 c[0] PropertySet@0AC41580 p#23 c#1 type="Product" vt=0 value=""
 {
 p["Class Id"] = "1-PJLX";
 p["Vendor Integration Id"] = "";
 p["Reference Price"] = "";
 p["Product Def Type Code"] = "None";
 p["Eligibility Reason"] = "";
 p["Id"] = "1-ZMH";
 p["Product Id"] = "1-ZMH";
 p["Pre Pick"] = "N";
 p["Part #"] = "IM-130";
 p["Eligibility Status"] = "";
 p["Integration Id"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Parent Item Id"] = "";
 p["Net Price"] = "";
 p["List Price"] = "";
 p["Effective To"] = "";
 p["Product Line Id"] = "1-3HZA";
 p["Product Configuration Model Id"] = "1-ZMH";
 p["Effective From"] = "01/01/1999";
 p["List Price Type Code"] = "STANDARD";
 p["Name"] = "InMotion 130XL";
 p["Quantity"] = "1";
 p["Price List Id"] = "1-1D79";
 c[0] PropertySet@0AC02DC8 p#0 c#2 type="ListOfRelated Promotion" vt=0
value=""
 {
 c[0] PropertySet@0AC68EB0 p#5 c#0 type="Related Promotion" vt=0 value=""
 {
 p["Eligibility Reason"] = "";
 p["Eligibility Status"] = "";
 p["Promotion Id"] = "42-519PJ";
 p["Name"] = "Test Promotion1";
 p["Description"] = "";
 }
 c[1] PropertySet@0AC4DD68 p#5 c#0 type="Related Promotion" vt=0 value=""
 {
 p["Eligibility Reason"] = "";
 p["Eligibility Status"] = "";
 p["Promotion Id"] = "42-53MU1";
 p["Name"] = "Z PROM1";
 p["Description"] = "";
 }
 }
 }
 }
 }
```

```
}
```

Error Conditions

None.

GetProductAttributeDomain Method

Gets all the attribute types and domain details for a given product. External.

Syntax

Product Data Service.InvokeMethod(“GetProductAttributeDomain", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Attribute Id String: Product Configuration Model Id Hierarchy: SiebelMessage (Integration object PDS Product Attribute Domain Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Attribute Domain Interface)

Example of Input Arguments

```
PropertySet@0012CCC4 p#2 c#0 type="" vt=0 value=""
{
 p["Product Configuration Model Id"] = "99-27KMW";
 p["Attribute Id"] = "OM_Edition";
}
```

Example of Output Arguments:

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0A767970 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Product Attribute Domain Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";

 c[0] PropertySet@0AA19870 p#0 c#1 type="ListOfPDS Product Attribute Domain
Interface" vt=0 value=""
 {
 c[0] PropertySet@0AB28978 p#26 c#1 type="Product Attribute" vt=0 value=""
 {
 p["Unit of Measure"] = "";
 p["Id"] = "OM_Edition";
 p["Attribute Definition Id"] = "99-27KFN";
```

```
 p["Default Value"] = "";
 p["Eligibility Procedure Name"] = "";
 p["Vector Flag"] = "Y";
 p["Minimum Value"] = "";
 p["Attribute Name"] = "OM_Edition";
 p["Increment Value"] = "";
 p["None Value"] = "";
 p["Parametric Search Falg"] = "Y";
 p["Read Only Flag"] = "N";
 p["Sequence Number"] = "-1";
 p["Inherited Flag"] = "Y";
 p["Overwritten Flag"] = "N";
 p["Required Flag"] = "N";
 p["Validation Spec"] = "";
 p["Display Name"] = "Edition";
 p["Maximum Value"] = "";
 p["Domain Type"] = "Text";
 p["Hidden Flag"] = "N";
 p["Enumerated Flag"] = "Y";
 p["Orig Id"] = "42@777ar";
 p["Pre Pick CD"] = "";
 p["Description"] = "";
 p["Asset Flag"] = "N";
 c[0] PropertySet@0AB284C8 p#0 c#3 type="ListOfAttribute LOV Value" vt=0
value=""
 {
 c[0] PropertySet@0AB271B0 p#1 c#0 type="Attribute LOV Value" vt=0
value=""
 {
 p["Value"] = "Enterprise";
 }
 c[1] PropertySet@0AB2AB70 p#1 c#0 type="Attribute LOV Value" vt=0
value=""
 {
 p["Value"] = "Home";
 }
 c[2] PropertySet@0AB2A7D8 p#1 c#0 type="Attribute LOV Value" vt=0
value=""
 {
 p["Value"] = "Professional";
 }
 }
 }
 }
 }
}
```

Error Conditions

None.

GetProductChildren Method

Gets a list of child business component details for a given product, the child business component name, and the required fields to return.

Syntax

Product Data Service.InvokeMethod(“GetProductChildren", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: ContextMode String: EligibilityMode String: PricingMode Hierarchy: SiebelMessage (Integration object PDS Catalog Context) Hierarchy: SiebelMessage (Integration object PDS Product Children Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Children Interface)

Example of Input Arguments

```
PropertySet@0012C6BC p#3 c#2 type="" vt=0 value=""
{
 p["EligibilityMode"] = "";
 p["ContextMode"] = "SiebelContext";
 p["PricingMode"] = "Y";
 c[0] PropertySet@0B89E3F0 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Context";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A959098 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0AB4A228 p#0 c#1 type="Context" vt=0 value=""
 {
 c[0] PropertySet@0A5D78B8 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0B6396E0 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0BAFF048 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Product Children Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@08BDC3C8 p#0 c#1 type="ListOfPDS Product Children Interface"
vt=0 value=""
 {
 c[0] PropertySet@0A73E4B0 p#1 c#0 type="Product" vt=0 value=""
 {
```

```
 p["Id"] = "99-28W7U";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0B96EAA8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1JLN";
 p["IntObjectName"] = "PDS Product Children Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0B9DFB90 p#0 c#1 type="ListOfPDS Product Children Interface"
vt=0 value=""
 {
 c[0] PropertySet@0BBDEA68 p#23 c#2 type="Product" vt=3 value="99-28W7U"
 {
 p["Class Id"] = "99-28VEN";
 p["Vendor Integration Id"] = "";
 p["Reference Price"] = "";
 p["Product Def Type Code"] = "Customizable";
 p["Eligibility Reason"] = "";
 p["Id"] = "99-28W7U";
 p["Product Id"] = "99-28W7U";
 p["Pre Pick"] = "N";
 p["Part #"] = "GPAP";
 p["Eligibility Status"] = "";
 p["Integration Id"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Parent Item Id"] = "";
 p["Net Price"] = "0";
 p["List Price"] = "0";
 p["Effective To"] = "";
 p["Product Line Id"] = "No Match Row Id";
 p["Product Configuration Model Id"] = "99-28W7U";
 p["Effective From"] = "";
 p["List Price Type Code"] = "STANDARD";
 p["Name"] = "CDA_gp_Audio Package";
 p["Quantity"] = "1";
 p["Price List Id"] = "1-31C5";
 c[0] PropertySet@0BABA878 p#0 c#2 type="ListOfKey Feature" vt=0 value=""
 {
 c[0] PropertySet@0BAB17E8 p#4 c#0 type="Key Feature" vt=0 value=""
 {
 p["Vendor Location"] = "";
 p["Vendor"] = "";
 p["Product Id"] = "99-28W7U";
 p["Feature"] = "24 x 7 Phone Support";
 }
 c[1] PropertySet@0BD94500 p#4 c#0 type="Key Feature" vt=0 value=""
 {
 p["Vendor Location"] = "";
 p["Vendor"] = "";
 p["Product Id"] = "99-28W7U";
 p["Feature"] = "Access Time";
 }
 }
 c[1] PropertySet@0BC8DD28 p#0 c#1 type="ListOfLiterature" vt=0 value=""
 {
 c[0] PropertySet@0B9A0688 p#10 c#0 type="Literature" vt=0 value=""
 {
```

```
 p["Internal"] = "Y";
 p["Sales Tool Type"] = "Sales Tool: Brochure";
 p["File Type"] = "pdf";
 p["Auto Update Flag"] = "Y";
 p["Integration Id"] = "";
 p["Size"] = "161754";
 p["LitFileDate"] = "03/02/2001 17:46:10";
 p["Name"] = " Brochure: Ernst & Young's The Centric Group";
 p["File Name"] = "ds_EndUser_S_LIT_12-GSYZV_12-1INJ";
 p["Description"] = "";
 }
 }
 }
 }
 }
}
```

Error Conditions

None.

GetSearchOptions Method

Displays product classes, such as Computer, Printer, and so on. It can also use All to refer to all products. External.

Syntax

Product Data Service.InvokeMethod(“GetSearchOptions", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy: SiebelMessage (Integration object PDS Searchable Product Class Interface)
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Searchable Product Class Interface)

Example of Input Arguments

```
PropertySet@0012CCC4 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0A69DB90 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Searchable Product Class Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A910EB8 p#0 c#1 type="ListOfPDS Searchable Product Class
Interface" vt=0 value=""
 {
 c[0] PropertySet@0A9F5FF8 p#1 c#0 type="ISS PS Direct Subclass" vt=0 value=""
 {
 p["Name"] = "OM*";
```

```
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AC49B70 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1NGI";
 p["IntObjectName"] = "PDS Searchable Product Class Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A9F3FC8 p#0 c#1 type="ListOfPDS Searchable Product Class
Interface" vt=0 value=""
 {
 c[0] PropertySet@0AC31BD8 p#3 c#0 type="ISS PS Direct Subclass" vt=0 value=""
 {
 p["Class Id"] = "99-27K1T";
 p["Display Name"] = "OM_Operating System";
 p["Name"] = "OM_Operating System";
 }
 }
 }
}
```

Error Conditions

None.

GetSearchParameters Method

Gets the list of attributes available for the class(es) using the input search specification in the SiebelMessage. External

This method only retrieves product attributes that are flagged as searchable. If the example is followed, then data is generated only if the attributes for the product are flagged as searchable in the database.

Syntax

Product Data Service.InvokeMethod(“GetSearchParameters", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: FieldNameClassId String: LookupFromCache (To be Obsolete) String: MaximumAttributes Hierarchy: SiebelMessage (Integration object PDS Searchable Product Class Interface)

Argument	Type	Description
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Class Attributes Interface)

Example of Input Arguments

```
PropertySet@0012CCC4 p#3 c#1 type="" vt=0 value=""
{
 p["LookupFromCache"] = "";
 p["MaximumAttributes"] = "";
 p["FieldNameClassId"] = "";
 c[0] PropertySet@0AAEBFC8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Searchable Product Class Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AAD45B0 p#0 c#1 type="ListOfPDS Searchable Product Class
Interface" vt=0 value=""
 {
 c[0] PropertySet@02C42F08 p#1 c#0 type="ISS PS Direct Subclass" vt=0 value=""
 {
 p["Class Id"] = "99-27K1T";
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0B918DE8 p#5 c#1 type="SiebelMessageOut" vt=0 value=""
 {
 p["MessageId"] = "42-1NJB";
 p["ID_ALL"] = "All";
 p["IntObjectName"] = "PDS Product Class Attributes Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@055458C0 p#0 c#1 type="ListOfPDS Product Class Attributes
Interface" vt=0 value=""
 {
 c[0] PropertySet@0B94F868 p#3 c#1 type="ISS PS Direct Subclass" vt=0 value=""
 {
 p["Class Id"] = "99-27K1T";
 p["Display Name"] = "OM_Operating System";
 p["Name"] = "OM_Operating System";
 c[0] PropertySet@0A0B55E0 p#0 c#9 type="ListOfISS Class Attribute" vt=0
value=""
 {
 c[0] PropertySet@0542CC18 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Edition";
 p["Attribute Id"] = "OM_Edition";
 c[0] PropertySet@0A6D7BB0 p#0 c#4 type="ListOfAttribute LOV
Value" vt=0 value=""
 {
 c[0] PropertySet@0543F558 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
```

```
 p["Value"] = "Professional";
 }
 c[1] PropertySet@0538FA40 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "Home";
 }
 c[2] PropertySet@0A7F7FA8 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "Enterprise";
 }
 c[3] PropertySet@0A8A2B08 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ALL";
 }
 }
 }
 c[1] PropertySet@0A9324D8 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Language";
 p["Attribute Id"] = "OM_Language";
 c[0] PropertySet@02C43A00 p#0 c#6 type="ListOfAttribute LOV
Value" vt=0 value=""
 {
 c[0] PropertySet@0A9F3D30 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ESN";
 }
 c[1] PropertySet@053D6FD8 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ENU";
 }
 c[2] PropertySet@0A893AE0 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "FRA";
 }
 c[3] PropertySet@0BCF40E8 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "DEU";
 }
 c[4] PropertySet@0536BA78 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ALL";
 }
 c[5] PropertySet@02B9F1B8 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "JPN";
 }
 }
 }
 c[2] PropertySet@0A982878 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Service Pack";
 p["Attribute Id"] = "OM_Service Pack";
 c[0] PropertySet@0AB4E3E0 p#0 c#4 type="ListOfAttribute LOV
Value" vt=0 value=""
```

```
 {
 c[0] PropertySet@0AC70068 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "3";
 }
 c[1] PropertySet@093FB308 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "2";
 }
 c[2] PropertySet@0AB032E8 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "1";
 }
 c[3] PropertySet@09642C40 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ALL";
 }
 }
 }
 c[3] PropertySet@09617088 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Version";
 p["Attribute Id"] = "OM_Version";
 c[0] PropertySet@09403568 p#0 c#4 type="ListOfAttribute LOV
Value" vt=0 value=""
 {
 c[0] PropertySet@096264A0 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "Upgrade";
 }
 c[1] PropertySet@093FC308 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "OEM";
 }
 c[2] PropertySet@09611680 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "Full";
 }
 c[3] PropertySet@09641B40 p#1 c#0 type="Attribute LOV Value"
vt=0 value=""
 {
 p["Value"] = "ALL";
 }
 }
 }
 c[4] PropertySet@0964C768 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Build";
 p["Attribute Id"] = "OM_Build";
 c[0] PropertySet@0AAB1C50 p#0 c#0 type="ListOfAttribute LOV
Value" vt=0 value=""
 }
 c[5] PropertySet@09616D00 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Default Browser";
 p["Attribute Id"] = "OM_Default Browser";
```

```
 c[0] PropertySet@09400380 p#0 c#0 type="ListOfAttribute LOV
Value" vt=0 value=""
 }
 c[6] PropertySet@09614D30 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Publish";
 p["Attribute Id"] = "OM_Publish";
 c[0] PropertySet@0940C548 p#0 c#0 type="ListOfAttribute LOV
Value" vt=0 value=""
 }
 c[7] PropertySet@094060B0 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Brand";
 p["Attribute Id"] = "OM_Brand";
 c[0] PropertySet@0BC91558 p#0 c#0 type="ListOfAttribute LOV
Value" vt=0 value=""
 }
 c[8] PropertySet@09642538 p#2 c#1 type="ISS Class Attribute" vt=0
value=""
 {
 p["Display Name"] = "Extended Warranty";
 p["Attribute Id"] = "OM_Extended Warranty";
 c[0] PropertySet@09611A00 p#0 c#0 type="ListOfAttribute LOV
Value" vt=0 value=""
 }
 }
 }
 }
 }
}
```

Error Conditions

Error Message	Description
The 'Data Sync Service: GetSearchParameters' user property is not defined in business service 'PDS Product Data Service	The specified user property can not be found on the business service, which must have value PDS Product Attributes by default.
Unable to create the Business Service 'PDS Product Attributes	Can not initiate the business service.

ExecuteSearch Method

Executes the search on products using the input search option, such as product class, class attributes, attribute constraints, pricing criteria, and so on. Based on the input parameters, the ExecuteSearch method supports pagination, returning the search output in pages. This is an external method.

Syntax

Product Data Service.InvokeMethod(“ExecuteSearch", Input Arguments,Output Arguments);

Arguments

There are two types of search supported: SearchByProduct or SearchByClass. SearchOption is a required input to indicate the type of search performed.

Arguments for SearchByProduct	Type	Description
Input Arguments	In	SearchOption: SearchByProduct. This input argument is required. IO: an instance of Integration Object PDS Product Interface. This input argument is required. PricingMode: Y/N (calculates price or not). Optional. Default value: Y. EligibilityMode: 0, 1, 2 (eligibility display mode. 0 do not run; 1 run; 2 run but only return eligible ones). Optional. Default value: 1. ContextMode: SiebelContext or ExternalConext. Optional, but if it's not provided,it won't run pricing and eligibility. Context: an instance of Integration Object PDS Catalog Context. It either contains SiebelContext or ExternalContext. Optional only if ContextMode is not provided; otherwise, required. PriceSearchExpression: price field search expression. For example: is greater than 100 and is less than 1000. Optional. ProductSearchSpec: product business component search specification. For example: [Name] like "IBM*". Optional. ResultSize: number of products returned. Optional. Default Value 30.
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Interface)

Arguments for SearchByClass	Type	Description
Input Arguments	In	SearchOption: SearchByClass. This input argument is required. IO: an instance of Integration Object PDS Product Class Attributes Interface. This input argument is required. The user must specify class ID, attribute ID, and an attribute value in attribute LOV value structure (that is, an array with 1 element for either freeform or LOV type attribute). PricingMode: Y/N (calculates price or not). Optional. Default value: Y. EligibilityMode: 0, 1, 2 (eligibility display mode. 0 do not run; 1 run; 2 run but only return eligible products). Optional. Default value is 1.

Arguments for SearchByClass	Type	Description
		ContextMode: SiebelContext or ExternalContext. Optional, but if it is not provided, it won't run pricing and eligibility. Context: an instance of Integration Object PDS Catalog Context. It either contains SiebelContext or ExternalContext. Optional only if ContextMode is not provided; otherwise, required. PriceSearchExpression: price field search expression. For example: is greater than 100 and is less than 1000. Optional. ProductSearchSpec: product business component search specification. For example: [Name] like "IBM*". Optional. ResultSize: number of products returned. Optional. Default Value 30.
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Product Interface)

Properties of Input Arguments

Property	Type	Value	Description
ContextMode	String	SiebelContext, ExternalContext	SiebelContext: Using existing Quote/Order Information in Siebel database. In this case, the API constructs the context using the provided row_id. ExternalContext: External Call provides the context information, API passes them directly to the Pricing and Eligibility Process.
ResultSize	Number	For example, 30	The number of products that would be retrieved from the query result.
Context	Hierarchy: SiebelMessage (Integration object PDS Catalog Context)	See the following examples.	None
Integration Object	SiebelMessage	PDS Product Interface	None
StartRowNum	Number	For example, 0.	Determines the starting point for record retrieval. This is the zero-based index of the record within the record set.
PageSize	Number	For example, 5.	The number of records to be returned.

Property	Type	Value	Description
RecordCountNeeded	String	True	When this is set to True, then this parameter indicates that a record count is needed as part of the output for this search. Valid values are True or False.

Properties of Output Arguments

Property	Type	Value	Description
RecordCount	String	For example, 30.	If the RecordCountNeeded input parameter is set to True, then this output parameter provides the number of elements returned by the search specification.
LastPage	String	True or False	Indicates if the returned record is the last record in the record set.

Example of Input Arguments

```
PropertySet@0012C82C p#7 c#3 type="" vt=0 value=""
{
 p["EligibilityMode"] = "1";
 p["ProductSearchSpec"] = "";
 p["PriceSearchExpression"] = "";
 p["ContextMode"] = "SiebelContext";
 p["ResultSize"] = "";
 p["PricingMode"] = "Y";
 p["SearchOption"] = "SearchByClass";
 c[0] PropertySet@0A7AEFC0 p#4 c#1 type="Context" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Catalog Context";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A778688 p#0 c#1 type="ListOfPDS Catalog Context" vt=0 value=""
 {
 c[0] PropertySet@0A7F7CE0 p#0 c#1 type="Context" vt=0 value=""
 {
 c[0] PropertySet@0AAD12D8 p#0 c#1 type="ListOfSiebel Context" vt=0 value=""
 {
 c[0] PropertySet@0AA29A88 p#4 c#0 type="Siebel Context" vt=0 value=""
 {
 p["BusObjName"] = "Quote";
 p["ContextRowId"] = "1-10MLH";
 p["SignalMode"] = "Quote";
 p["BusCompName"] = "Quote";
 }
 }
 }
 }
 }
 c[1] PropertySet@0AA524A0 p#4 c#1 type="ClassAttributes" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Product Class Attributes Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
```

```
 c[0] PropertySet@0AACF400 p#0 c#1 type="ListOfPDS Product Class Attributes
Interface" vt=0 value=""
 {
 c[0] PropertySet@0AA92A18 p#1 c#1 type="ISS PS Direct Subclass" vt=0 value=""
 {
 p["Class Id"] = "99-27K1T";
 c[0] PropertySet@0AA53858 p#0 c#1 type="ListOfISS Class Attribute" vt=0
value=""
 {
 c[0] PropertySet@0A9B38B8 p#2 c#0 type="ISS Class Attribute" vt=0
value=""
 {
 p["Value"] = "Professional";
 p["Attribute Id"] = "OM_Edition";
 }
 }
 }
 }
 }
 c[2] PropertySet@054B87F0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["IntObjectName"] = "PDS Product Interface";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@00C8E8F8 p#0 c#1 type="ListOfPDS Product Interface" vt=0 value=""
 {
 c[0] PropertySet@0A798FF8 p#0 c#0 type="Product" vt=0 value=""
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012CCB8 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@02C806E0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Product Interface";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@02B9FF00 p#0 c#2 type="ListOfPDS Product Interface" vt=0 value=""
 {
 c[0] PropertySet@0B97D3F8 p#24 c#0 type="Product" vt=3 value="99-27KMW"
 {
 p["Class Id"] = "99-27K1T";
 p["Product Def Type Code"] = "Customizable";
 p["Reference Price"] = "";
 p["Vendor Integration Id"] = "";
 p["Id"] = "99-27KMW";
 p["Eligibility Reason"] = "";
 p["Pre Pick"] = "N";
 p["Product Id"] = "99-27KMW";
 p["Part #"] = "";
 p["Eligibility Status"] = "";
 p["Parent Item Id"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Integration Id"] = "";
 p["Net Price"] = "";
 p["Effective To"] = "";
 p["List Price"] = "";
 p["Product Configuration Model Id"] = "99-27KMW";
 p["Product Line Id"] = "No Match Row Id";
 p["Effective From"] = "";
```

```
p["List Price Type Code"] = "STANDARD";
p["Product Type Code"] = "Product";
p["Quantity"] = "1";
p["Name"] = "OM_Windows NT";
p["Price List Id"] = "99-27LE5";
}
c[1] PropertySet@0B824EB8 p#24 c#0 type="Product" vt=3 value="99-27KN1"
{
p["Class Id"] = "99-27K1T";
p["Product Def Type Code"] = "None";
p["Reference Price"] = "";
p["Vendor Integration Id"] = "";
p["Id"] = "99-27KN1";
p["Eligibility Reason"] = "";
p["Pre Pick"] = "N";
p["Product Id"] = "99-27KN1";
p["Part #"] = "";
p["Eligibility Status"] = "";
p["Parent Item Id"] = "";
p["Inclusive Eligibility Flag"] = "N";
p["Integration Id"] = "";
p["Net Price"] = "";
p["Effective To"] = "";
p["List Price"] = "";
p["Product Configuration Model Id"] = "99-27KN1";
p["Product Line Id"] = "No Match Row Id";
p["Effective From"] = "";
p["List Price Type Code"] = "STANDARD";
p["Product Type Code"] = "Product";
p["Quantity"] = "1";
p["Name"] = "OM_Windows XP";
p["Price List Id"] = "99-27LE5";
}
}
}
}
```

Error Conditions

Error Message	Description
The 'Data Sync Service: ExecuteSearch' user property is not defined in business service 'PDS Product Data Service	The specified user property can not be found on the business service, which must have value PDS Product by default.
Cannot get property set 1%.	The required property set input 1% is not provided.
Unable to locate record with row id as '%1' in '%2	The specified row_id 1% in BC 2% could not be found.
Unable to create the Business Service PDS Product	Can not initiate the business service.

AddFavoriteToQuote, AddFavoriteToOrder, AddFavoriteItemtoQuote, AddFavoriteItemtoOrder, SaveQuoteAsFavorite, and SaveOrderAsFavorite Methods

Add a favorite to the active quote. External.

Note: AddFavoriteToOrder, AddFavoriteItemtoQuote, AddFavoriteItemtoOrder, SaveQuoteAsFavorite, and SaveOrderAsFavorite have similar syntax and logic to AddFavoriteToQuote.

Syntax

Product Data Service.InvokeMethod(“AddFavoriteToQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Favorite Name String: Quote Number
Output Arguments	Out	Hierarchy: SiebelMessage (Integration object PDS Quote Interface)

Error Conditions

None.

DeleteQuote Method

Deletes a quote that matches the input.

Syntax

Product Data Service.InvokeMethod(“DeleteQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)

Argument	Type	Description
Output Arguments	Out	Hierarchy :SiebelMessage

Example of Input Arguments

```
PropertySet@0012D384 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0BAEB900 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "2-3ZGU";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Quote";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0BACBEF0 p#0 c#1 type="ListOfPDS Quote" vt=0 value=""
 {
 c[0] PropertySet@0BE2F288 p#23 c#1 type="Header" vt=0 value=""
 {
 p["Account Location"] = "";
 p["Opportunity Id"] = "";
 p["Id"] = "2-47Q9P";
 p["Revision"] = "1";
 p["Account Id"] = "";
 p["Status"] = "In Progress";
 p["Conflict Id"] = "0";
 p["Account"] = "";
 p["Contact Id"] = "";
 p["Currency Code"] = "USD";
 p["Discount"] = "";
 p["Updated"] = "02/11/2005 12:34:46";
 p["Primary Organization Id"] = "0-R9NH";
 p["Integration Id"] = "";
 p["End Date"] = "03/13/2005 00:00:00";
 p["Created"] = "02/11/2005 12:34:46";
 p["Active"] = "Y";
 p["Due Date"] = "02/12/2005 00:00:00";
 p["Quote Type"] = "Quotation";
 p["Name"] = "PDS Quote Test";
 p["Start Date"] = "02/11/2005 00:00:00";
 p["Quote Number"] = "2-47Q9P";
 p["Price List Id"] = "";
 c[0] PropertySet@0BDDC2E8 p#0 c#1 type="ListOfLine Item" vt=0 value=""
 {
 c[0] PropertySet@0BBD5478 p#63 c#1 type="Line Item" vt=0 value=""
 {
 p["NRC CxTotal"] = "0";
 p["Parent Id"] = "";
 p["Cfg State Code"] = "New Item";
 p["Unit of Measure"] = "";
 p["Volume Discount Item Id"] = "";
 p["Id"] = "2-47Q9S";
 p["Cfg Type"] = "";
 p["Eligibility Reason"] = "";
 p["Exclude Pricing Flag"] = "N";
 p["Extended Quantity Requested"] = "1";
 p["Asset Integration Id"] = "2-47Q9S";
 p["Root Id"] = "2-47Q9S";
 p["Conflict Id"] = "0";
 p["Need Refresh"] = "Y";
 p["Prod Prom Name"] = "";
 p["Volume Discount Item"] = "";
 p["Line Number"] = "1";
```

```
 p["Product Id"] = "2-47Q6J";
 p["Currency Code"] = "USD";
 p["Smart Part Number"] = "";
 p["Discount"] = "";
 p["Prod Item Id"] = "";
 p["Discount Percent"] = "";
 p["Original List Price"] = "";
 p["Updated"] = "02/11/2005 12:37:02";
 p["Current Price"] = "";
 p["Eligibility Status"] = "";
 p["Port Item Id"] = "";
 p["Prod Prom Id"] = "";
 p["Has Generics Flag"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Post Pick CD"] = "";
 p["Rollup Amount"] = "0";
 p["Integration Id"] = "2-47Q9S";
 p["Prod Prom Instance Id"] = "";
 p["Forcastable Flag"] = "Y";
 p["Prod Prom Rule Id"] = "";
 p["Discount Source"] = "Manual";
 p["Effective To"] = "";
 p["List Price"] = "";
 p["Pricing Comments"] = "";
 p["Volume Upsell Item"] = "";
 p["Product Line Id"] = "No Match Row Id";
 p["Rollup Item Price"] = "0";
 p["Unit Price"] = "";
 p["Price Type"] = "One-Time";
 p["Volume Discount Id"] = "";
 p["Created"] = "02/11/2005 12:34:52";
 p["Effective From"] = "";
 p["Prod Prom Source Id"] = "";
 p["Discount Amount"] = "";
 p["Pre Pick CD"] = "";
 p["Exchange Date"] = "03/13/2005 00:00:00";
 p["Item Price List Id"] = "";
 p["Volume Upsell Message"] = "";
 p["Configuration Model Id"] = "2-47Q6J";
 p["Name"] = "PDS Product";
 p["Quantity"] = "1";
 p["Action Code"] = "Add";
 p["Description"] = "";
 p["Price List Id"] = "";
 p["MRC CxTotal"] = "0";
 p["Volume Upsell Item Id"] = "";
 c[0] PropertySet@070A6B30 p#0 c#0 type="ListOfXA" vt=0 value=""
 }
 }
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012DA78 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@086490D0 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "2-3ZGU";
 p["IntObjectName"] = "PDS Quote";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0BAB2950 p#0 c#1 type="ListOfPDS Quote" vt=0 value=""
```

```
{
c[0] PropertySet@0BF41DF8 p#0 c#0 type="Header" vt=0 value=""
}
}
}
```

Error Conditions

None

InsertQuote Method

Insert a quote that is defined in input. If that quote already exists, then the method updates the existing quote.

Syntax

Product Data Service.InvokeMethod(“InsertQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

UpdateQuote Method

Update Quote a quote that matches the input by user key.

Syntax

Product Data Service.InvokeMethod(“UpdateQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)

Argument	Type	Description
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

DeleteQuoteItem Method

Deletes the quote line item that is defined in the input.

Syntax

Product Data Service.InvokeMethod("DeleteQuoteItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

InsertQuoteItem Method

Inserts a quote line item that is defined in the input. If the quote line item already exists, then the method updates the existing quote line item.

Syntax

Product Data Service.InvokeMethod("InsertQuoteItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

UpdateQuoteItem Method

Updates an existing quote line item that is matched by user key and updated with value in the input.

Syntax

Product Data Service.InvokeMethod(“UpdateQuoteItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Quote)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

DeleteOrder Method

Deletes the order that matches the input.

Syntax

Product Data Service.InvokeMethod(“DeleteOrder", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)
Output Arguments	Out	Hierarchy :SiebelMessage

Example of Input Arguments

```
PropertySet@0012D384 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0A6E4FE8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1D9U";
 p["MessageType"] = "Integration Object";
 p["IntObjectName"] = "PDS Order";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0ACADCC0 p#0 c#1 type="ListOfPDS Order" vt=0 value=""
 {
 c[0] PropertySet@0A903FE8 p#17 c#1 type="Header" vt=0 value=""
 {
 p["Account Location"] = "";
 p["Id"] = "42-50SHP";
 p["Revision"] = "1";
 p["Account Id"] = "";
 p["Status"] = "Pending";
 p["Conflict Id"] = "0";
 p["Order Type"] = "Service Order";
 p["Account"] = "";
 p["Contact Id"] = "";
 p["Currency Code"] = "USD";
 p["Discount"] = "";
 p["Updated"] = "02/11/2005 13:54:39";
 p["Integration Id"] = "";
 p["Opty Id"] = "";
 p["Order Number"] = "PDS Order Item Test";
 p["Created"] = "02/11/2005 13:54:39";
 p["Price List Id"] = "";
 c[0] PropertySet@0A900550 p#0 c#1 type="ListOfLine Item" vt=0 value=""
 {
 c[0] PropertySet@0AB61530 p#62 c#1 type="Line Item" vt=0 value=""
 {
 p["NRC CxTotal"] = "0";
 p["Parent Id"] = "";
 p["Cfg State Code"] = "New Item";
 p["Volume Discount Item Id"] = "";
 p["Id"] = "42-50SHW";
 p["Cfg Type"] = "";
 p["Eligibility Reason"] = "";
 p["Exclude Pricing Flag"] = "N";
 p["Extended Quantity Requested"] = "1";
 p["Asset Integration Id"] = "42-50SHW";
 p["Root Id"] = "42-50SHW";
 p["Conflict Id"] = "0";
 p["Need Refresh"] = "Y";
 p["Prod Prom Name"] = "";
 p["Volume Discount Item"] = "";
 p["Line Number"] = "1";
 p["Product Id"] = "42-50S1M";
```

```
 p["Currency Code"] = "USD";
 p["Smart Part Number"] = "";
 p["Discount"] = "";
 p["Prod Item Id"] = "";
 p["Discount Percent"] = "";
 p["Original List Price"] = "";
 p["Updated"] = "02/11/2005 13:56:09";
 p["Current Price"] = "";
 p["Eligibility Status"] = "";
 p["Port Item Id"] = "";
 p["Prod Prom Id"] = "";
 p["Has Generics Flag"] = "";
 p["Inclusive Eligibility Flag"] = "N";
 p["Integration Id"] = "42-50SHW";
 p["Post Pick CD"] = "";
 p["Prod Prom Instance Id"] = "";
 p["Rollup Amount"] = "0";
 p["Forcastable Flag"] = "";
 p["Prod Prom Rule Id"] = "";
 p["Discount Source"] = "Manual";
 p["List Price"] = "";
 p["Pricing Comments"] = "";
 p["Volume Upsell Item"] = "";
 p["Effective To"] = "";
 p["Unit Price"] = "";
 p["Price Type"] = "One-Time";
 p["Product Line Id"] = "No Match Row Id";
 p["Rollup Item Price"] = "0";
 p["Volume Discount Id"] = "";
 p["Created"] = "02/11/2005 13:54:50";
 p["Due Date"] = "02/12/2005 00:00:00";
 p["Effective From"] = "";
 p["Prod Prom Source Id"] = "";
 p["Discount Amount"] = "";
 p["Exchange Date"] = "";
 p["Item Price List Id"] = "";
 p["Pre Pick CD"] = "";
 p["Volume Upsell Message"] = "";
 p["Configuration Model Id"] = "42-50S1M";
 p["Name"] = "PDS Product";
 p["Quantity"] = "1";
 p["Action Code"] = "Add";
 p["Price List Id"] = "";
 p["MRC CxTotal"] = "0";
 p["Volume Upsell Item Id"] = "";
 c[0] PropertySet@0A92EE38 p#0 c#0 type="ListOfXA" vt=0 value=""
 }
 }
 }
 }
 }
}
```

Example of Output Arguments

```
PropertySet@0012DA78 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AB9F028 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1D9U";
 p["IntObjectName"] = "PDS Order";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AC32B10 p#0 c#1 type="ListOfPDS Order" vt=0 value=""
 {
```

```
    c[0] PropertySet@0A63EB20 p#0 c#0 type="Header" vt=0 value=""
    }
    }
  }
```

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

InsertOrder Method

Insert an order that is defined in the input. If the order already exists, then the method updates the existing order.

Syntax

Product Data Service.InvokeMethod("InsertOrder", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

UpdateOrder Method

Updates an order that is matched with the input by user key.

Syntax

Product Data Service.InvokeMethod("UpdateOrder", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)

Argument	Type	Description
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

DeleteOrderItem Method

Delete the order line item that is defined in the input.

Syntax

Product Data Service.InvokeMethod(“DeleteOrderItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

InsertOrderItem Method

Inserts an order line item that is defined in the input. If the order line item already exists, then the method updates the existing order line item.

Syntax

Product Data Service.InvokeMethod(“InsertOrderItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

UpdateOrderItem Method

Updates the existing order item that is matched by user key and updated with the value in the input.

Syntax

Product Data Service.InvokeMethod(“UpdateOrderItem", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	Hierarchy :SiebelMessage(Integration object PDS Order)
Output Arguments	Out	Hierarchy :SiebelMessage

Error Conditions

None.

This method calls the help method described in *DeleteQuote Method*.

GetActiveQuote Method

Finds the current active quote with input account Id, compound Product Number and Price List Id and returns this Quote. If it cannot find the quote, then the method creates and returns a new quote.

Syntax

Product Data Service.InvokeMethod(“GetActiveQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Account Id (optional) String: Compound Product Number (Optional) String: Price List Id (Optional)
Output Arguments	Out	Hierarchy :SiebelMessage(Integration object PDS Quote)

Example of Input Arguments

```
PropertySet@0012D370 p#4 c#0 type="" vt=0 value=""
{
 p["Account Id"] = "";
 p["ProcessName"] = "SIS OM Active Quote Sub-process";
 p["Compound Product Number"] = "1-R234";
 p["Price List Id"] = "1-D0D8";
}
```

Example of output Arguments

```
PropertySet@0012DA78 p#0 c#1 type="" vt=0 value=""
{
 c[0] PropertySet@0AD2BB48 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "2-3ZJM";
 p["IntObjectName"] = "PDS Quote";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0AD85DA0 p#0 c#1 type="ListOfPDS Quote" vt=0 value=""
 {
 c[0] PropertySet@0AAFC110 p#23 c#1 type="Header" vt=0 value=""
 {
 p["Account Location"] = "";
 p["Opportunity Id"] = "";
 p["Revision"] = "1";
 p["Id"] = "2-47QI3";
 p["Status"] = "In Progress";
 p["Account Id"] = "";
 p["Conflict Id"] = "0";
 p["Currency Code"] = "USD";
 p["Contact Id"] = "";
 p["Account"] = "";
 p["Discount"] = "";
 p["Primary Organization Id"] = "0-R9NH";
 p["Updated"] = "02/11/2005 13:16:32";
 p["Integration Id"] = "";
 p["End Date"] = "03/13/2005 00:00:00";
 p["Due Date"] = "02/12/2005 00:00:00";
 p["Active"] = "Y";
 p["Created"] = "02/11/2005 13:15:11";
 p["Quote Type"] = "Quotation";
 p["Quote Number"] = "2-47QI3";
 p["Start Date"] = "02/11/2005 00:00:00";
 p["Name"] = "2-47QI3";
 p["Price List Id"] = "1-D0D8";
```

```
    c[0] PropertySet@0A98D8E8 p#0 c#0 type="ListOfLine Item" vt=0 value=""
    }
    }
    }
  }
```

Error Conditions

None.

Help Method Syntax

ErrCode ActiveDOCOperator(Input Arguments,Output Arguments, const SSstring strMode)

Help Method Arguments

Argument	Type	Description
Input Arguments	In	String: Account Id (optional) String: Compound Product Number (Optional) String: Price List Id (Optional)
Output Arguments	Out	Hierarchy :SiebelMessage(Integration object PDS Quote)
const SSstring& strMode	In	String: Name help to get corresponding workflow, available value could be GetActiveQuote, GetActiveOrder, SetActiveQuote, SetActiveOrder,

Help Method Error Conditions

IDS_ERR_ANIESALES_MISSING_SERVICE_NAME: The '%1' user property is not defined in business service '%2'

SSAOMErrCannotCreateService: Unable to create the Business Service '%1'

SetActiveQuote Method

Finds the current active quote with input account Id, compound Product Number and Price List Id. If it can find a match, then it returns this quote. If it cannot find a match, then it creates a new quote, sets it as active quote, and returns this quote.

Syntax

Product Data Service.InvokeMethod(“SetActiveQuote", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Account Id (optional) String: Compound Product Number (Optional) String: Price List Id (Optional)
Output Arguments	Out	Hierarchy :SiebelMessage(Integration object PDS Quote)

This method calls the help method described in *GetActiveQuote Method*.

GetActiveOrder Method

Finds the current active order with input account Id, compound Product Number and Price List Id. If it can find a match, then it returns this order. If cannot find a match, then it creates new order, sets it as the active order, and returns this order.

Syntax

Product Data Service.InvokeMethod(“GetActiveOrder", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Account Id (optional) String: Compound Product Number (Optional) String: Price List Id (Optional)
Output Arguments	Out	Hierarchy :SiebelMessage(Integration object PDS Quote)

Example of Input Arguments

```
PropertySet@0012D370 p#4 c#0 type="" vt=0 value=""
{
 p["Account Id"] = "";
 p["ProcessName"] = "SIS OM Active Order Sub-Process";
 p["Compound Product Number"] = "";
 p["Price List Id"] = "";
}
```

Example of Output Arguments

```
PropertySet@0012DA78 p#0 c#1 type="" vt=0 value=""
```

```
{
 c[0] PropertySet@0A98A6A8 p#4 c#1 type="SiebelMessage" vt=0 value=""
 {
 p["MessageId"] = "42-1DFE";
 p["IntObjectName"] = "PDS Order";
 p["MessageType"] = "Integration Object";
 p["IntObjectFormat"] = "Siebel Hierarchical";
 c[0] PropertySet@0A9CD9C0 p#0 c#1 type="ListOfPDS Order" vt=0 value=""
 {
 c[0] PropertySet@0A97E488 p#17 c#1 type="Header" vt=0 value=""
 {
 p["Account Location"] = "";
 p["Revision"] = "1";
 p["Id"] = "42-50SNB";
 p["Status"] = "Pending";
 p["Account Id"] = "";
 p["Order Type"] = "Sales Order";
 p["Conflict Id"] = "0";
 p["Currency Code"] = "USD";
 p["Contact Id"] = "";
 p["Account"] = "";
 p["Discount"] = "";
 p["Updated"] = "02/11/2005 14:06:32";
 p["Integration Id"] = "";
 p["Order Number"] = "146-8435207";
 p["Opty Id"] = "";
 p["Created"] = "02/11/2005 14:06:32";
 p["Price List Id"] = "";
 c[0] PropertySet@0A9F7E98 p#0 c#0 type="ListOfLine Item" vt=0 value=""
 }
 }
 }
}
```

Error Conditions

None.

This method calls the help method described in *GetActiveQuote Method*.

SetActiveOrder Method

Finds current active order with input account Id, compound Product Number and Price List Id, and returns this order. If it cannot find a matching order, then it creates and returns a new one.

Syntax

Product Data Service.InvokeMethod(“SetActiveOrder", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
Input Arguments	In	String: Account Id (optional) String: Compound Product Number (Optional) String: Price List Id (Optional)

Argument	Type	Description
Output Arguments	Out	Hierarchy :SiebelMessage(Integration object PDS Order)

Error Conditions

None.

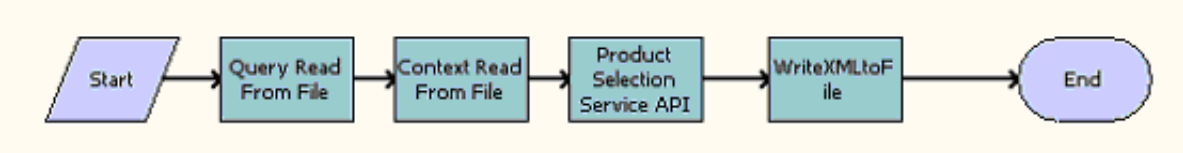
This method calls the help method described in *GetActiveQuote Method*.

PDS PublishCatalog Test Sample Workflow

This sample workflow illustrates how to create your own workflows that use the Product Data Service API. This workflow is not meant to be used directly.

The inputs and outputs of the Product Data Service API methods are all in the format of the Siebel property set, which can be converted to and from xml files using Siebel EAI. The hierarchy in the output property set is controlled by the Integration Object specified in the input Siebel message.

This workflow is shown in the following figure: Query Read From File, Context Read From File, Product Selection Service API, WriteXMLtoFile.



Workflow Description

This workflow does the following:

- **Query Read From File.** Reads the contents from the xml file and converts to SiebelMessage propertyset, which may include integration object for EAI, searchspec that would be applied on the following query and so on. A sample input file is the following:

```
<?xml version="1.0" encoding="UTF-8" ?>
 <?Siebel-Property-Set EscapeNames="false"?>
- <SiebelMessage MessageId="" IntObjectName="PDS Publish Catalog Interface"
MessageType="Integration Object" IntObjectFormat="Siebel Hierarchical">
 - <ListOfPublishCatalog>
<ProductCatalog searchspec="EXISTS ([Channel Partner] like 'UANPartner1')" />
 </ListOfPublishCatalog>
 </SiebelMessage>
```

- **Context Read From File.** Reads the contents from the context xml file and converts to context propertyset. A sample context file is the following:

```
 <?xml version="1.0" encoding="UTF-8" ?>
 <?Siebel-Property-Set EscapeNames="false"?>
 <Context BusObjName="Quote" BusCompName="Quote" SignalMode="Quote"
```

```
ContextRowId="2-47J4V" />
```

- **Product Selection Service API.** Calls the PublishCatalog method of the PDS Product Data Service. The input arguments SiebelMessage and Context are from the output of the previous two steps.
- **WriteXMLtoFile.** Converts the output PropertySet of the previous step into xml format and writes to the result file.

ISS Authoring Import Export Business Service Reference

The main purpose of this business service is to let you move versioned object definitions (VODs) between databases. It supports the following VODs:

- **ISS_ATTR_DEF - Product attribute Definition.** Uses the Integration Object ISS Attr Def ImpExp.
- **ISS_CLASS_DEF - Product Class Definition.** Uses the Integration Object ISS Class Def ImpExp.
- **ISS_PROD_DEF - Product Definition.** Uses the Integration Object ISS Prod Def ImpExp.
- **ISS_SIGNAL - Signal Definition.** Uses the Integration Object ISS Signal ImpExp IntObj.
- **ISS_VAR_MAP - Variable Map Definition.** Uses the Integration Object ISS Variable ImpExp IntObj.

In addition, this business service allows you to fix S_PROD_INT records created with the EIM process so they are usable by the ISS framework. In the case of versioned objects, it creates the minimum set of necessary records that is needed.

Note: You can import simple products using EIM and run a process to generate a default Workspace. However, you cannot use EIM to import customizable objects, which you must import using ISS Authoring Import Export Service.

Modeling Data Before Using ISS Import Export Authoring Service

To be able to use the standard approach successfully, users must take some precautions during modeling. VODs manually created on two databases are be treated as different objects even if their VOD Names match.

CAUTION: Therefore, you must never create 2 master copies of the same VOD manually on two different DBs. Instead move existing definitions from one DB to another through Import/Export, or ADM.

After import, you can freely modify replicated objects in any database and move them around.

Observe the following guidelines to make objects compatible when multiple development environments are required:

- Use only one development database to create new VODs.
- If a customer is upgrading from release earlier than Siebel CRM version 7.8:
    - Do not use upgraded development databases. Instead of using those databases, create a new one from single production master.
    - If you cannot possibly avoid using upgraded development databases, then do not use the VODs that were created during the upgrade of the development databases. Discard those definitions, and change the

names of all VODs and products to show clearly that they have been discarded. Then, import necessary definitions from your production database.

- Use ADM or the Export/Import process to populate other development databases with existing VOD definitions.
- If you must create a new VOD in some database other than the master development database, then first check the master database to see if such object already exists, and move the existing object to your development database. If it does not exist, then create it in your development database and immediately move it to the master development database, so other modelers do not create a duplicate.

Methods of the ISS Import Export Authoring Service

This business service includes the following methods:

- Data Export Methods:
  - *Export VOD Method*
  - *ExportFullVOD Method*
- Data Import Method:
  - *ImportVOD Method*
- Application Deployment Manager (ADM) specific Methods:
  - *ADMValidate Method*
  - *ADMPreview Method*
  - *ADMExport Method*
  - *ADMImport Method*
- Helper Methods:
  - *ReleaseSingleObject Method*
  - *ReleaseJwsObjects Method*
  - *Post_EIM_Upgrade Method*
  - *CleanupSingleObject Method*

ExportVOD Method

Exports stand-alone objects from a Workspace or Joint WorkSpace (JWS) project or a single VOD Definition, based on supplied parameter values. Based on the input, the method determines a set of objects to be exported and produces a list of EAI property sets, corresponding to each of them. The resulting set of PropSets is wrapped in a single PropSet and returned in the output argument.

Syntax

ISS Import Export Authoring Service.InvokeMethod(“ExportVOD", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ExportVOD
Input Arguments	In	Hierarchical property set which supplies the starting information about export content
Output Arguments	Out	Hierarchical property set which contains the resulting content

Properties of Input Arguments: Combination 1

Property	Value	Description
RootObjectType	ISS_JWS	Indicates that the data is exported from a Joint Workspace Project
RootObjectId	Row Id of JWS project	None

If this combination of values is passed to the method, then it considers each entry in the content of the JWS project as a Root object. The resulting combined property set will include all of them.

Properties of Input Arguments: Combination 2

Property	Value	Description
RootObjectType	ISS_ATTR_DEF or ISS_CLASS_DEF or ISS_PROD_DEF or ISS_SIGNAL or ISS_VAR_MAP	Indicates that the data is exported from a single root object.
VODObjectNum	Root's S_VOD.OBJECT_NUM	Points to the VOD to be exported
VODVersion	version number	Points to the version number to be used in export

If this combination of values is passed to the method, then it considers a single entry as a Root object. The resulting property set will include only this object (as a child property set).

Error Conditions

Might return generic error codes from Object Manager, From EAI or from service. These codes are mainly about the input format, for example, that a parameter is missing.

ExportFullVOD Method

Similar to the ExportVOD Method. In addition, for ISS_PROD_DEF and ISS_CLASS_DEF structures, for each root object, runs a recursive algorithm to collect a list of dependencies, a set of self-contained objects.

This set includes: parent class definition, attribute definitions used to create local attributes; relationship base class definition; product definitions referred in relationship domain as sub-products.

Resulting combined list of objects is used as a set of objects to be exported in a same way as in ExportVOD.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ExportFullVOD", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ExportFullVOD
Input Arguments	In	Depending on the set of properties, supplies the starting information about export content
Output Arguments	Out	Contains Resulting Property Set

Properties of Input Arguments: combination 1

Property	Value	Description
RootObjectType	ISS_JWS	Indicates that the data is exported from a Joint Workspace Project
RootObjectId	Row Id of JWS project	None

If this combination of values is passed to the method, then it considers each entry in the content of the JWS project as a Root object. The resulting combined property set will include all of them.

Properties of Input Arguments: combination 2

Property	Value	Description
RootObjectType	ISS_ATTR_DEF or ISS_CLASS_DEF or ISS_PROD_DEF or ISS_SIGNAL or ISS_VAR_MAP	Indicates that the data is exported from a single root object.
VODObjectNum	Root's S_VOD.OBJECT_NUM	Points to VOD to be exported

Property	Value	Description
VODVersion	version number	Points to the version number to be used in export

If this combination of values is passed to the method, then it considers a single entry as a Root object. The resulting property set includes only this object (as a child property set).

Elements and their Properties in Output Arguments:

The output argument property set is a wrapper around multiple elements. The type of the first element is TableOfContent, and it contains a summary of exported VOD structures.

Property	Value	Description
FullStructure	Y or N	Indicates whether PropSet is self-contained
NumOutputObjects	number	Counter of VOD objects included in PropSet

Child elements are typed according to VOD Types of each exported Object and represent a short description.

Property	Value	Description
VOD Type	object type	LIC values from MLOV ISS_VOD_TYPE
VOD Name	object name	None
VOD Object Num	object integration id	OBJECT_NUM value
IntObjectName	EAI object name	Specifies the integration object to be used for importing

Other elements are:

- PropSets, containing the results of EAI Query operation for a particular VOD object. The standard format, conforming to Integration Object definition (depending on VOD type).
- PropSets, containing the results of the EAI Query operation for additional structures, related to Product Promotions, in case one of the listed VODs was a header of a promotion object. The standard format, conforming to Integration Object definition (ISS Promotion).

Error Conditions

Might return generic error codes from Object Manager, From EAI or from service. These codes are mainly about input format, for example, that a parameter is missing parameter.

ImportVOD Method

This method:

- Decomposes the content of the input argument's PropSet into individual objects that are used as source Sub-Propsets.
- Locates appropriate VOD in the target database.
- Adjusts some values in source Sub-Propsets.
- Locks the object and overwrites Workspace version content with data supplied in the Sub-Propsets.
- Adds successfully imported object references to the JWS project content specified in the input arguments, which is locked for further editing by the user who initiated this import operation.

In the default mode, object resolution is performed by S_VOD.OBJECT_NUM (object identifier) matching (for example, field VOD Object Num of VOD ImpExp BC element), representing a stable Integration ID, persistent across different databases.

> **CAUTION:** The ImportVOD method treats VODs that are manually created on two databases as different objects, even if their VOD Names match. To avoid errors, read *Modeling Data Before Using ISS Import Export Authoring Service*.

In rare cases, when a one-time import is required, importing the full structure might be performed in a special mode where objects and their references are resolved by names, instead of IDs. This name-based import can be used for objects created in versions earlier than Siebel CRM 7.8. In this mode, the input property-set must contain a full structure. All VODs referenced any VOD in the property set must be present. For VODs, already present in the target database, the source OBJECT_NUMs and their references are replaced with target values. For new objects, the source OBJECT_NUMs are preserved. If two VODs have the same OBJECT_NUMs, then the application treats this conflict as an unresolvable conflict. The resulting imported objects are different objects and will not map to each other in the standard scenario.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ImportVOD", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ImportVOD
Input Arguments	In	Property set with the input parameters containing VOD structures
Output Arguments	Out	Contains a description of the performed actions. In case of conflicts, it contains a report string in the Warning property.

Property set input arguments must have the same structure as the property set output arguments described in *Export VOD Method*.

Properties of Input Arguments

Property	Value	Description
RootObjectType	ISS_JWS	Indicates that the data is imported from a Joint Workspace Project.
RootObjectName	JWS project Name	Name of JWS Project to be used for import. **Note:** A new JWS project of this name must have been created in Administration - Products, Workspace Projects view.
NoTargNameChange	Y or N	Forbid VOD and Product name change in target database. Optional Default = N **Note:** If you need to change the product name, as part of the product import, then this parameter must be set to Y.
NoSrcNameChange	Y or N	Forces strict preservation of source VOD and Product names. Optional Default = N
ReportOnly	Y or N	Forces the process to end immediately after the initial analysis of input property set is done to examine the potential problems of VOD mapping. Optional Default = N **Note:** If ReportOnly = Y is one of the input arguments, then the data is not committed to the database. Only an analysis of the export file is done for possible conflicts. The log file in this case will always have Import Status = Aborted. This input argument is used to validate the file being imported and reports the possible conflicts without doing an import.
UseNamesAsKeys	Y or N	Triggers 7.7 style import Optional Default = N

Property	Value	Description
		Note: In order to trigger Name-Based Import, you must set the input argument UseNameAsKeys = Y.
AutoProceed	Y or N	Allows automatic conflict resolution if possible. Optional Default = N **Note:** If you are doing a Name-Based Import (UseNamesAsKeys=Y), then you must also set input argument AutoProceed=Y.

Modifying Defaults

The following default settings can be modified through repository changes:

- NoTargNameChange. The default is set to Y if the business service ISS Authoring Import Export Service has a User Property VOD_NO_TRG_NAME_CHANGE = TRUE, or if Applet ISS VOD Import Popup Applet has User Property NoTargNameChange = Y or N. The applet user property overrides the business service user property.
- NoSrcNameChange. The default is set to Y if the business service ISS Authoring Import Export Service has a User Property VOD_NO_SRC_NAME_CHANGE = TRUE, or if Applet ISS VOD Import Popup Applet has User Property NoSrcNameChange = Y or N. The applet user property overrides the business service user property.
- AutoProceed. The default is set to Y or N if the Applet ISS VOD Import Popup Applet has a User Property AutoProceed = Y or N.
- ReportOnly. The default is set to Y or N if the Applet ISS VOD Import Popup Applet has a User Property ReportOnly = Y or N.

Conflict Resolutions

By default, conflicting records are not imported. If you want to resolve conflicts automatically, you must set one of the following parameters:

- Set the property AutoProceed of the input arguments to the ImportVOD method to Y.
- Set User Property AutoProceed of the Applet ISS VOD Import Popup Applet to Y.

By default, both of these parameters are set to N. You must set one or the other to Y to enable automatic conflict resolution.

Automatic conflict resolution is done in the following way:

- For same Object Num, the target or source VOD's Name wins depending on the parameters, if possible.
- For different Object Num, and same Name (VOD name or Product Name), the target name remains unchanged. A new object name is appended with Object Num in parenthesis, if possible.

Problems are reported, in string format, in the output parameter of the PropSet Warning.

Properties of Output Argument

Property	Value	Description
Warning	string	*Success* or all encountered problems. (List of problems might be incomplete if the method exited with fatal error.)

One Child Element of Type Object

Property	Value	Description
RootObjectType	ISS_JWS	Used by Application Deployment Manager (ADM). For more information, see *Siebel Application Deployment Manager Guide* .
RootObjectName	name of JWS	Used by Application Deployment Manager (ADM). For more information, see *Siebel Application Deployment Manager Guide* .
Import Status	*Success*, *Aborted*, *Partial*, or *Failed*	Overall status of JWS import.
ErrorMessage	ErrorMessage	This error message is present only in case of catastrophic error (if Imp.Status == Failed)

The set of elements of the type determined by MLOV ISS_VOD_TYPE, with each element corresponding to each individual VOD from the input arguments.

Property	Value	Description
VOD Type	ISS_ATTR_DEF,ISS_CLASS_DEF, ISS_PROD_DEF	The type of the object
Src VOD Object Num	Persistent id	Source integration ID
Src VOD Name	VOD name	Source VOD Name
Src Product Name	Product name	Source product name (only for products)
Src Org Name	Organization name	Source organization name (only for products)
Src Vendor Name	Vvendor name	Source vendor name (only for products)

Property	Value	Description
Trg VOD Object Num	Persistent id	Target integration ID
Trg VOD Name	VOD name	Target VOD Name
Trg Product Name	Product name	Target product name (only for products)
Trg Org Name	Organization name	Target organization name (only for products)
Trg Vendor Name	Vendor name	Target vendor name (only for products)
Res VOD Object Num	Persistent id	Resulting integration ID
Res VOD Name	VOD name	Target VOD Name
Res Product Name	Product name	Resulting product name (only for products)
Res Org Name	Organization name	Resulting organization name (only for products)
Res Vendor Name	Vendor name	Resulting vendor name (only for products)
UK Conflict Status	*0*, *1*, *2*, *3*, *4*, *5*, or *6*	Internal conflict code used in UK conflict
Conflict Type	*Resolvable* or *Unesolvable*	Specifies whether the conflict can be resolved using automatic conflict resolution
Warning	Part of a global warning string related to this object	Warning message for this object (if a one exists)
Import Status	*Failed* or *Success*	The import status of this object

Error Conditions

Might return generic error codes from Object Manager, from EAI, or from service. These codes are mainly about the input format, for example, that a parameter is missing.

ADMValidate Method

Given an ADM project item's deployment filter, determines if it is valid. Performs the following validation checks:

- Checks that the deployment filter has a joint Workspace name specified.
- Checks that the given joint Workspace exists.

- Checks that if the [Full] export method is given, that it is Y or N.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ADMValidate", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ADMValidate
Input Arguments	In	Hierarchical property set containing VOD structures
Output Arguments	Out	none

Properties of Input Arguments

Property	Value	Description
EAI Search Spec	ADM deployment filter	ADM deployment filter

ADMPreview Method

Given an ADM session's deployment filter (such as an EAI Search Specification), returns a valid search specification for ISS Joint Workspace View. Basically, this function parses out the [Full] parameter in the deployment filter.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ADMPreview", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ADMPreview
Input Arguments	In	Unparsed filter
Output Arguments	Out	Search specification

Properties of Input Arguments

Property	Value	Description
EAI Search Spec	ADM deployment filter	ADM deployment filter

Properties of Output Arguments

Property	Value	Description
EAI Search Spec	Search Specification	Search Specification for ISS Joint Workspace View

ADMExport Method

Given an input argument property set with a valid SessionId, which represents the Row_Id of the deployed ADM session, exports the joint Workspace specified in the deployment filter of the ADM session. Export is done through a call to ExportAlgorithm. The deployment filter also specifies whether to do a full structure export. Returns an output argument property set with the exported VOD property set, along with additional bookkeeping information (JSW name, Status, Reason)

Syntax

ISS Import Export Authoring Service.InvokeMethod(“ADMExport", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ADMExport
Input Arguments	In	ADM Information
Output Arguments	Out	Same as ExportVOD with additional properties

Properties of Input Arguments

Property	Value	Description
SessionId	Row_Id	Row_Id of the deployed ADM session

Properties of Output Arguments

Property	Value	Description
JWSName	Name of JWS Project	Name of JWS project to be used when importing prop set on other side
Status	Processed or Failed	Result of operation
Reason	String	In case of failure contains the cause of failure

ADMImport Method

Given an input argument property set, which contains the exported joint Workspace data as a single child property set, import the data using ImportAlgorithm.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ADMImport", Input Arguments,Output Arguments);

Arguments

Same as *ImportVOD Method*.

ReleaseSingleObject Method

This helper method releases a particular object. If the object is locked by another user, then the method generates an error.

Syntax

ISS Import Export Authoring Service.InvokeMethod("ReleaseSingleObject", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ADMExport
Input Arguments	In	Information for releasing an object
Output Arguments	Out	None

Properties of Input Arguments

Property	Value	Description
ReleaseDT	Release date in canonical format (user time)	Release date in canonical format (user time)
RootObjectType	ISS_ATTR_DEF or ISS_CLASS_DEF or ISS_PROD_DEF or ISS_SIGNAL or ISS_VAR_MAP	Type of the object (LIC value from ISS_VOD_TYPE mlov)
RootObjectId	Id of the object	S_VOD.OBJECT_NUM or S_PROD_INT.CFG_MODEL_ID

ReleaseJwsObjects Method

This method releases all objects contained in a particular Joint Workspace. The Release date of all objects is determined using the value of S_ISS_JWS.EFF_RELLEASE_DT field; if this field is empty, then the objects are released as of the current moment.

If one of the objects is locked by another user, then the method generates an error.

Note: Because each content object is published as a separate transaction, new versions are committed into the database one by one. Although they will have same effective start time, customizable product models are incomplete until all participating objects are published. Therefore, during this short period of loading data, dependencies among versions might temporarily result in unexpected structure or an error. To avoid such behavior, it is recommended that you specify effective start time in near future (such as Now() + 15 min), so all the participating objects are committed and effective by the time the new versions are accessible.

Syntax

ISS Import Export Authoring Service.InvokeMethod(“ReleaseJwsObject", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	ReleaseJwsObjects
Input Arguments	In	Information for releasing JWS content
Output Arguments	Out	None

Properties of Input Arguments

Property	Value	Description
RootObjectType	ISS_JWS	Keyword
RootObjectName	Name of the JWS record	Name of the JWS record

Post_EIM_Upgrade Method

This method locates all records in S_PROD_INT with missing or broken association with the S_VOD record and creates the necessary supporting structure for versioning.

Depending on the input property set parameters, the method can operate in two different modes: standard or staging mode.

For each identified entry in S_PROD_INT, it does the following:

- Creates an appropriate header record (S_VOD)
- Creates an appropriate non-released definition record (S_ISS_OBJ_DEF)
- Creates an appropriate non-released version record (S_VOD_VER)

In Staging mode, it continues by doing the following:

- Registers the newly created header record in the supplied Joint Workspace

In Standard behavior mode, it continues by doing the following:

- Creates appropriate released definition record (S_ISS_OBJ_DEF)
- Creates appropriate released version record (S_VOD_VER), with the current time as the Start Time of the version

In both modes, if you set the property ReportOnly = TRUE, then the upgrade does not occur and the method only generates a report of orphan records. In this case, the output property set's property Warning contains a report of errors or list of orphans.

> **Note:** In both modes, if you set the property ReportOnly = TRUE, then the upgrade does not occur and the method only generates a report of orphan records. In this case, the output property set's property Warning contains a report of errors or list of orphans.

Syntax

ISS Import Export Authoring Service.InvokeMethod(“Post_EIM_Upgrade", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	Post_EIM_Upgrade
Input Arguments	In	Information for releasing JWS content
Output Arguments	Out	Reporting problems

Properties of Input Arguments (Standard Mode)

Property	Value	Description
ReportOnly	TRUE or FALSE	Optional parameter triggering output that is only a report, with no upgrade.

Properties of Input Arguments (Staging Mode)

Property	Value	Description
RootObjectType	ISS_JWS	Keyword triggering staging mode
RootObjectName	Name of JWS project	Name of existing JWS project
ReportOnly	TRUE or FALSE	Optional parameter triggering output that is only a report, with no upgrade.

Properties of Output Arguments

Property	Value	Description
Warning	String	Report of errors or list of orphans records if ReportOnly is on.

Example

After using EIM to import data into S_PROD_INT, you must execute the method Post_EIM_Upgrade with the Input Argument ReportOnly = False to release the products. If ReportOnly = TRUE, then no changes are made to the database.

There are two modes for using this method:

- **Staging mode.** The resulting upgraded products are not published (released) automatically. Use staging mode if you are not testing the result prior to releasing the products. Instead, you test them from the interface before releasing them.
- **Standard mode.** Use the parameter ReportOnly = FALSE for the official run. Use ReportOnly = TRUE for testing purposes so that the database is not changed and the report specifies what data might be generated.

After EIM imports, you can use the business service simulator to execute with ReportOnly = FALSE to release the imported products. Execute using the business service simulator as follows:

- Name: ISS Authoring Import Export Service:
- Method: Post_EIM_Upgrade
- Test#: 1
- Properties: create 1 records:
- Name: ReportOnly:
- Value: FALSE

CleanupSingleObject Method

CAUTION: Use this method with extreme caution as it deletes all published versions of a product.

This helper method performs deep delete on all published versions of an object. It resets version numbering to 1. The Workspace version remains intact.

This method works even if the object is not locked.

Syntax

ISS Import Export Authoring Service.InvokeMethod(“CleanupSingleObject", Input Arguments,Output Arguments);

Arguments

Argument	Type	Description
MethodName	In	CleanupSingleObject
Input Arguments	In	Information for cleaning an object
Output Arguments	Out	None

Properties of Input Arguments

Property	Value	Description
RootObjectType	Possible values are: • ISS_ATTR_DEF • ISS_CLASS_DEF • ISS_PROD_DEF • ISS_SIGNAL • ISS_VAR_MAP	Type of the object, which is the LIC value from the ISS_VOD_TYPE list of values.
VODObjectNum	Id of the object	S_VOD.OBJECT_NUM or S_PROD_INT.CFG_MODEL_ID