Sterling Field Sales

IBM

Implementation Guide

Version 9.2

Sterling Field Sales

IBM

Implementation Guide

Version 9.2

Contents

Chapter 1. Introduction to IBM Sterling Field Sales Implementation

Sterling Field Sales enables users to manage opportunities and quotes effectively, which in turn, helps them close business deals more efficiently.

Opportunity management involves the creation and management of an opportunity and its associated quotes through the life cycle of the opportunity, until the opportunity is either won or lost.

Quote management involves the management of quotes that are associated with an opportunity.

For information about the Sterling Field Sales reference implementation, refer to the *Sterling Selling and Fulfillment Suite: Applications Reference Implementation Guide*.

Chapter 2. Initial Configurations to be Performed in Channel Applications Manager

Before you start using the Sterling Field Sales application, you must ensure that the following configurations have been performed appropriately in the Channel Applications Manager:

- Quote administration
- Opportunity administration

For more information about performing the quote administration configurations and opportunity administration configurations, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 3. An Introduction to Opportunity Life Cycle

An opportunity is a container for holding a quote or related alternative quotes that can be presented to a customer. Throughout its life cycle, an opportunity moves through different statuses.

An opportunity's life cycle is represented by a pipeline. The pipeline determines the statuses that an opportunity can be in, and the transactions that process an opportunity in a particular status. An opportunity may move through multiple statuses in a particular stage of its life cycle.

The Opportunity Fulfillment pipeline is used to process opportunities within IBM® Sterling Field Sales. For more information about this pipeline, refer to *Sterling Selling and Fulfillment Foundation: Setting Up Quotes in Distributed Order Management*.

Chapter 4. An Introduction to Quote Life Cycle

A quote is a document that allows a customer to see a predetermined set of items and quantities at a predetermined price, to be delivered on a specific date.

A quote's life cycle is represented by a pipeline. The pipeline determines the statuses that a quote can be in and the transactions that process a quote in a particular status. A quote may move through multiple statuses in a particular stage of its life cycle.

The Quote Fulfillment pipeline is used to process quotes within IBM Sterling Field Sales. For more information about this pipeline, refer to *Sterling Selling and Fulfillment Foundation: Setting Up Quotes in Distributed Order Management*.

Chapter 5. Setting User Preferences

A Field Sales Representative may want to set user preferences to either display or hide the Navigation panel in the Sterling Field Sales application. The Navigation panel enables a Field Sales Representative to perform tasks, such as searching for an opportunity or a quote, creating an opportunity or a quote, and so on. If the Navigation panel is hidden, the related tasks that a user can perform using this panel are available as task hyperlinks. These task hyperlinks are displayed when a user clicks the **Actions** button in the screen.

Solution

Sterling Field Sales enables a user to set user preferences to either display or hide the Navigation panel. When a user clicks the **Preferences** hyperlink in the application header bar, the getUserUiStateList API is called to retrieve the user preferences that have been set for the corresponding user. After a user clicks **Save** in the Preferences dialog box, the manageUserUiState API is called to save the changes.

End-User Impact

None.

Implementation

None.

Chapter 6. Managing an Opportunity

An opportunity is a container for holding a quote or related alternative quotes that can be presented to a customer interested in purchasing a set of items. Sterling Field Sales enables users to create opportunities, and modify the details of the opportunities after they are created.

As part of managing an opportunity, a user can perform the following tasks:
* Creating an Opportunity
* Viewing Opportunity Details
* Modifying Opportunity Details
* Modifying Customer Details
* Creating a Quote for an Opportunity
* Creating an Alternative Quote
* Copying a Quote to Create a New Quote and New Opportunity for a Customer
* Rejecting a Quote on Behalf of a Customer
* Marking an Opportunity as Lost

Chapter 7. Creating an Opportunity

An opportunity is a container for holding a quote or related alternative quotes that can be presented to a customer interested in purchasing a set of items. A Field Sales Representative may get leads about a prospective customer or an existing customer's interest in purchasing a set of items that are available in a catalog. An opportunity must be created to capture relevant information about the customer and the quote that can be presented to the customer.

Solution

Sterling Field Sales enables a user to create an opportunity. If a user has access to multiple enterprises, the user can select the relevant enterprise and create an opportunity. The getOrganizationList API is called to retrieve the enterprises the user has access to. A user can create an opportunity for an existing customer whose records already exist in the system, or for a new customer.

Users can assign an opportunity to themselves as the owner of the opportunity or choose a different user as the owner. Users can also identify a co-owner for the opportunity by searching for the user. The getUserList API is called to retrieve the list of users.

A user can specify an opportunity's lead origin, probability of success (in %), and opportunity value. You can configure the codes for the lead origin in the Channel Applications Manager.

When creating an opportunity for an existing customer, a user can search for the customer for whom the opportunity has to be created. You can configure the customer type that should be selected by default in the Customer Search dialog box. The getCustomerList API is called to retrieve the list of customers. After the customer is selected, the getCustomerDetails API is called to retrieve the details of the customer to be displayed in the Create Opportunity screen.

The names of the city and state in which the customer or the contact resides can be derived from the **country or region** and **Postal Code** fields, if the enterprise has defined a region schema in the Channel Applications Manager. The getCommonCodeList API is called to retrieve the values displayed in the **country or region** drop-down list. The country or region codes are stored as common codes with country or region as the code type. The getRegionList API is called to retrieve the names of the city and state. The region schema that is called corresponds to the enterprise to which the user belongs. If multiple values are retrieved for the city and state names, the first value returned by the getRegionList API is displayed.

When creating an opportunity for a new business customer, the getCurrencyList API is called to populate the **Default Currency** drop-down list in the Create Opportunity screen.

The manageOpportunity API is called to create the opportunity after the user provides the necessary information.

End-User Impact

This section explains the end-user impact of this functionality:

- Based on the configuration, a user can create and search for customers based on the customer type.
- Based on the configuration, either the **Business** radio button or the **Consumer** radio button is selected by default in the Customer Search dialog box.
- Based on the configuration, the **State** field is displayed as a drop-down list or a box when the user is creating an opportunity for a new customer.

Implementation

This section explains the configurations for this functionality:
- The notes entered when a user creates an opportunity are saved in the opportunity using the SAA_OPPORTUNITY_CREATED note type.
- You can configure the **Enable Customer Type Search** rule to enable users to create and search for a customer based on the type of customer.
- You can configure the customer type that is to be selected by default in the Customer Search dialog box by configuring the **Customer Type To Be Defaulted In Search** rule.
- You can configure the region schema to be used for deriving the names of the city and state when creating an opportunity for a new customer by configuring the country or region and state display rules.
- You can configure the codes to be used for the **Lead Origin** drop-down list.

  For more information about configuring these rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 8. Viewing Opportunity Details

A Field Sales Representative may sometimes want to view the details of an opportunity, such as the opportunity's status, owner, and so on. For example, a Field Sales Representative may want to know the Anticipated Close Date of a particular opportunity so that the Field Sales Representative can start working on it if the date is close to the current date.

Solution

Sterling Field Sales enables a user to search for an opportunity and view the details of a particular opportunity.

Users can search for an opportunity using the following options:
- Quick Search: A user can enter the name of the opportunity as the search criteria and search for the opportunity.
- Advanced Opportunity Search: A user can enter more restrictive search criteria to narrow down the results and search for an opportunity.

If a user has access to multiple enterprises, the **Enterprise** drop-down list is displayed in the Search Criteria panel of the Opportunity Search screen. The getOrganizationList API is called to retrieve the list of enterprises. The queryTypeList API is called to populate the operators or the query types for the following fields:
- Opportunity Name
- Success %
- Opportunity Value
- Anticipated Close Date

When searching for an opportunity, a user can search for the customer for whom the opportunity was created. You can configure the customer type that is to be selected by default in the Customer Search dialog box. The getCustomerList API is called to retrieve the list of customers that match the search criteria. A user can also search for the user who has been assigned as the owner of the opportunity. The getTeamList API is called to populate the values in the **Team** drop-down list in the User Search dialog box. The getUserList API is called to retrieve the list of users matching the search criteria.

If only one opportunity matches the search criteria, the Opportunity Details screen is displayed. If multiple opportunities match the search criteria, the list of opportunities matching the search criteria is displayed. The getOpportunityList API is called to retrieve the list of opportunities that match the search criteria.

Users can view the details of an opportunity in the Opportunity Details screen. Users can view information such as the opportunity's name, status, the details of the customer for whom the opportunity has been created, and so on.

The getCustomerDetails API is called to retrieve the available customer contacts and determine the type of customer. The getOpportunityDetails API is called to retrieve the details of the opportunity.

End-User Impact

This section explains the end-user impact of this functionality:

- Based on the configuration, a user can search for customers based on the customer type.
- Based on the configuration, either the **Business** radio button or the **Consumer** radio button is selected by default in the Customer Search dialog box.

Implementation

This section explains the configurations for this functionality:

- You can configure the **Enable Customer Type Search** rule to enable users to search for a customer based on the type of customer.
- You can configure the customer type that is to be selected by default in the Customer Search dialog box by configuring the **Customer Type To Be Defaulted In Search** rule.

  For more information about configuring these rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 9. Modifying Opportunity Details

Based on the requirements, a Field Sales Representative may have to modify the details of an opportunity. For example, if a user, who has been assigned the role of owner for an opportunity, has resigned from the company, the opportunity must be reassigned to a different user.

Solution

Sterling Field Sales enables a user to modify the details of an opportunity, such as the opportunity's name, owner and co-owner, lead origin, probability of success (in %), and so on.

A user can search for the users to whom they want to reassign an opportunity, either as its owner or co-owner. The getTeamList API is called to populate the values in the **Team** drop-down list in the User Search dialog box. The getUserList API is called to retrieve the list of users matching the search criteria. The manageOpportunity API is called to save the changes made to the opportunity.

A user can modify the lead origin for an opportunity. You can configure the codes for the lead origin in the Channel Applications Manager. The getCommonCodeList API is called to populate the values in the **Lead Origin** drop-down list.

End-User Impact

None.

Implementation

You can configure the codes to be used for the **Lead Origin** drop-down list. For more information about defining opportunity lead origins, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 10. Modifying Customer Details

After an opportunity or a quote is created, a Field Sales Representative may have to modify the details of a customer in the opportunity or the quote. For example, if the address of a customer has changed, a Field Sales Representative may have to make the required modifications in the opportunity or the quote.

Solution

A user can modify the details of a customer by performing the following tasks:

- Assigning a different contact to a business customer
- Modifying the address of a business customer or consumer customer

A user can assign a different contact to a business customer. The getCustomerDetails API is called to populate the **Contact** drop-down list in the Opportunity Details screen. A user can also create a new contact and assign the contact to the business customer in the Add New Contact dialog box.

A user can modify the address information of a business customer or a consumer customer in an opportunity. A user can either select an existing address or add a new address for the customer or the contact. The getCustomerDetails API is called to populate the **Existing Addresses** drop-down list in the Edit Address dialog box.

In the Add New Contact dialog box and the Edit Address dialog box, the names of the city and state in which the customer or the contact resides can be derived from the **country or region** and **Postal Code** fields, if the enterprise has defined a region schema in the Channel Applications Manager.

The getRegionList API is called to retrieve the names of the city and state. The region schema that is called corresponds to the enterprise to which the user belongs. If multiple values are retrieved for the city and state names, the first value returned by the getRegionList API is displayed. For more information about configuring the way in which the city and state names are to be derived using the **country or region** and **Postal Code** fields, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

When an address is either added or modified, the verifyAddress API is called to verify the address. If the address verification fails and the Address Verification System (AVS) returns multiple addresses, the user can select an address from the list. The manageCustomer API is called to save the changes made to the contact and address information of a customer. When a new address is added for a customer or a contact, it is saved against the customer or the contact as an additional address. If the AllAddressesVerified flag is passed as "Y" in the input XML of the verifyAddress API, Sterling Field Sales does not attempt to verify any addresses on the opportunity or quote unless any of them is changed.

The manageOpportunity API is called to save the changes made to the opportunity.

Solution of Address Verification User Exit

The YCDVerifyAddressWithAVSUE user exit allows Sterling Field Sales to interact with the external AVS. The YCDVerifyAddressWithAVSUE user exit is called from the verifyAddress API. The user exit returns an XML output, which is the same as the verifyAddress API output. For more information about the YCDVerifyAddressWithAVSUE user exit, see the *Sterling Selling and Fulfillment Foundation: Javadocs*.

The `AVSReturnCode` attribute returned by the verifyAddress API and the YCDVerifyAddressWithAVSUE user exit can have the following values:

- VERIFIED if the address is valid.
- FAILED if the address is not valid.
- AVS_DOWN if the AVS system could not be connected with.

End-User Impact

Based on the configuration, the **State** field is displayed as a drop-down list or a box in the Add New Contact dialog box and the Edit Address dialog box.

Implementation

This section explains the configurations for this functionality:

- You can configure the region schema to be used for deriving the names of the city and state when a user enters the address of a new contact for a business customer or modifies the address of a customer. For more information about configuring the country or region and state display rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- You must implement the YCDVerifyAddressWithAVSUE user exit to verify the addresses with the AVS.

Chapter 11. Marking an Opportunity as Lost

If an opportunity no longer serves any purpose, a Field Sales Representative may have to mark that opportunity as Lost. For example, when an opportunity is still in the Inquiry status, a customer may have purchased the items from another seller. In such a scenario, the Field Sales Representative who has been assigned that opportunity may want to mark the opportunity as Lost and provide the reasons for losing the opportunity.

Solution

A user can mark an opportunity as Lost and provide the reasons for losing that opportunity. You can configure the loss reason codes in the Channel Applications Manager. The getCommonCodeList API is called to populate the **Loss Reason Code** drop-down list. These reason codes are stored as common codes with LOST_REASON_CODE as the code type. This data is cached. The changeOpportunityStatus API is called to change the status of the opportunity.

End-User Impact

None.

Implementation

This section explains the configurations for this functionality:

- You can configure the loss reason codes to enable users to provide reasons for marking an opportunity as Lost. For more information about defining opportunity loss reasons, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- The notes entered when a user marks an opportunity as Lost are saved in the opportunity using the SAA_MARK_AS_LOST note type.

Chapter 12. Creating an Alternative Quote

At times, a Field Sales Representative may want to present alternative quotes to a customer. For example, let us suppose that a customer is interested in purchasing 10 computers. A Field Sales Representative may want to create a quote for 10 desktop computers, offering a discount of 5% on the quote total. To provide the customer with another option to choose from, the Field Sales Representative may want to create another quote for 5 desktop computers and 5 laptops, offering a discount of 7% on the quote total.

Solution

A user can create an alternative quote in an opportunity by copying an existing quote. The copyOrder API is called to create the alternative quote. When an alternative quote is created, all the attributes defined in the Copy Order template for the quote fulfillment process type in the Applications Manager are copied from the original quote to the new quote.

End-User Impact

None.

Implementation

You can define the attributes that should be copied when copying a quote in the Copy Order template in the Applications Manager. For more information about defining process type details, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Chapter 13. Creating a Quote for an Opportunity

After creating an opportunity, a Field Sales Representative must be able to create a quote for that opportunity so that the Field Sales Representative can capture relevant information about the items a customer is interested in purchasing, and the payment terms and shipment terms.

Solution

A user can create a quote for an opportunity from the Opportunity Details screen. The createOrder API is called to create the quote. By default, the first quote created for an opportunity is marked as the Primary Quote for an opportunity.

End-User Impact

None.

Implementation

None.

Chapter 14. Copying a Quote to Create a New Quote and New Opportunity for a Customer

A Field Sales Representative may sometimes want to copy a quote that is associated with an opportunity to create a new quote and a new opportunity. For example, a Field Sales Representative may want to use the product information and payment terms of an existing quote for a different quote and opportunity.

Solution

A user can copy a quote that is associated with an opportunity to create a new quote and opportunity. The copyOrder API is called to create the new quote and opportunity. When the new quote is created, all the attributes defined in the Copy Order template for the quote fulfillment process type in the Applications Manager are copied to the new quote from the original quote.

End-User Impact

None.

Implementation

You can define the attributes that should be copied when copying a quote in the Copy Order template in the Applications Manager. For more information about defining process type details, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

Chapter 15. Browsing Catalog in IBM Sterling Field Sales

A Field Sales Representative may want to browse the catalog to search for and view the details of an item a customer has shown interest in purchasing. For example, let us suppose that a Field Sales Representative has a prospective customer who wants to purchase office furniture for a new office that the customer wants to set up. In such a scenario, the Field Sales Representative may want to browse the catalog to see the furniture items that are available.

Solution

Sterling Field Sales enables a user to browse the catalog, search for items in the catalog, and view the details of the items. A user can either browse the catalog from one of the home pages or from the Quote Details screen. When browsing the catalog from the home page, if a user has access to multiple enterprises, the user can select the enterprise that owns the catalog. The getOrganizationList API is called to retrieve the enterprises the user has access to.

A user can search for an item within the search results by providing a search term. You can configure the search terms that can be used by a user to search for an item. The search terms are configured in the `CatalogSearchConfigProperties.xml` file that is located in the `resources.jar/template/resource` directory. For example, to enable a user to search for an item by the Item ID, you must set the **XMLName** attribute to ItemID, and **Searchable** attribute to Y.

In the out-of-the-box implementation, a user can search for an item by providing any of the following search criteria:

- Item ID
- Short Description
- Long Description
- Extended Description
- Item Keywords
- Manufacturer Name
- Manufacturer Item ID
- Manufacturer Item Description
- Category Keywords
- Datasheet Content

A user can use wild card characters to search for an item. For example, if the user is searching for an item with XYZ123 as Item ID, the user can search for the item by providing XY* as the search criteria. If the user provides XY "Z" as the search criteria, the search will be conducted for either of the terms, XY or Z. However, if the user provides "XY Z" as the search criteria, the search will be conducted for both the terms, XY and Z. The searchCatalogIndex API is called to retrieve the list of items matching the search criteria.

Based on the search criteria, the items and the categories they belong to are displayed. The searchCatalogIndex API is called to retrieve the items and the categories. A user can search within the search results by entering the appropriate search criteria in the **Search in Results** box under the Search panel. Additionally,

the user can further narrow down the search results using the Browse Categories panel and the Narrow By panel. The user can click the *<category>* hyperlink under the Browse Categories panel or the *<attribute value>* hyperlink under the Narrow By panel to narrow down the search results by the item's category or attributes. The searchCatalogIndex API is called to retrieve the items.

As a user narrows down the search, bread crumbs are generated at every step. Clicking a bread crumb navigates the user to that step of the search and removes the subsequent bread crumbs.

A user can view an image adjacent to the corresponding item in the Search Results panel if you have associated an image with the item in the Sterling Business Center application. For more information about defining an item's main image as part of the merchandising and sales details of an item, refer to the *Business Center: Item Administration Guide*.

In the Search Results panel, a user can select the number of search results to be displayed on each page. A user can also sort the search results by **Relevancy**, and the **Product ID** and **Short Description** in either an ascending order or a descending order. The sorting criteria displayed in the **Sort By** drop-down list are configured in the `CatalogSearchConfigProperties.xml` file. The getSearchIndexFieldList API is called to retrieve the sorting criteria. The searchCatalogIndex API is called to sort the items based on the sorting criteria.

A user can view the details of an item, such as the item's description, unit of measure, unit price, whether the item is configurable, and so on, by either clicking the item's image or the *<short description of the item (Item ID)>* hyperlink. Additionally, a user can view the following information based on the item type:

- The **Overview** tab displays the Extended Description of the item.
- The **Specifications** tab displays the attributes and values associated with the attributes, and the physical dimensions associated with the item.
- If the item is a bundle, physical kit, or logical kit item, the **Components** tab displays the components of the item.

  Note: The **Components** tab is not displayed for a configurable bundle item that does not contain any configurations.
- If the item is a variation item, the **Variation** tab displays the details of the variations assigned to the item.
- If the item has any related items associated with it, the related items are displayed in separate tabs based on the configuration. For example, if the item has cross-sell items associated with it, the **Cross-Sell** tab displays the details of the cross-sell items based on the configuration.

The getAssociationRelationshipList API is called to verify which association types have the **Show In UI** rule enabled. The getCompleteItemList API is called to retrieve the details of the item.

Solution for Pagination When Browsing a Catalog

When browsing the catalog, if the number of items matching the search criteria is spread across multiple pages, the page numbers are displayed as hyperlinks. A user can navigate to any of the pages by clicking the corresponding *<page number>* hyperlink. The searchCatalogIndex API is called to retrieve the page to be displayed. The value of the `PageSize` attribute passed in the searchCatalogIndex

API changes dynamically based on the value selected in the **Results per page** drop-down list.

Solution for Verifying Item Availability When Browsing a Catalog

A user can verify whether an item is currently available. The findInventory API is called to retrieve the availability details of the item.

End-User Impact

This section explains the end-user impact of this functionality:

- An image is displayed for an item only if an image has been associated with the item when defining the item in the Sterling Business Center application.
- The related items of an item are displayed only if the **Show In UI** rule is enabled for the applicable association types in the Channel Applications Manager.

Implementation

This section explains the configurations for this functionality:

- To generate the catalog search index, you must configure the Catalog Index Build transaction, run the agent server, and then trigger the agent configured for the transaction. The Catalog Index Build transaction builds the catalog index file that is used during catalog search. This index file ensures that item searches are accomplished with fast response time. For more information about defining transactions, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

  Note: Ensure that the searchindex.rootDirectory property is set appropriately to point to the directory in which the generated search index files are located.
- You can associate an image to an item using the Sterling Business Center application to enable a user to view the item's image in the search results. For more information about defining an item's main image, refer to the *Business Center: Item Administration Guide*.
- You can configure the search terms that can be used to search for an item in the `CatalogSearchConfigProperties.xml` file.
- You can configure the sorting criteria to be used for sorting the items displayed in the search results in the `CatalogSearchConfigProperties.xml` file.
- You can configure the related items that should be displayed for an item. For example, to display the alternative items of an item, you must enable the **Show In UI** rule for the **Alternative** association type in the Channel Applications Manager. For more information about configuring item association rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 16. Managing a Quote

A quote is a commercial document that allows a prospective buyer to see a predetermined set of items and quantities at a predetermined price, to be delivered on a specific date. Sterling Field Sales enables users to create quotes, and modify the details of the quotes after they are created.

In the user interface, an element (a field or an action) is enabled or disabled based on the quote's status provided a corresponding **ModificationType** exists for that particular user interface element. The **ModificationType** for a user interface element is available in the output XML of the getCompleteOrderDetails API, and this is used to enable or disable the user interface elements.

As part of managing a quote, a user can perform the following general tasks:
- Creating a Quote
- Viewing a Quote Summary
- Marking a Quote as the Primary Quote
- Creating an Alternative Quote
- Copying a Quote to Create a New Quote and New Opportunity for a Customer
- Adding and Viewing a Note
- Abandoning a Quote
- Validating a Quote
- Presenting a Quote to a Customer
- Accepting a Quote on Behalf of a Customer
- Rejecting a Quote on Behalf of a Customer
- Converting a Quote to an Order

Managing Customer Details

A user can manage the customer-related information in a quote from the **Customer** tab of the Quote Details screen. The getCompleteOrderDetails API and the getCustomerDetails API are called to retrieve the details to be displayed in the **Customer** tab of the Quote Details screen.

A user can perform the following customer-related tasks:
- Modifying Quote Name and Expiration Date
- Modifying Customer Details

Managing Item Details

A user can manage the item-related information in a quote from the **Products** tab of the Quote Details screen.

The following table describes the APIs that are called when a user clicks the **Products** tab of the Quote Details screen:

API	Purpose
getOrderLineList	To retrieve the details of the line items in the quote.
getCompleteOrderDetails	To retrieve the details of the quote.

A user can perform the following tasks from the **Products** tab:
- Adding Items to a Quote
- Adding Related Items to a Quote
- Performing Price Adjustments for a Quote
- Modifying Item Quantity in a Quote
- Deleting an Item from a Quote
- Moving Items Between Quote Lines and Recommended Lines

Managing Payment Terms and Shipment Terms

A user can manage the payment terms and shipment terms in a quote from the **Terms** tab of the Quote Details screen.

The following table describes the APIs that are called to retrieve the details to be displayed in the **Terms** tab of the Quote Details screen:

API	Purpose
getCompleteOrderDetails	To retrieve the details of the quote.
getScacList	To validate the items in the quote.
getCommonCodeList	To retrieve the payment terms and shipment terms, and the level of service for the quote. The payment terms are stored as common codes with PAYMENT_TERMS as the code type. The level of service is stored as common codes with LEVEL_OF_SERVICE as the code type.

A user can modify the payment terms and shipment terms. For more information about modifying payment terms and shipment terms, refer to the topic "Modifying the Payment Terms and Shipment Terms in a Quote".

Chapter 17. Creating a Quote

A quote is a document that allows a prospective customer to see a predetermined set of items and quantities at a predetermined price, to be delivered on a specific date. A Field Sales Representative must be able to create quotes that can be used to store all the information that is required during the quoting process, including customer details, information about the items and the adjustments made to the price of items, and the payment terms and shipment terms that will be used in the order if the quote is converted to an order.

Solution

Sterling Field Sales enables a user to create a quote for an existing customer, whose records already exist in the system, or a new customer. If a user has access to multiple enterprises, the user can select the relevant enterprise for which the quote is to be created. The getOrganizationList API is called to retrieve the enterprises the user has access to.

You can configure the default expiration period of a quote if the user does not specify an expiration date. The expiration date of a quote is determined by the following configuration rules:

* Recalculate Expiration Date
* Default Expiration Period (Number Of Days)

If a user does not specify the expiration date for a quote, the expiration date is computed when the quote is sent to the customer for the first time as follows:

`Current date + Default Expiration Period (Number of Days)`

Note: The expiration date is recalculated only if the **Recalculate Expiration Date** rule is enabled.

If the expiration date is already stamped and persisted in the database for a quote, the expiration date on the quote is not computed when the quote is sent to the customer.

When creating a quote for an existing customer, a user can search for the customer for whom the quote has to be created. You can configure the customer type that is to be selected by default in the Customer Search dialog box. The getCustomerList API is called to retrieve the list of customers. After the customer is selected, the getCustomerDetails API is called to retrieve the details of the customer to be displayed in the Quote Details screen.

The names of the city and state in which the customer or the contact resides can be derived from the **country or region** and **Postal Code** fields if the enterprise has defined a region schema in the Channel Applications Manager. The getCommonCodeList API is called to retrieve the values displayed in the **country or region** drop-down list. The country or region codes are stored as common codes with country or region as the code type. The getRegionList API is called to retrieve the names of the city and state. The region schema that is called corresponds to the enterprise to which the user belongs. If multiple values are retrieved for the city and state names, the first value returned by the getRegionList API is displayed.

When creating a quote for a new business customer, the getCurrencyList API is called to populate the **Default Currency** drop-down list in the Quote Details screen. The manageCustomer API is called to save the customer information entered for the new customer.

The createOrder API is called to create a quote.

End-User Impact

- Based on the configuration, a user can create and search for customers based on the customer type.
- Based on the configuration, either the **Business** radio button or the **Consumer** radio button is selected by default in the Customer Search dialog box.
- Based on the configuration, the **State** field is displayed as a drop-down list or a box when a user is creating a quote for a new customer.

Implementation

This section explains the configurations for this functionality:

- The notes entered when a user creates a quote are saved in the quote using the SAA_QUOTE_CREATED note type.
- You can configure the **Enable Customer Type Search** rule to enable users to create and search for a customer based on the type of customer.
- You can configure the customer type that is to be selected by default in the Customer Search dialog box by configuring the **Customer Type To Be Defaulted In Search** rule.
- You can configure the region schema to be used for deriving the names of the city and state when creating an opportunity for a new customer, by configuring the country or region and state display rules.
- You can configure the **Recalculate Expiration Date** rule to enable the quote expiration date to be computed before a quote is presented to a customer.
- You can configure the **Default Expiration Period (Number Of Days)** rule appropriately to specify when a quote should expire by default if a user does not specify the quote expiration date.

  For more information about these rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 18. Viewing a Quote Summary

A Field Sales Representative may sometimes want to view a quote's summary, such as the quote's status, the items added to the quote, and so on. For example, after a Field Sales Representative has presented a quote to a customer, if the customer neither accepts nor rejects the quote for a considerable period of time, the Field Sales Representative may want to know the quote's expiration date so that the user can follow up with the customer.

Solution

Sterling Field Sales enables a user to search for a quote and view the details of that quote.

A user can search for a quote using the following options:

- Quick Search: A user can enter the name of the quote as the search criteria and search for the quote.
- Advanced Quote Search: A user can enter more restrictive search criteria and search for the quote.

If a user has access to multiple enterprises, the **Enterprise** drop-down list is displayed in the Search Criteria panel of the Quote Search screen. The getOrganizationList API is called to retrieve the list of enterprises. The queryTypeList API is called to populate the operators or the query types for the following fields:

- Quote Name
- Phone #
- Expiration Date

When searching for a quote, a user can search for the customer for whom the quote was created. You can configure the customer type that is to be selected by default in the Customer Search dialog box. The getCustomerList API is called to retrieve the list of customers that match the search criteria.

If only one quote matches the search criteria, the Quote Summary screen is displayed. If multiple quotes match the search criteria, the list of quotes matching the search criteria is displayed. The getOrderList API is called to retrieve the quotes that match the search criteria. The quotes displayed in the Search Results panel of the Quote Search screen can be sorted in either an ascending order or a descending order based on the **Quote Name** or the **Expiration Date**. The sorting criteria is defined in the metadata file, which is located in the `<INSTALL_DIR>/sfs/metadata/sortingorder` directory.

In the Quote Summary screen, a user can view the details of a quote such as the quote name, status, the details of the customer for whom the quote was created, the items that have been added to the quote, and so on.

The following table describes the APIs that are called to retrieve the details to be displayed in the Quote Summary screen:

API	Purpose
getCompleteOrderDetails	To retrieve the details of the quote.
getCommonCodeList	To retrieve the payment terms and shipment terms, and the level of service for the quote.

The payment terms are stored as common codes with PAYMENT_TERMS as the code type. The level of service is stored as common codes with LEVEL_OF_SERVICE as the code type. |

End-User Impact

* Based on the configuration, a user can search for customers based on the customer type.
* Based on the configuration, either the **Business** radio button or the **Consumer** radio button is selected by default in the Customer Search dialog box.

Implementation

This section explains the configurations for this functionality:

* You can configure the **Enable Customer Type Search** rule appropriately to enable users to search for a customer based on the type of customer.
* You can configure the customer type that is to be selected by default in the Customer Search dialog box by configuring the **Customer Type To Be Defaulted In Search** rule.

  For more information about configuring these rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Note: To improve the quote search performance, it is recommended that you create an index for the ORDER_NAME column of the YFS_ORDER_HEADER table.

Chapter 19. Marking a Quote as the Primary Quote

When there are multiple quotes for an opportunity, a Field Sales Representative can mark any one of the quotes as the primary quote. A quote that is marked as the primary quote is used for analysis and reporting. By default, the first quote created for an opportunity is considered as the primary quote.

Solution

The primary quote for an opportunity is identified by a check mark and a descriptive text, and the products that are contained in the primary quote are displayed in the Products related list on the opportunity screen.

The manageOpportunity API is called for marking a quote as the primary quote in an opportunity.

End-User Impact

None.

Implementation

None.

Chapter 20. Modifying Quote Name and Expiration Date

A Field Sales Representative may sometimes want to modify the name of a quote and the expiration date of a quote. For example, let us suppose that a Field Sales Representative, who is working on an important quote, does not receive a response from the customer for a considerable period of time after the quote has been presented to the customer. Because the expiration date of the quote is close to the current date, the Field Sales Representative may want to modify the quote expiration date.

Solution

Sterling Field Sales enables a user to modify the quote name and the quote expiration date. The changeOrder API is called to save the changes made to the quote.

End-User Impact

None.

Implementation

This section explains the configurations for this functionality:

- You can enable the **Recalculate Expiration Date** rule in the Channel Applications Manager to recalculate the quote expiration date when the changeOrder API is called. If this rule is enabled, the expiration date is calculated as follows:

  ```
  Expiration date + Default Expiration Period (Number of Days)
  ```

- You must configure the quote modification rules appropriately to allow a user to modify the name and expiration date of a quote in a particular status.

For more information about these rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 21. Adding Items to a Quote

A quote is a document that allows a prospective buyer to see a predetermined set of items and quantities at a predetermined price, to be delivered on a specific date. A Field Sales Representative must be able to add items to a quote.

Solution

Sterling Field Sales enables a user to add product items from the catalog to a quote. A user can add an item as a quote line item or a recommended line item.

A user can either enter the complete Item ID of the item and add that item to the quote, or browse the catalog to search for the item and add it to the quote. The getCompleteItemList API is called to retrieve the details of the item.

Note: When searching for an item, if item entitlement has been configured for the customer for whom the quote has been created, the getCompleteItemList API returns only the items that have been configured for that customer.

Solution for Validating Items

In the Quote Details screen, Sterling Field Sales validates whether an item or the quantity of an item is valid for the corresponding quote. The validations are performed only if the item validation rules are appropriately configured in the IBM Sterling Business Center application.

Based on the rules configured in the Sterling Business Center application, the following criteria are used for validating the items in the quote:

- Minimum order quantity and maximum order quantity: The quantity ordered for the item must not be less than the minimum order quantity and not more than the maximum order quantity that is configured for the item.
- Item can be sold separately: The item must be configured in such a way that it can be sold separately.
- Item entitlement: The customer to whom the order belongs must have entitlements to the item.
- Item status: The status of the item must be Published.
- Item's effectivity date: The effective start date of the item must not be later than the current date and the effective end date of the item must not be earlier than the current date.

The validateItemForOrdering API is called to validate the items.

End-User Impact

None.

Implementation

This section explains the configurations for this functionality:

- You must configure the following item validation rules to enable Sterling Field Sales to validate items before they are added to a quote:

– Enable Extended Item Validation

For more information about defining installation rules, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

– Validate Item

For more information about configuring a document's order validation, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

– Validate Item During Order Creation/Modification

For more information about defining a process type's primary information, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

- You must configure the customer entitlements appropriately to enable a user to add an item to a quote created for a particular customer. For more information about creating a customer entitlement rule, refer to the *Business Center: Item Administration Guide*.

- You must configure the quote modification rules appropriately to allow the addition of items to a quote in a particular status. For more information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 22. Performing Price Adjustments for a Quote

Based on the negotiations with a customer, a Field Sales Representative may have to make price adjustments in a quote. Additionally, when performing price adjustments in a quote, a Field Sales Representative may require advice on determining the correct adjustment amount to be applied to a line item. For example, if the discount offered on a particular item is much less than what was provided in quotes on earlier occasions, the Field Sales Representative may want to increase the discount offered on the item in the quote.

Solution

Sterling Field Sales enables a user to perform line-level and header-level price adjustments for a quote. When performing adjustments, a user can view the customer's rating and the metrics used to calculate the rating. Additionally, to arrive at an appropriate price for a product line in a quote, a user can refer to the Discount Advisor.

When adjusting the price for a quote, a user can perform the following tasks:
* Viewing Customer Rating
* Viewing Discount Advisor Report
* Adjusting Price

Solution for Viewing Customer Rating

When performing price adjustments for a quote, a user can view the rating of the customer for which the quote has been created. A user can refer to the customer's rating to arrive at an appropriate price for the items in a quote. The getCustomerDetails API is called to retrieve the customer's rating based on whether the YSCGetAdditionalCustomerInformationUE user exit is implemented or not.

If an image exists with the same name as the value of the `GradeCode` attribute in the output XML of the getCustomerDetails API, the image is used to display the customer's rating. The image is displayed only if a CSS entry has been defined in the sfs-cust-grade-<Grade_Code> format. If an image does not exist, the description of the customer grade is displayed.

Additionally, a user can view the metrics used to calculate the rating if you enable the **Get Customer Grade Information from Sterling Business Intelligence** rule in the Channel Applications Manager. Sterling Field Sales provides a common Java Server Page (JSP) that interacts with the Sterling Business Intelligence data mart to retrieve embedded reports. You must pass the name of the report to be executed as a request parameter to the JSP.

The following table describes the attributes that must be passed as request parameters to the `getfusionchart.jsp` file:

Field	Description
chartHeight	The height of the FusionCharts chart.
chartWidth	The width of the FusionCharts chart.

Field	Description
Customer_Org_Code	The Organization Code of the customer for which the quote was created.
Document_Type_Code	The document type of the quote.
CustomerId	The identifier of the customer for which the quote was created.
ReportName	The name of the report that is to be retrieved from the Sterling Business Intelligence data mart. For example, /Embedded Analytics/Customer Rating Report.
ServiceID	The service that is invoked to convert the report retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph. For example, SAA_getCustomerRatingGraph_9.1.

The getfusionchart.jsp file invokes the SAA_getCustomerRatingGraph_9.1 service, and this service is used to convert the report that is retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph as illustrated in the following figure:

The SAA_getCustomerRatingGraph_9.1 service is invoked for each of the metrics that is used to calculate the rating. This service uses the SCUICognosXMLStandardizer Java class and SCUIFusionChartCreator Java class to convert the report from the data mart to a FusionCharts graph as described below:

1. The SCUICognosXMLStandardizer Java class, a custom API, converts the report from the data mart to a standard XML file.

2. An XSLT is invoked to convert the data in the standard XML file to the FusionCharts format. This XSLT maps the fields in the standard XML file to the attributes in the FusionCharts XML file.

3. The SCUIFusionChartCreator Java class, another custom API, is invoked to append the standards to the standard XML file based on the Chart Type and create the FusionCharts graph.

Solution for Viewing Discount Advisor Report

Sterling Field Sales enables a user to refer to the Discount Advisor when adjusting the price of a line item to arrive at an appropriate price for that item. The Discount Advisor provides a graphical representation of the minimum, maximum, and average adjustments provided to the item in the past. These adjustments are displayed based on metrics such as the industry that the customer belongs to or the category that the item belongs to. For example, if a user wants to view the adjustments provided to a 21" LCD television, the user can select the **Category** tab and view the minimum, maximum, and average adjustments provided to items in the same category as the category of the selected item.

Note: A user can use the Discount Advisor only for a single line in a quote.

Sterling Field Sales provides a common JSP that interacts with the Sterling Business Intelligence data mart to retrieve embedded reports. You must pass the name of the report to be executed and the service that will be used to create a FusionCharts graph from the report as request parameters to the JSP. For example, you must pass the `ReportName` attribute as `/Embedded Analytics/Quote Advisor Report By Industry` and the `ServiceID` attribute as SAA_getIndustryDiscountAdvisorGraph_9.1 to the `getfusionchart.jsp` file to retrieve the report that is executed when the **Industry** tab is selected. Similarly, you must pass the `ReportName` attribute as `/Embedded Analytics/Quote Advisor Report By Category` and the `ServiceID` attribute as SAA_getCategoryDiscountAdvisorGraph_9.1 to retrieve the report that is executed when the **Category** tab is selected.

The following table describes the attributes that must be passed as request parameters to the `getfusionchart.jsp` file:

Field	Description
EnterpriseCode	The Organization Code of the enterprise for which the quote was created.
Document_Type_Code	The document type of the quote.
CustomerId	The identifier of the customer for which the quote was created.
ItemID	The identifier of the item for which the past adjustments are being viewed.
UOM	The unit of measure of the item for which the past adjustments are being viewed.
ReportName	The name of the report that is to be retrieved from the Sterling Business Intelligence data mart. For example, Embedded Analytics/Quote Advisor Report By Industry, Embedded Analytics/Past Discount By Category, and so on.
ServiceID	The service that is invoked to convert the report retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph. For example, SAA_getIndustryDiscountAdvisorGraph_9.1, SAA_getCategoryDiscountAdvisorGraph_9.1, and so on.

The `getfusionchart.jsp` file invokes the SAA_getIndustryDiscountAdvisorGraph_9.1 service or the SAA_getCategoryDiscountAdvisorGraph_9.1 service, and the service is used to convert the report that is retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph as illustrated in the following figure:

The SAA_getIndustryDiscountAdvisorGraph_9.1 service and the
SAA_getCategoryDiscountAdvisorGraph_9.1 service use the
SCUICognosXMLStandardizer Java class and SCUIFusionChartCreator Java class to
convert the report from the data mart to a FusionCharts graph as described below:

1. The SCUICognosXMLStandardizer Java class, a custom API, converts the report
   from the data mart to a standard XML file.

2. An XSLT is invoked to convert the data in the standard XML file to the
   FusionCharts format. This XSLT maps the fields in the standard XML file to the
   attributes in the FusionCharts XML file.

3. The SCUIFusionChartCreator Java class, another custom API, is invoked to
   append the standards to the standard XML file based on the Chart Type and
   create the FusionCharts graph.

Solution for Adjusting Price

Sterling Field Sales enables a user to perform line-level and header-level price
adjustments for a quote using the manual pricing rules that have been configured
in Sterling Field Sales. A user can also adjust the price of the components of a
bundle item in a quote.

The following types of manual pricing rules can be used to perform adjustments:

* Combination: Manual pricing rules of this type can be used to perform only
  header-level price adjustments.
* Order Total: Manual pricing rules of this type can be used to perform only
  header-level price adjustments.
* Item Quantity: Manual pricing rules of this type can be used to perform both
  line-level and header-level price adjustments.

A user can adjust the price for a single line or multiple lines.

The getApplicablePricingRulesForOrder API is called to retrieve the list of
applicable manual pricing rules that can be used to perform price adjustments. The
getApplicablePricingRulesForOrder API is called to retrieve the details of the
applicable manual pricing rule. The changeOrder API is called to save the changes
to the quote.

End-User Impact

This section explains the end-user impact of this functionality:

* If Sterling Business Intelligence is not installed, a user cannot perform the
  following tasks:
  – View the customer rating report.
  – Use the Discount Advisor to view past adjustments for an item.

Implementation

This section explains the configurations for this functionality:

* You must create the manual pricing rules that you want to use to perform
  line-level and header-level price adjustments for a quote. For more information
  about creating a manual pricing rule, refer to the *Business Center: Pricing
  Administration Guide*.
* You must configure the quote modification rules appropriately to allow users to
  perform price adjustments to a quote in a particular status. For more

information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

- To enable the profit margin to be calculated, you must enable the **Inventory Costing Required** rule and configure the costing method in the Applications Manager. For more information about defining additional inventory rules, refer to the *Sterling Selling and Fulfillment Foundation: Global Inventory Visibility Configuration Guide*.

- You must configure the customer grades that you want to assign to customers using the Channel Applications Manager. For more information about defining customer grades, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

- You must configure the **Get Customer Grade Information from Sterling Business Intelligence** rule appropriately in the Channel Applications Manager to enable Sterling Field Sales to retrieve the metrics used to calculate the customer rating. For more information about configuring customer rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

- If you are implementing a custom logic to calculate the customer's rating that is displayed when a user adjusts the price for a line item, you must implement the YSCGetAdditionalCustomerInformationUE user exit. For more information about the YSCGetAdditionalCustomerInformationUE user exit, refer to the *Sterling Selling and Fulfillment Foundation: Javadocs*.

- You must configure the services appropriately to enable the reports retrieved from the Sterling Business Intelligence data mart to be converted to a FusionCharts graph when a user views the past adjustments for a line item and the details of the customer rating. The services are created in the Service Definition Framework. For more information about defining service definitions, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

- You cannot extend, copy, or modify the embedded reports retrieved from the Sterling Business Intelligence data mart. For more information about the embedded reports retrieved from the Sterling Business Intelligence data mart, refer to the *Sterling Business Intelligence: Implementation Guide*.

- To display the customer's rating as an image, you must define a CSS entry in the `sfs-cust-grade-<Grade_Code>` format in the `sfs-images.css` file located in the `<INSTALL-DIR>/repository/eardata/sfs/war/sfs/resources/default/css` directory. The size of the image should be 16 X 120 pixels for ideal display.

Chapter 23. Moving Items between Quote Lines and Recommended Lines

A quote can contain both quote line items and recommended line items. Recommended items are items that a Field Sales Representative recommends to a customer in addition to the list of items a customer has requested. Recommended items are not taken into consideration when the quote's profit margin and the quote total are calculated.

Solution

A user can move a quote line item as a recommended line item. A user can also move a recommended line item as a quote line item. For example, a customer interested in purchasing an LCD television may decide to purchase a television stand that has been included as a recommended line item in the quote. In such a scenario, a user can move the television stand as a quote line item. You can configure the line type to be used for recommended items. The changeOrder API is called to save the changes after the item is moved as a quote line or a recommended line.

End-User Impact

None.

Implementation

You can configure the line type to be used for recommended items. For more information about configuring the **Quote Line Type To Use For Recommended Items** rule, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 24. Modifying Item Quantity in a Quote

Sometimes a Field Sales Representative may have to modify the quantity of the items in a quote. For example, a customer, who had earlier planned to purchase only 10 desktop computers, may have had a change of mind and may want to purchase 15. In such a scenario, a Field Sales Representative may want to change the item quantity as requested by the customer.

Solution

A user can modify the quantity of the items in a quote. A user can modify the quantity of the quote line items and the recommended line items. The changeOrder API is called to save the changes made to the order.

End-User Impact

None.

Implementation

You must configure the quote modification rules appropriately to allow a user to modify the quantity of items in a quote in a particular status. For more information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 25. Adding Related Items to a Quote

A related item is an item that is associated with another item by way of a predefined relationship. For example, a 29" flat screen television can have a 32" LCD television as a related item, with the LCD television defined as an up-sell item for the flat screen television. Sometimes a Field Sales Representative may want to add a related item as a separate line item to a quote.

Solution

Sterling Field Sales enables a user to view the related items of a line item and add them as separate line items to a quote. A user can add a related item either as a quote line item or a recommended line item. You can configure the related items that should be displayed in the Related Products dialog box, based on the association type of the related items. The getAssociationRelationshipList API is called to verify which association types have the **Show In UI** rule enabled. The getCompleteItemList API is called to retrieve the related items to be displayed in the Related Products dialog box.

Note: Sterling Field Sales supports only up-sell and cross-sell association types.

You can define the line relationship types to be used when linking two related lines together. The relationship types defined should support sorting.

After the associations and line relationship types are defined, you can specify the line relationship type to be used when related product items are added to a quote. The getAssociationRelationshipList API is called to retrieve the line relationship type to be used for a related item.

Solution for Validating Items

In the Quote Details screen, when a user adds a related item to a quote, Sterling Field Sales validates whether the item or the quantity of that item is valid for the quote. The validations are performed only if the item validation rules are appropriately configured in IBM Sterling Business Center.

Based on the rules configured in the Sterling Business Center application, the following criteria are used for validating the items in the quote:
* Minimum order quantity and maximum order quantity: The quantity ordered for the item must not be less than the minimum order quantity and not more than the maximum order quantity that is configured for the item.
* Item can be sold separately: The item must be configured in such a way that it can be sold separately.
* Item entitlement: The customer to whom the order belongs must have entitlements to the item.
* Item status: The status of the item must be Published.
* Item's effectivity date: The effective start date of the item must not be later than the current date and the effective end date of the item must not be earlier than the current date.

The validateItemForOrdering API is called to validate the items.

End-User Impact

In the Related Products dialog box, the user will see only the related items of association types for which the **Show In UI** rule is enabled.

Implementation

This section explains the configurations for this functionality:

- You must configure the following item validation rules to enable Sterling Field Sales to validate the related items before they are added to a quote:
  - Enable Extended Item Validation

    For more information about defining installation rules, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
  - Validate Item

    For more information about configuring a document's order validation, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.
  - Validate Item During Order Creation/Modification

    For more information about defining a process type's primary information, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
- You can configure the related items that should be displayed in the Related Products dialog box based on the association type of the related items. For more information about configuring item association rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- You can configure the relationship types to be used when linking two related lines together. Ensure that the **Consider For Sorting** check box is selected when defining the line relationship types. For more information about configuring line relationship types, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- You can configure the relationship type to be associated with a particular association type. For more information about configuring relationship types, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- You must configure the quote modification rules appropriately to allow the addition of items to a quote in a particular status. For more information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 26. Viewing the Components of a Bundle Item in a Quote

Sometimes, a Field Sales Representative may want to view certain information about the components of a bundle item in a quote. For example, if a quote contains a dining room set as a line item and a table as a component of this line item, the Field Sales Representative may want to view the unit price of the table.

Solution

Sterling Field Sales enables a user to view the components of a bundle item in a quote. A user can also view the quantity, unit price, and line total of the components. The getOrderLineList API is called to retrieve the components of a bundle item displayed in the Component Pricing dialog box.

End-User Impact

None.

Implementation

None.

Chapter 27. Configuring or Reconfiguring an Item in a Quote

A configurable item is an item that can be customized, depending on the requirements of a customer. For example, a desktop computer may be available with various options, such as processors with 400 Mhz, 500 Mhz, or 600 MHz speed, and 256 MB or 512 MB memory.

A preconfigured item is a configurable item that has one or more options already configured for it. For example, a desktop computer may be preconfigured with a 500 MHz processor and 256 MB memory.

A Field Sales Representative may sometimes have to configure or reconfigure an item in a quote based on a customer's requirements. For example, if a customer requires a desktop computer with a 400 MHz processor and 256 MB memory, the Field Sales Representative may have to configure the desktop computer accordingly.

Solution

Sterling Field Sales enables a user to configure or reconfigure an item in a quote by selecting the options available for that item. The configurator/configure.action action is invoked to display the product configuration screen. The changeOrder API is called to save the configurations.

End-User Impact

None.

Implementation

None.

Chapter 28. Filtering the Items in a Quote

When viewing the list of items in a quote, a Field Sales Representative may want to narrow down the list of items displayed, based on certain criteria. For example, a Field Sales Representative may want to view only those items that are pending approval so that they can be taken up on priority.

Solution

A user can search for items in a quote by entering the corresponding Item ID or the short description of the item as the search criteria. Additionally, the user can filter the list of items that are displayed based on the following criteria:

- Search for all the lines
- Search for only the lines that are pending approval

The getOrderLineList API is called to retrieve the items that match the filter criteria.

End-User Impact

None.

Implementation

None.

Chapter 29. Deleting an Item from a Quote

A Field Sales Representative may sometimes have to delete items from a quote. For example, a Field Sales Representative may want to delete an item from a quote because the customer is no longer interested in purchasing that item.

Solution

A user can delete an item from a quote. When a user deletes an item, which has related items, from a quote, the related items are automatically deleted. The changeOrder API is called to save the changes made to the quote after the item is deleted.

End-User Impact

None.

Implementation

You must configure the quote modification rules appropriately to enable users to delete items from a quote in a particular status. For more information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

Chapter 30. Modifying the Payment Terms and Shipment Terms in a Quote

A Field Sales Representative may have to modify the payment terms and shipment terms in a quote. For example, if a customer wants to change the carrier service that will be used to transport the items after a quote is converted to an order, a Field Sales Representative may have to specify a different carrier service.

Solution

A user can modify the payment terms in a quote. A user can also modify the shipment terms in a quote, such as the carrier service, the level of service, the requested delivery date, and the shipment instructions. The getScacList API is called to retrieve the list of carrier services displayed in the **Carrier Service** drop-down list. The getCommonCodeList API is called to populate the values in the **Payment Term** and the **Level of Service** drop-down lists. The payment terms are stored as common codes with PAYMENT_TERMS as the code type, and the level of service with LEVEL_OF_SERVICE as the code type. This data is cached. The changeOrder API is called to save the changes made to the quote.

End-User Impact

None.

Implementation

This section explains the configurations for this functionality:
- You must configure the quote modification rules appropriately to allow the modification of payment terms and shipments terms in a quote in a particular status. For more information about configuring quote modification rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- Instructions entered in a quote are saved in the quote using the SHP instruction type.

Chapter 31. Adding and Viewing a Note

A note can be used as an information repository to track an opportunity or a quote through its life cycle, and can be used for future references. Sometimes a Field Sales Representative may want to enter notes manually to describe the actions performed on an opportunity or a quote.

Solution

Notes can be either automatically generated by the system or manually entered by a user.

Sterling Field Sales enables users to enter notes when performing certain tasks. For example, when a user abandons a quote, the user can enter notes to explain why the quote has been abandoned. Notes are automatically generated by the system when important events occur during the life cycle of an opportunity or a quote. Additionally, a user can track the important events and changes that have occurred in an opportunity or a quote by viewing the notes logged earlier.

A user can view and add notes to a quote both at the line level and header level.

Sterling Field Sales provides the following features:
- A Notes dialog box, where a user can view and add notes to an opportunity or a quote.
- A Notes panel, where a user can view and add notes.
- A mechanism to automatically log notes when important events occur during the life cycle of an opportunity or a quote.

Sterling Field Salesprovides a Notes dialog box, which can be launched by clicking the **View/Add Notes** hyperlink in the opportunity-related screens and quote-related screens.

Users can filter the notes that are displayed in the Notes dialog box based on the following criteria:
- Note Type
- System-Generated or Manually Entered

The following table lists the APIs that are called when viewing and adding notes to an opportunity or a quote:

API	Purpose
getCommonCodeList	To retrieve the note types displayed in the **Note Type** drop-down list.
getOpportunityDetails	To retrieve the notes that match the filter criteria.
getCompleteOrderDetails	To retrieve the quote's header-level notes.
getOrderLineList	To retrieve the quote's line-level notes.
manageOpportunity	To save a note added to an opportunity.
changeOrder	To save a note added to a quote.

System-Generated Notes

You can configure the system to log notes automatically when the following events occur:

- Quote Created: The system generates a note when an opportunity is created.
- Quote Sent for Approval: The system generates a note when a quote is sent for approval.
- Quote Presented to Customer: The system generates a note when a quote is presented to a customer.
- Quote Converted to Order: The system generates a note when a quote is converted to an order.

End-User Impact

The notes that have been configured to be logged automatically will be displayed to users in the Notes dialog box and the Notes panel in the opportunity-related screens and quote-related screens.

Implementation

This section explains the configurations for this functionality:

- Sterling Field Sales enables you to define note types and configure automatic note logging. For more information about defining quote note types and configuring automatic note logging, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- Sterling Field Sales provides sample XSL files to generate the text of the notes being logged for each event. The sample XSL files are located in the `resources.jar/template/service/saa` directory. The following table lists the events and the sample XSL files used to log notes for the corresponding events:

Event	Sample XSL File
Quote Created	`SAA_Log_Note_Create_Quote.xsl.sample`
Quote Sent for Approval	`SAA_Log_Note_Send_For_Approval.xsl.sample`
Quote Presented to Customer	`SAA_Log_Note_Send_To_Customer.xsl.sample`
Quote Converted to Order	`SAA_Log_Note_Convert_Quote_To_Order.xsl.sample`

Note: You can use the sample XSL files listed in the table or you can use your own XSL files while selecting the XSL transformation file for an event in the Quote Audit Note Wizard. If you use your own XSL files, you must perform either of the following tasks:

- Save the XSL files in the `<INSTALL_DIR>/extensions/template/service/saa` directory and then re-create the EAR.
- Save the XSL files in the `resources.jar/template/service/saa` directory and then restart your server.

  Note: You can use the second option only if you are using the SAA_LogNotesForQuoteAudit_9.0 service that is provided out of the box.

- Sterling Field Sales logs notes with a unique note type for certain tasks performed by a user. The note types used for saving the notes are listed in the following table:

Task	Note Type
Create Opportunity	SAA_OPPORTUNITY_CREATED
Mark as Lost	SAA_MARK_AS_LOST
Create Quote	SAA_QUOTE_CREATED
Send for Approval	SAA_SENT_FOR_APPROVAL
Send to Customer	SAA_SENT_TO_CUSTOMER
Convert to Order	SAA_CONVERTED_TO_ORDER
Abandon Quote	SAA_QUOTE_ABANDONED
Accept for Customer	SAA_ACCEPTED_BY_CUSTOMER
Reject for Customer	SAA_REJECTED_BY_CUSTOMER

Chapter 32. Implementation of Quote Attachments

A Field Sales Representative (FSR) may want to attach files to quotes. For example, the requirements of the customer might have been e-mailed to the FSR as a Microsoft Office Word document, and the FSR may want to attach the document to the quote, while creating the quote.

Solution

Sterling Field Sales allows users to upload attachments to quotes. Users can also delete existing attachments.

The user can use the Manage Attachments related task to add or delete existing attachments. The `getCompleteOrderDetails` API is called to retrieve the details of the quote to be displayed in the Manage Attachments screen.

Users can add attachments, by clicking the **Add** button. The **New Attachment** dialog box is displayed and the `scfileuploadwidget` is used to upload the files. The files are uploaded using the following widgets:

- When the user browses and selects a file the `sc.file.upload.type` widget is set to `ONSELECT`. When a single file is, or multiple files, are selected, a multipart request stores the file temporarily on the server.

- After selecting one or more files to be uploaded, when the user clicks the **Save** button, the `sc.file.upload.type` widget is set to `ONSUBMIT` and the files are saved as attachments for the quote. The `changeOrder` API transfers the files from the temporary location on the server to the database.

- After the files are attached to the quote, the details of the file attachment such as the file name, file description, file content type, and the file size are displayed in the UI. The value of the file type is fetched from the common codes.

- The file attachment API output returns an internet mime type. This internet mime type is mapped with the common codes to generate the value of the file type.

After the files are attached to the quote, the `FileAttachments` element with file information, such as a unique File ID, File location and Widget ID is uploaded on the server. The **New Attachment** dialog box is closed, and the **Attachments** panel of the Manage Attachments screen is refreshed to display the newly added attachment.

When the user clicks the **Delete** button on the Manage Attachments screen, the selected attachments are removed from the quote using the `changeOrder` API.

Note: Multiple attachments can be deleted at the same time.
The user can download previously uploaded attachments. When the user click the link on the **Name** of the attachment, the **File Download** dialog box is displayed prompting the user to open or save the attachment.

Implementation

This section explains the configurations for this feature:

- You can configure the directory where the attachments are temporarily stored. For more information on configuring the directory location, see the *Sterling Field Sales Customization Guide*.

  Note: The temporary file location should not be the same as the application server location.
- You can customize Sterling Field Sales to add files with file extensions, which are not provided, by default. You must override the properties specified in the `customeroverride.properties` file. For more information, see the *Sterling Field Sales: Customization Guide*.
- You can configure the application to encode or decode the files being attached to Sterling Field Sales. A new interface, `IFileEncoderDecoder`, is provided to encode the file content before it is streamed to the temporary file. For more information, see the *Sterling Field Sales: Customization Guide*.
- You can configure the application to enable virus scan on the file being attached to Sterling Field Sales. A hook is provided using which Sterling Field Sales can plug- in the logic to scan the file being uploaded for viruses. The `sc.file.upload.virus.scanner` property should be provided in `yfs.properties` file. `IVirusScanner` is the newly added interface which enables virus scans. For more information on enabling virus scan, see the *Sterling Field Sales: Customization Guide*.
- You can configure the secure delete feature that is available in Sterling Field Sales, by default. A new interface, `ISecureFileDelete` is provided that performs a secure delete of the files placed in the temporary location. For more information on customizing the secure delete feature, see the *Sterling Field Sales: Customization Guide*.

Chapter 33. Validating a Quote

A quote may contain items that are not valid for the quote. Therefore, it is necessary that a quote is validated to ensure that all the items or the item quantity in the quote is valid for that quote. For example, if the item quantity in a quote is more than the maximum quantity that can be ordered for that particular item, the item quantity must be modified before a Field Sales Representative can present the quote to a customer.

Solution

Sterling Field Sales enables a user to view the quote lines and recommended lines in a quote that have failed validation in a quote. A user can also view the reasons why the lines failed validation. The validateItemsForOrdering API is called to retrieve the quote lines and recommended lines failing validation. The getCompleteOrderDetails API is called to retrieve the quote header information.

Sterling Field Sales verifies whether the items or the item quantity in a quote is valid for the quote when a user performs any of the following tasks:

* Requests approval for the quote
* Presents the quote to a customer
* Converts the quote to an order

End-User Impact

None.

Implementation

None.

Chapter 34. Implementation of Quote Approval

Sometimes a quote may be placed on hold for various reasons. In such a scenario, the quote must be approved by all the approvers defined in the approval plan to enable a Field Sales Representative to present the quote to a customer. For example, if the adjustments applied to a quote exceed the maximum limit that has been configured, a Field Sales Representative can present the quote to a customer only after it has been approved by all the required approvers.

Solution

In IBM Sterling Field Sales, a quote requires approval if any or all of the following conditions are met:

- Any of the items in the quote violates an approval rule.
- The adjustments applied to a quote or any of the items in a quote under a manual pricing rule exceed the maximum discount limit configured under the rule.

A quote must be approved by all the required approvers defined in the approval plan before it can be presented to a customer. An approver can also assign a delegate user, who can approve or reject quotes for a specified period of time.

The quote approval process involves the following steps:

1. Setting a Delegate User
2. Requesting Approval
3. Viewing the Approval Summary
4. Accepting or Rejecting an Approval Request

Solution for Setting a Delegate User

From time-to-time, a user with permissions to approve or reject a quote, may be unavailable for reasons such as vacation, out-sick, traveling for extended periods, and so forth. Or, a user may want to assign another user to approve or reject a quote on his or her behalf. Such users can set up a delegate user.

This delegate user can approve or reject quotes using the **Preferences** link. When Preferences dialog box is displayed, the user can search and assign a delegate user. The getUserList API is called to retrieve a list of users when the user performs the user search and the getUserUiStateList API, getUserHierarchy are called to display the delegate user.

Note: The **getUserHierarchy** API does not return any record if there is no existing delegate for the quote.

Solution for Requesting Approval

IBM Sterling Field Sales enables a user to request approval for a quote.

The following table describes the APIs that are called when a user requests approval for a quote:

API   Purpose

validateItemsForOrdering
>To validate whether the items or the item quantity in the quote is valid for the quote.

getCompleteOrderDetails
>To retrieve the quote header information.

sendOrderForApproval
>To place a request for approval of the quote.

IBM Sterling Field Sales provides the capability to apply a hold to a quote that is pending approval. The hold type that is applied to the quote is determined by the **Hold Type To Be Applied When Order Needs Approval** rule.

Solution for Viewing the Approval Summary

A user can view the summary of all the approvals that have been obtained or are required for a quote. The getCompleteOrderDetails API is called to display the Approvals screen. In the Approvals screen, along with the quote's header-level information, you can view the list of approvers who are involved in the approval process and the status of the approval for each approver, the pricing summary, and notes that have been added to the quote.

A user can view the list of header-level and line-level rule violations that require approval in the Violations That Require Approval panel. Additionally, for line-level rule violations, a user can view the details of the lines that have violated the corresponding rule in the Violations For Approver dialog box. The getCompleteOrderDetails API is called to retrieve the details of the lines that have violated the rule.

When viewing the approval summary of a quote, a user can view the rating of the customer for which the quote has been created. The getCustomerDetails API is called to retrieve the customer's rating based on whether the YSCGetAdditionalCustomerInformationUE user exit is implemented or not.

If an image exists with the same name as the value of the `GradeCode` attribute in the output XML of the getCustomerDetails API, the image is used to display the customer's rating. The image is displayed only if a CSS entry has been defined in the `sfs-cust-grade-<Grade_Code>` format. If an image does not exist, the description of the customer grade is displayed.

Additionally, a user can view the metrics used to calculate the rating if you enable the **Get Customer Grade Information from Sterling Business Intelligence** rule in the Applications Manager. IBM Sterling Field Sales provides a common Java Server Page (JSP) that interacts with the Sterling Business Intelligence data mart to retrieve embedded reports. You must pass the name of the report to be executed as a request parameter to the JSP.

The following table describes the attributes that must be passed as request parameters to the `getfusionchart.jsp` file:

Field   Description

chartHeight
>The height of the FusionCharts chart.

chartWidth
>The width of the FusionCharts chart.

Customer_Org_Code
> The Organization Code of the customer for which the quote was created.

Document_Type_Code
> The document type of the quote.

CustomerId
> The identifier of the customer for which the quote was created.

ReportName
> The name of the report that is to be retrieved from the Sterling Business Intelligence data mart. For example, /Embedded Analytics/Customer Rating Report.

ServiceID
> The service that is invoked to convert the report retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph. For example, SAA_getCustomerRatingGraph_9.1.

The `getfusionchart.jsp` file invokes the SAA_getCustomerRatingGraph_9.1 service, and this service is used to convert the report that is retrieved from the Sterling Business Intelligence data mart to a FusionCharts graph as illustrated in the following figure:

The SAA_getCustomerRatingGraph_9.1 service is invoked for each of the metrics that is used to calculate the rating. This service uses the SCUICognosXMLStandardizer Java class and SCUIFusionChartCreator Java class to convert the report from the data mart to a FusionCharts graph as described below:

1. The SCUICognosXMLStandardizer Java class, a custom API, converts the report from the data mart to a standard XML file.
2. An XSLT is invoked to convert the data in the standard XML file to the FusionCharts format. This XSLT maps the fields in the standard XML file to the attributes in the FusionCharts XML file.
3. The SCUIFusionChartCreator Java class, another custom API, is invoked to append the standards to the standard XML file based on the Chart Type and create the FusionCharts graph.

Solution for Accepting or Rejecting an Approval Request

A user can either accept or reject the request for approval of a quote. The recordOrderApproval API is called to save the changes to a quote after the request for approval is accepted or rejected.

End-User Impact

If Sterling Business Intelligence is not installed, a user cannot view the customer rating report.

Implementation

This section explains the configurations for this functionality:

- You must configure the approval rules appropriately in the IBM Sterling Business Center application to determine the conditions under which approvals are required for quotes. For more information about creating an approval rule, refer to the *Business Center: Pricing Administration Guide*.
- You must configure approval plans in the IBM Sterling Business Center to determine who must approve a quote, and the sequence, if any, in which the approvals must occur. For more information about how to configure approval plans, refer to the *Business Center: Item Administration Guide*.
- To enable IBM Sterling Field Sales to apply a hold to a quote that is pending approval, you must enable the **Hold Type To Be Applied When Order Needs Approval** rule in the Applications Manager. For more information about this rule, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.
- You must configure the customer grades that you want to assign to customers using the Channel Applications Manager. For more information about defining customer grades, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- You must configure the **Get Customer Grade Information from Sterling Business Intelligence** rule appropriately in the Channel Applications Manager to enable IBM Sterling Field Sales to retrieve the metrics used to calculate the customer rating. For more information about configuring customer rules, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- If you are implementing a custom logic to calculate the customer's rating that is displayed when a user adjusts the price for a line item, you must implement the YSCGetAdditionalCustomerInformationUE user exit. For more information about the YSCGetAdditionalCustomerInformationUE user exit, refer to the *Sterling Selling and Fulfillment Foundation: Javadocs*.
- You must configure the services appropriately to enable the reports retrieved from the Sterling Business Intelligence data mart to be converted to a FusionCharts graph when a user views the details of the customer rating. The services are created in the Service Definition Framework. For more information about defining service definitions, refer to the *Sterling Selling and Fulfillment Foundation: Configuration Guide*.
- You cannot extend, copy, or modify the embedded reports retrieved from the Sterling Business Intelligence data mart. For more information about the embedded reports retrieved from the Sterling Business Intelligence data mart, refer to the *Sterling Business Intelligence: Implementation Guide*.
- To display the customer's rating as an image, you must define a CSS entry in the `sfs-cust-grade-<Grade_Code>` format in the `sfs-images.css` file located in the `<INSTALL-DIR>/repository/eardata/sfs/war/sfs/resources/default/css` directory. The size of the image should be 16 X 120 pixels for ideal display.
- The notes entered when a user sends a quote for approval are saved in the quote using the SAA_SENT_FOR_APPROVALS note type.

Chapter 35. Implementation of Generating a Proposal

Sometimes, a Field Sales Representative may want to print a proposal for a quote so that it can be sent to a customer. For example, a customer may request a Field Sales Representative to e-mail the quote so that the customer can view the prices of the items in the quote.

Solution

Sterling Field Sales provides the **Generate Proposal** related task to enable a user to generate a proposal for a quote. A user can print the proposal as a PDF by selecting the PDF format from the **Choose a print format** drop-down list, which is generated using the SAA_GeneratePrint service.

The getCompleteOrderDetails API is called to retrieve the details of the quote to be displayed in the Generate Proposal screen. The getTransformationData API is called to retrieve the values to be displayed in the **Choose a formatting template** drop-down list. The getCommonCodeList API is called to retrieve the values to be displayed in the **Choose a print format** drop-down list. The print formats are stored as common codes with code type SAA_QUOTE_PRNT_FRMT. Based on the print template selected, the locale specific formats configured for the selected template are displayed under the **Choose a Locale** drop-down list.

A user can select the attributes that should be printed in the proposal from the Include In Print panel. The attributes displayed under the Include In Print panel are defined in the XML format in Sterling Business Center, that is stored in the database and is retrieved when the getTransformationConfigList API is called.

Note: To view the attributes in the Include In Print panel in Sterling Field Sales, the name of the Print Template created in Sterling Business Center has to be **PrintProposal**.
Following is a sample metadata XML file:

```
<Resource>
<Group Name="ProposalHeader" DisplayName="Quote_Proposal_Header" Id="header">
 <Attribute Name="ProposalDate" DisplayName="Quote_Proposal_Date" Value="Y"/>
 <Attribute Name="Status" DisplayName="Quote_Proposal_Quote_Status" Value="Y"/>
 <Attribute Name="ExpirationDate" DisplayName="Quote_Proposal_Expiration_Date" Value="Y"/>
 <Attribute Name="ProposalTitle" DisplayName="Quote_Proposal_Title" Value="Y"/>
 <Attribute Name="ProposalID" DisplayName="Quote_Proposal_ID" Value="Y"/>
</Group>

<Group Name="Address" DisplayName="Quote_Proposal_Address">
 <Attribute Name="BillTo" DisplayName="Quote_Proposal_Bill_Address" Value="Y"/>
 <Attribute Name="ShipTo" DisplayName="Quote_Proposal_Shipping_Address" Value="Y"/>
</Group>

<Group Name="OrderLines" DisplayName="Quote_Proposal_Lines">
 <Attribute Name="QuoteLines" DisplayName="Quote_Proposal_Quote_Lines" Value="Y"/>
 <Attribute Name="RecommendedLines" DisplayName="Quote_Proposal_Recommended_Lines" Value="Y"/>
</Group>

<Group Name="PricingInformation" DisplayName="Quote_Proposal_Quote_Totals">
 <Attribute Name="SubTotal" DisplayName="Quote_Proposal_Line_Sub_Total" Value="Y"/>
 <Attribute Name="Discounts" DisplayName="Quote_Proposal_Adjustments" Value="Y"/>
 <Attribute Name="ShippingCharges" DisplayName="Quote_Proposal_Shipping" Value="Y"/>
 <Attribute Name="Taxes" DisplayName="Quote_Proposal_Taxes" Value="Y"/>
 <Attribute Name="GrandTotal" DisplayName="Quote_Proposal_Total" Value="Y"/>
```

```
    </Group>

  <Group Name="Terms" DisplayName="Quote_Proposal_Terms">
   <Attribute Name="PaymentTerms" DisplayName="Quote_Proposal_Payment_Terms" Value="Y"/>
   <Attribute Name="ShippngTerms" DisplayName="Quote_Proposal_Carrier" Value="Y"/>
   <Attribute Name="LevelOfService" DisplayName="Quote_Proposal_Level_Service" Value="Y"/>
   <Attribute Name="DeliveryDate" DisplayName="Quote_Proposal_Devievry_Date" Value="Y"/>
   <Attribute Name="InstructionText" DisplayName="Quote_Proposal_Shipping_Instructions" Value="Y"/>
  </Group>
</Resource>
```

Note: The attribute value of the **DisplayName** is read from the server side bundle literals and is displayed in the UI. If a new attribute is added, the **DisplayName** attribute's value should be added to the server side bundle file.

The getOrganizationHierarchy API invokes the SAA_GeneratePrint service to get the address details of the organization. The getUserHierarchy API gets the name of the sales person to be displayed in the proposal.

When a user clicks **Generate** in the Generate Proposal screen, the SAA_GeneratePrint service is invoked to generate the proposal.

A sample implementation of the service that is used to generate a proposal is illustrated in the following figure:

Based on the selected template and locale, the processPrint API retrieves the configured format and along with the XML, generates the proposal in the selected format (PDF) using the Apache FOP (Formatting Objects Processor) component.

Implementation

This section explains the configurations for this functionality:

Sterling Field Sales ships a config file to the FOP component. By default, the config file shipped to the FOP component, searches for all the fonts installed on the system. The FOP component of the **SAA_GeneratePrint** service takes **FOPConfigFile** argument, which points to the location of the config file in the classpath.

You can customize the config file to add or remove the fonts based on the requirement . If the config file is customized, the **FOPConfigFile** argument should be changed to point to the customized config file. This config file is used by FOP processor to generate the PDF.

Note:
• To generate the proposal in various languages, it is necessary that the required language fonts are installed on the system. Also, ensure that the font-family in the corresponding style sheet is changed appropriately to reflect these fonts.

- Including all the system fonts may affect the performance of the FOP proposal generation. Hence it is recommended to customize the configuration file to include only the fonts which are required for the proposal generation.

For more information on the FOP configuration file and fonts configuration, refer to http://xmlgraphics.apache.org/fop/1.0/configuration.html

Note:
- To view the formatting templates in Sterling Field Sales, the name of the Print Template created in Sterling Business Center has to be **PrintProposal**.
- You can configure the print template to generate a proposal. For more information about configuring proposal templates, refer to the *Business Center: System Configuration Guide*.
- You can configure the print format to be used to generate a proposal. For more information about configuring proposal formats, refer to the *Business Center: System Configuration Guide*.

Install the FOP Jars

To install the FOP Jars, perform the following steps:
1. Ensure that the following FOP jars are shipped with Foundation and is available in **<INSTALL_DIR>/jar\smcfs\<PRODUCT_VERSION>**
   - avalon-framework-4.2.0.jar
   - fop.jar
   - xmlgraphics-commons-1.4.jar
2. Install the FOP jars by running the appropriate command from the **<INSTALL_DIR>/bin** folder.
   a. Run the following command to install the jar file on Windows: **install3rdParty.cmd**
   b. Run the following command to install the jar file on Unix or Linux: **install3rdParty.sh**

   For more information about installing the third-party jar files, refer the *Sterling Selling and Fulfillment Foundation: Installation Guide*.

Chapter 36. Presenting a Quote to a Customer

After all the necessary approvals have been obtained for a quote, a Field Sales Representative must present the quote to the customer for whom the quote has been created so that the customer can provide feedback on the quote.

Solution

Sterling Field Sales enables a user to present a quote to a customer. If a quote requires approval, a user can present the quote to a customer only if it has been approved by all the approvers.

The following table describes the APIs that are called when a user presents a quote to a customer:

API	Purpose
validateItemsForOrdering	To validate whether the items or the item quantity in the quote is valid for the quote.
getCompleteOrderDetails	To retrieve the quote header information.
presentToBuyer	To change the status of the quote to Presented after the user presents the quote to the customer.

End-User Impact

None.

Implementation

The notes entered when a user presents a quote to a customer are saved in the quote using the SAA_SENT_TO_CUSTOMER note type.

Chapter 37. Accepting a Quote on Behalf of a Customer

After a quote is presented to a customer, the customer may want to accept the quote in its current form so that the quote can be converted to an order.

Solution

Sterling Field Sales provides the **Accept** related task that enables a user to accept a quote on behalf of a customer. The getCompleteOrderDetails API is called to retrieve the details of the quote to be displayed in the Accept For Customer screen after a user clicks the **Accept** related task. The sendBuyerResponse API is called to change the status of the quote to Accepted after the user clicks **Confirm** in the Accept For Customer screen.

End-User Impact

None.

Implementation

The notes entered when a user rejects a quote are saved in the quote using the SAA_ACCEPTED_BY_CUSTOMER note type.

Chapter 38. Rejecting a Quote on Behalf of a Customer

A customer may want to reject a quote because the contents of the quote do not meet the customer's requirements. In such a scenario, a Field Sales Representative may have to reject the quote on behalf of the customer.

Solution

A user can reject a quote in an opportunity on behalf of a customer. A user can reject a quote from the Opportunity Details screen or the Quote Summary screen. The getCompleteOrderDetails API is called to retrieve the details of the quote to be displayed in the Reject For Customer screen after a user clicks the **Reject** related task. The sendBuyerResponse API is called to change the status of the quote to Rejected.

End-User Impact

None.

Implementation

The notes entered when a user rejects a quote are saved in the quote using the SAA_REJECTED_BY_CUSTOMER note type.

Chapter 39. Reworking a Quote

Sometimes a quote may be rejected by a customer in its current form. For example, a customer may reject a quote because the customer is not satisfied with the price of some of the items in the quote. After a quote is rejected by a customer, if a Field Sales Representative wants to negotiate with the customer further, the Field Sales Representative must rework the quote and make the necessary modifications so that the quote can be presented to the customer again.

Solution

After a quote has been rejected by a customer, a user can rework the quote. The reworkQuote API is called to change the status of the quote to **Created**.

End-User Impact

None.

Implementation

The notes entered when a quote is reworked are saved in the quote using the SAA_QUOTE_REWORKED note type.

Chapter 40. Converting a Quote to an Order

After a customer accepts a quote in its current form, a Field Sales Representative may want to convert the quote to an order.

Solution

Sterling Field Sales enables a user to convert a quote to an order after the quote is accepted by a customer. After a user clicks the **Convert to Order** related task, Sterling Field Sales verifies whether the addresses in the quote are valid or not, as follows:

* If the IsAddressVerified flag is passed as Y for the Bill To address and the Ship To address in the quote, the getCompleteOrderDetails API is called to retrieve the details of the quote to be displayed in the Convert to Order screen.
* If the IsAddressVerified flag is passed as N for the Bill To address and the Ship To address in the quote, the verifyAddress API is called to verify whether the address is a valid address or not.
  – If the address verification succeeds, the addresses are saved to the quote and the getCompleteOrderDetails API is called to retrieve the details of the quote to be displayed in the Convert to Order screen.
  – If the address verification fails, the Quote Details screen is displayed. The user must make the required changes to the addresses that have failed verification to proceed with the task of converting the quote to an order. If the status modification rules are configured to not allow modifications to a quote in the Accepted status, the user must rework the quote to make the necessary changes.

Sterling Field Sales also verifies whether the items or the item quantity in the quote is valid for the quote. The validateItemsForOrdering API is called to validate the items in a quote.

A user can either perform the necessary modifications in the Convert to Order screen or proceed with the conversion without making any modifications. If the user clicks **Confirm** without performing any modifications, the createOrderFromQuote API is called to create an order for the quote based on the attributes defined in the Default From Quote template.

A user can perform the following modifications in the Convert to Order screen:

* Add a payment method
* Modify the amount to be charged
* Delete a payment method
* Add a note

When converting a quote to an order, a user can specify one or more payment methods to be used to pay for the order and the amount to be charged to the payment methods. In the Payment Method dialog box, the user can either select an existing payment method available for the customer or create a new payment method. The user can also select an existing payment method and modify its details to create a new payment method.

The following APIs are called when a user adds a payment method to a quote:

API	Purpose
getCustomerDetails	To retrieve the payment methods that are available for the customer for whom the quote has been created. This API is also called to retrieve the list of additional addresses for a customer that are displayed if the user selects a payment method of the Credit Card payment type.
getPaymentTypeList	To retrieve the list of payment types available for the enterprise for which the quote has been created. This API is called to retrieve the values to be displayed in the **Payment Type** drop-down list.
getCommonCodeList	To retrieve the common codes for credit card types and credit card expiration months. The credit card types are stored as common codes with CREDIT_CARD_TYPE as the code type. The credit card expiration months are stored as common codes with MONTHS as the code type.
createAccessToken	To create the session token that is passed to the *Sterling Sensitive Data Capture Server, Release 1.1* to process the credit card numbers and stored value card numbers. This API is called only when a user adds a payment method of the Credit Card or Store Value Card payment type.

The *Sterling Sensitive Data Capture Server, Release 1.1* is an application that integrates with the IBM Sterling Selling and Fulfillment Foundation to ensure that credit card numbers and stored value card numbers are secure by tokenization. Before the *Sterling Sensitive Data Capture Server, Release 1.1* processes any credit card number or stored value card number, it validates with the Sterling Selling and Fulfillment Foundation whether the session token that has been passed to it is valid. This enables the Sterling Selling and Fulfillment Foundation to achieve Payment Application Data Security Standard (PA-DSS) compliance.

After adding a payment method to a quote, a user can delete the payment method. Additionally, a user can modify the amount to be charged to a customer's credit card or customer account. A user can also add notes before converting a quote to an order. For more information about adding notes, refer to the topic "Adding and Viewing a Note".

After the user makes the necessary modifications and clicks **Confirm** in the Convert to Order screen, the changeOrder API is called to add the payment information to the quote. The createOrderFromQuote API is called to create an order for the quote by copying the information from the quote to the order based on the attributes defined in the Default From Quote template.

When a quote is converted to an order, all the attributes defined in the Default From Quote template in the Applications Manager are copied from the quote to the order.

Note: If you want the shipping method to be set automatically on an order when it is converted from a quote, it is recommended that you implement a service that sets the shipping method based on the level of service. This service can be invoked after the order is successfully created.

End-User Impact

When a user adds a credit card payment method to a quote, the user can enter the name of the credit card holder in a single box or two boxes based on the configuration.

Implementation

This section explains the configurations for this functionality:

- You can configure the **Use A Single Input Field For When Entering The Name On Card Information On A Credit Card In the User Interface** rule to enable users to enter the name of the credit card holder in a single box or two boxes. For more information about this rule, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.

- The *Sterling Sensitive Data Capture Server, Release 1.1* must be configured to enable a user to add payment methods of the Credit Card and Stored Value Card payment types. For more information, refer to the *Sterling Selling and Fulfillment Foundation: Sterling Sensitive Data Capture Server, Release 1.1: Configuration Guide* and the *Sterling Sensitive Data Capture Server, Release 1.1: PA-DSS Implementation Guide*.

- You can define the attributes that should be copied when converting a quote to an order in the Default From Quote template in the Applications Manager. For more information about defining process type details, refer to the *Sterling Selling and Fulfillment Foundation: Distributed Order Management Configuration Guide*.

- The notes entered when a user converts a quote to an order are saved in the order using the SAA_CONVERTED_TO_ORDER note type.

Chapter 41. Abandoning a Quote

If a quote no longer serves a purpose, a Field Sales Representative may have to abandon that quote. For example, if a Field Sales Representative creates a quote by mistake, the quote may have to be abandoned.

Solution

A user can abandon a quote using the **Abandon Quote** related task. When abandoning a quote, a user must provide reasons for performing that action. You can configure the reason codes in the Channel Applications Manager. The getCommonCodeList API is called to populate the **Reason Code** drop-down list. This data is cached. After a quote is abandoned, the changeOrder API is called to save the changes to the quote.

End-User Impact

None.

Implementation

This section explains the configurations for this functionality:

- You can configure the reason codes to enable users to provide reasons for abandoning a quote. For more information about defining quote abandonment reasons, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- The notes entered when a user abandons a quote are saved in the quote using the SAA_QUOTE_ABANDONED note type.

Chapter 42. Managing Saved Search

A Field Sales Representative may want to save a search for future use to avoid entering the same criteria every time the Field Sales Representative wants to search for a particular entity. For example, a Field Sales Representative, who is currently working on closing all the quotes created for a customer, may have to frequently search for the quotes created for that customer.

Solution

Sterling Field Sales enables a user to manage saved searches for the following entities:

* Opportunity
* Quote

As part of saved search management, a user can perform the following tasks:

* Saving a Search
* Searching for a Saved Search
* Renaming a Saved Search
* Viewing and Executing a Saved Search
* Deleting a Saved Search

Solution for Saving a Search

A user can save a search to avoid entering the search criteria every time the user searches for an entity. The createSavedSearch API is called to save a search after the user clicks **Save** in the Saved Search dialog box.

Solution for Searching for a Saved Search

A user can search for a saved search. The getSavedSearchList API is called to retrieve the saved searches matching the search criteria.

Solution for Renaming a Saved Search

A user can search for a saved search and rename the saved search. After a user modifies the name of the saved search, the modifySavedSearch API is called to save the changes.

Solution for Viewing and Executing a Saved Search

A user can view and execute a saved search. The getSavedSearchList API is called to retrieve the available saved searches that a user can view and execute. The getSavedSearchDetails API is called to display the corresponding search screen for a saved search. Based on the `SearchContext` attribute of the saved search, the corresponding strut action is invoked to execute the saved search.

Solution for Deleting a Saved Search

A user can delete a saved search if the saved search is no longer required. The deleteSavedSearch API is called to delete the saved search.

End-User Impact

None.

Implementation

None.

Chapter 43. Implementation of Input Parameter Validation

Input Parameter Validation in Sterling Field Sales ensures that the malicious scripts do not reach the business layer as the validation layer validates the data and sends it for further processing only on successful validation. The main purpose of this activity is not to validate the individual input fields in the UI, but to safe guard the application as a whole from external attacks like the XSS.

Solution

Input Parameter Validation provides the data validation functionality for validating the input requests. You can define your own validation rules for validating different request parameters. Input validation can be performed for various kinds of inputs such as parameter name, parameter value, cookie name, cookie value, and so on. Sterling Field Sales supports regular expression based validation.

End-User Impact

Validation rules are set for the user ID which is entered to access the Sterling Field Sales application. As a result, you may be restricted from entering a few characters in user ID. Hence, ensure that the user ID is created as per the default regular expression defined in Sterling Field Sales to log in to the application successfully. For more information on default regular expression, refer to the *Sterling Field Sales: Customization Guide*.

Validation rules are set for the data that is entered in the user interface. As a result, you may not be able to enter few characters in the user interface. If the data entered by you does not pass the validation, a descriptive error message is displayed indicating the reason for the validation failure.

Out of the box, for the input validation failure, stack trace is suppressed and only a descriptive message is displayed in the user interface. However, to view the stack trace for input parameter validation failure in the user interface, you must grant the **View Stack Trace** permission to the User Group. For more information on assigning the resource permissions, refer to *Sterling Selling and Fulfillment Foundation: Configuration Guide*.

Implementation

This section explains the configurations for this functionality:
- You can define the **Input Parameter Validation Rules** to prevent the malicious scripts from reaching the business layer. For know more on validation rules,

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Index

IBM ®

Printed in USA