

Sterling Field Sales

Implementation Guide

9.1.0.2



Copyright

This edition applies to the 9.1 Version of IBM® Sterling Field Sales and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices* on page 72.

Licensed Materials - Property of IBM

IBM® Sterling Field Sales

© Copyright IBM Corp. 2010, 2011. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP

Schedule Contract with IBM Corp.

Contents

Introduction to IBM Sterling Field Sales	7
Initial Configurations to be Performed in Channel Applications Manager	8
Opportunity Life Cycle	9
Quote Life Cycle	10
Setting User Preferences	11
Solution	11
End-User Impact	11
Implementation	11
Managing an Opportunity	12
Creating an Opportunity	13
Solution	13
End-User Impact	13
Implementation	14
Viewing Opportunity Details	15
Solution	15
End-User Impact	16
Implementation	16
Modifying Opportunity Details	17
Solution	17
End-User Impact	17
Implementation	17
Modifying Customer Details	18
Solution	18
Address Verification User Exit	18
End-User Impact	19
Implementation	19
Marking an Opportunity as Lost	20
Solution	20
End-User Impact	20
Implementation	20
Creating an Alternative Quote	21
Solution	21
End-User Impact	21
Implementation	21
Creating a Quote for an Opportunity	22
Solution	22
End-User Impact	22
Implementation	22
Copying a Quote to Create a New Quote and New Opportunity for a Customer	23

Solution	23
End-User Impact	23
Implementation	23
Browsing Catalog	24
Solution	24
Pagination	25
Verifying Item Availability	26
End-User Impact	26
Implementation	26
Managing a Quote	27
Managing Customer Details	27
Managing Item Details	27
Managing Payment Terms and Shipment Terms	28
Creating a Quote	29
Solution	29
End-User Impact	30
Implementation	30
Viewing a Quote Summary	31
Solution	31
End-User Impact	32
Implementation	32
Marking a Quote as the Primary Quote	33
Solution	33
End-User Impact	33
Implementation	33
Modifying Quote Name and Expiration Date	34
Solution	34
End-User Impact	34
Implementation	34
Adding Items to a Quote	35
Solution	35
Validating Items	35
End-User Impact	35
Implementation	35
Performing Price Adjustments for a Quote	37
Solution	37
Viewing Customer Rating	37
Viewing Discount Advisor Report	38
Adjusting Price	40
End-User Impact	40
Implementation	40
Moving Items Between Quote Lines and Recommended Lines	42
Solution	42
End-User Impact	42
Implementation	42
Modifying Item Quantity in a Quote	43
Solution	43
End-User Impact	43
Implementation	43
Adding Related Items to a Quote	44
Solution	44

Validating Items	44
End-User Impact	45
Implementation	45
Viewing the Components of a Bundle Item in a Quote	46
Solution	46
End-User Impact	46
Implementation	46
Configuring or Reconfiguring an Item in a Quote	47
Solution	47
End-User Impact	47
Implementation	47
Filtering the Items in a Quote	48
Solution	48
End-User Impact	48
Implementation	48
Deleting an Item from a Quote	49
Solution	49
End-User Impact	49
Implementation	49
Modifying the Payment Terms and Shipment Terms in a Quote	50
Solution	50
End-User Impact	50
Implementation	50
Adding and Viewing a Note	51
Solution	51
System-Generated Notes	52
End-User Impact	52
Implementation	52
Validating a Quote	54
Solution	54
End-User Impact	54
Implementation	54
Quote Approval	55
Solution	55
Requesting Approval	55
Viewing the Approval Summary	55
Accepting or Rejecting an Approval Request	57
End-User Impact	57
Implementation	57
Generating a Proposal	59
Solution	59
End-User Impact	61
Implementation	61
Presenting a Quote to a Customer	62
Solution	62
End-User Impact	62
Implementation	62
Accepting a Quote on Behalf of a Customer	63
Solution	63
End-User Impact	63
Implementation	63

Rejecting a Quote on Behalf of a Customer	64
Solution	64
End-User Impact	64
Implementation	64
Reworking a Quote	65
Solution	65
End-User Impact	65
Implementation	65
Converting a Quote to an Order	66
Solution	66
End-User Impact	67
Implementation	68
Abandoning a Quote	69
Solution	69
End-User Impact	69
Implementation	69
Managing Saved Search	70
Solution	70
Saving a Search	70
Searching for a Saved Search	70
Renaming a Saved Search	70
Viewing and Executing a Saved Search	70
Deleting a Saved Search	70
End-User Impact	71
Implementation	71
Notices	72
Trademarks	74
Index	76

Introduction to IBM Sterling Field Sales

IBM® 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*.

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*.

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 Sterling Field Sales. For more information about this pipeline, refer to *Sterling Selling and Fulfillment Foundation: Setting Up Quotes*.

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 Sterling Field Sales. For more information about this pipeline, refer to *Sterling Selling and Fulfillment Foundation: Setting Up Quotes*.

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.

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](#)

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*.

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*.

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*.

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.

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.

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.

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 Distributed Order Management: Configuration Guide*.

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.

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 Distributed Order Management: Configuration Guide*.

Browsing Catalog

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 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 *Sterling 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.

Pagination

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.

Verifying Item Availability

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 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: Application Platform 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 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 *Sterling 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*.

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.

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
<code>getOrderLineList</code>	To retrieve the details of the line items in the quote.

API	Purpose
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](#)".

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*.

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
<code>getCompleteOrderDetails</code>	To retrieve the details of the quote.

API	Purpose
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.
getScacList API	To retrieve the details of the carrier service to be used for the quote.

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.

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.

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*.

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.

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 Business Center application.

Based on the rules configured in the 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: Application Platform Configuration Guide*.
 - **Validate Item**

For more information about configuring a document's order validation, refer to the *Sterling 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: Application Platform 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 *Sterling 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*.

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](#)

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
<code>chartHeight</code>	The height of the FusionCharts chart.
<code>chartWidth</code>	The width of the FusionCharts chart.
<code>Customer_Org_Code</code>	The Organization Code of the customer for which the quote was created.

Field	Description
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.

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.

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 Business Center. 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 `getPricingRuleList` API is called to retrieve the manual pricing rules that can be used to perform price adjustments. The `getPricingRuleDetails` API is called to retrieve the details of a 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 *Sterling 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 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: Application Platform 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.

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*.

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*.

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.

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 Business Center.

Based on the rules configured in the 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: Application Platform Configuration Guide*.
 - † **Validate Item**
For more information about configuring a document's order validation, refer to the *Sterling 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: Application Platform 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*.

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.

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.

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.

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*.

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.

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 Sales provides 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	<code>SAA_Log_Note_Create_Quote.xml.sample</code>
Quote Sent for Approval	<code>SAA_Log_Note_Send_For_Approval.xml.sample</code>
Quote Presented to Customer	<code>SAA_Log_Note_Send_To_Customer.xml.sample</code>
Quote Converted to Order	<code>SAA_Log_Note_Convert_Quote_To_Order.xml.sample</code>

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

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.

Quote Approval

Sometimes a quote may be placed on hold pending approval 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 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.

The quote approval process involves the following steps:

1. [Requesting Approval](#)
2. [Viewing the Approval Summary](#)
3. [Accepting or Rejecting an Approval Request](#)

Requesting Approval

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.

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.

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

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:

4. The `SCUICognosXMLStandardizer` Java class, a custom API, converts the report from the data mart to a standard XML file.
5. 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.
6. 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.

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 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 *Sterling Business Center: Pricing Administration Guide*.
- You must configure approval plans in the Applications Manager 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 *Sterling Selling and Fulfillment Foundation: Setting Up Quotes*.
- To enable 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 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 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: Application Platform 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.

Generating a Proposal

Sometimes a Field Sales Representative may want to print 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 from a quote. Based on the configuration, a user can either print the proposal directly using a printer, view the proposal in the configured format, and/or save it in a local directory.

The `getCompleteOrderDetails` API is called to retrieve the details of the quote to be displayed in the Generate Proposal screen. A user can select the format in which the proposal should be printed. The `getCommonCodeList` API is called to retrieve the values to be displayed in the **Choose a print template** drop-down list and **Choose a print format** drop-down list. The print templates and print formats are stored as common codes with code types as `SAA_QUOTE_PRNT_TMPLT` and `SAA_QUOTE_PRNT_FRMT` respectively. Based on the print template selected, the attributes that can be included in the proposal are displayed under the **Include In Print** panel. A user can select the attributes that should be printed in the proposal.

The attributes displayed under the **Include In Print** panel are defined in a metadata XML file that is located in the `resources.jar/template/extn/quoteTemplates` directory. Following is a sample metadata XML file:

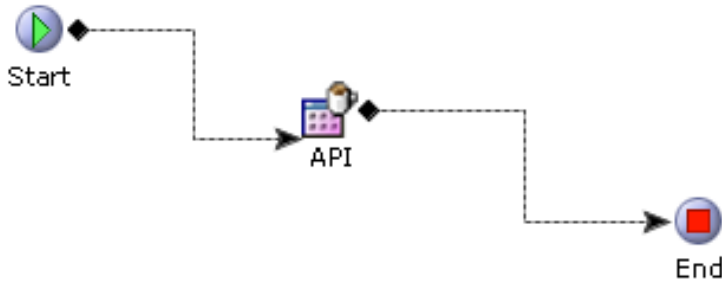
```
<Resource ServiceName="SAA_GenerateQuoteProposal_9.0">
  <Group Name="ProposalHeader" DisplayName="Quote Header">
    <Attribute Name="ProposalTitle" DisplayName="Proposal Title" Value="Y"/>
    <Attribute Name="Status" DisplayName="Status" Value="Y"/>
    <Attribute Name="ProposalID" DisplayName="Proposal ID" Value="Y"/>
    <Attribute Name="ExpirationDate" DisplayName="Expiration Date" Value="Y"/>
    <Attribute Name="ProposalDate" DisplayName="Proposal Date" Value="Y"/>
  </Group>
  <Group Name="Address" DisplayName="Address">
    <Attribute Name="BillTo" DisplayName="Bill To Address" Value="Y"/>
    <Attribute Name="ShipTo" DisplayName="Ship To Address" Value="Y"/>
  </Group>
  <Group Name="OrderLines" DisplayName="Lines">
    <Attribute Name="QuoteLines" DisplayName="Quote Lines" Value="Y"/>
    <Attribute Name="RecommendedLines" DisplayName="Recommended Lines"
Value="Y"/>
  </Group>
  <Group Name="PricingInformation" DisplayName="Quote Totals">
    <Attribute Name="SubTotal" DisplayName="Sub Total" Value="Y"/>
    <Attribute Name="ShippingCharges" DisplayName="Shippng Charges" Value="Y"/>
    <Attribute Name="Taxes" DisplayName="Taxes" Value="Y"/>
    <Attribute Name="GrandTotal" DisplayName="Grand Total" Value="Y"/>
    <Attribute Name="Discounts" DisplayName="Adjustments" Value="Y"/>
  </Group>
  <Group Name="Terms" DisplayName="Terms">
    <Attribute Name="PaymentTerms" DisplayName="Payment Terms" Value="Y"/>
  </Group>
</Resource>
```

```

    <Attribute Name="ShippngTerms" DisplayName="Carrier" Value="Y" />
    <Attribute Name="LevelOfService" DisplayName="Level Of Service" Value="Y" />
    <Attribute Name="DeliveryDate" DisplayName="Delivery Date" Value="Y" />
    <Attribute Name="InstructionText" DisplayName="Shipping Instructions"
Value="Y" />
  </Group
</Resource>

```

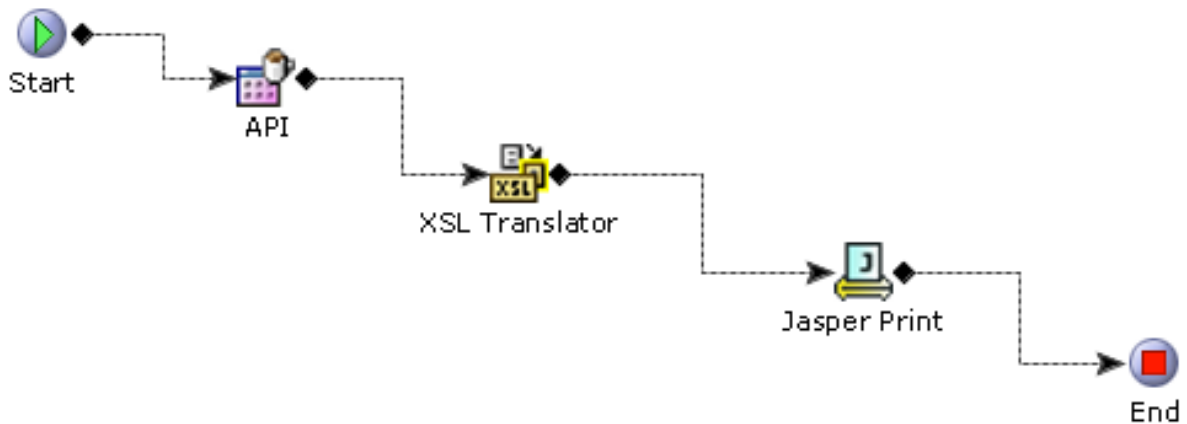
The attributes defined in the metadata XML file are retrieved by invoking the SAA_GetMetaDataXML_9.0 service. This service calls a custom API to get the data defined in the metadata XML file as illustrated in the following figure:



The name of the service that will be used to generate a proposal in a particular print template is defined as the value of the *ServiceName* attribute in the metadata XML file for the corresponding print template.

When a user clicks **Generate** in the Generate Proposal screen, the service that has been configured for the print template 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 input to the service and the print template selected by a user, a custom API is called to retrieve the data that is to be printed. An XSL translator is used to massage the output of the custom API to include the details passed as input to the service, such as the selected template and the attributes to be printed. Depending on the configuration of the JasperPrint component for the service, the proposal is either automatically generated from a printer or generated in a configured format. If the JasperPrint component

has been configured to generate the output to a printer and not to a JasperPrint object such as a PDF or RTF, an exception is thrown.

End-User Impact

You cannot generate a proposal using the JasperReport reports provided out of the box if the ARIALUNI.TTF font is not installed on the application server.

Implementation

This section explains the configurations for this functionality:

- You can configure the print template to generate a proposal. For more information about configuring proposal templates, refer to the *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- For every print template that you configure, you must create a corresponding metadata XML file with the same name as the common code value of the print template in the `resources.jar/template/extn/quoteTemplates` directory. Additionally, this metadata XML file must include the name of the service that will be used to generate the proposal. The name of the service must be defined as the value of the `serviceName` attribute in the metadata XML file. For example, if you configure a print template, `OrderTemplate`, an XML file with `OrderTemplate.xml` as the file name must exist in the `resources.jar/template/extn/quoteTemplates` directory to enable you to use the print template to generate a proposal. Additionally, this `OrderTemplate.xml` file must contain the name of the service that will be invoked to generate a proposal.
- For every print template that you configure, you must configure a corresponding service that will be used to generate the proposal. You must configure the service appropriately to enable a user to print a proposal automatically from a printer or generate the proposal in a PDF, RTF, or any other format that is supported by JasperReports 3.6. The service is created in the Service Definition Framework. For more information about defining service definitions, refer to the *Sterling Selling and Fulfillment Foundation: Application Platform 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 *Sterling Selling and Fulfillment Suite: Applications Configuration Guide*.
- If you run the application server on a Linux environment without an X server or if you cannot connect to the X server due to security restrictions or policies (usually called a "headless" environment), the JasperReport reports are not rendered correctly. To overcome this situation, set the `-Djava.awt.headless` property in the application server startup script to `True`.

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.

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.

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.

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.

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 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 is an application that integrates with the Sterling Selling and Fulfillment Suite to ensure that credit card numbers and stored value card numbers are secure by tokenizing them. Before the *Sterling Sensitive Data Capture Server* 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 Suite 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 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 Sensitive Data Capture Server, Release 1.1: Configuration Guide* and the *Sterling Sensitive Data Capture Server : 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 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.

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.

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](#)

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.

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.

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.

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.

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.

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.

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 2011. Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2011.

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](http://www.ibm.com/legal/copytrade.shtml)" at 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.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

A

alternative quote
creating 21

B

browsing catalog
pagination 25
bundle item
viewing components 46

C

catalog
browsing 24
Channel Applications Manager
configuring 8
configurable item 47
customer detail
modifying for opportunity or quote 18

D

discount advisor report
viewing 38

E

expiration date
modifying 34

I

item
adding to quote 35
deleting from quote 49
moving to quote lines 42
moving to recommended lines 42
validating 35, 44
item quantity

modifying 43

items
filtering 48
items failing validation
viewing 54

N

note
adding and viewing 51

O

opportunity 9
copying quote 23
creating 13
creating quote 22
managing 12
marking as Lost 20
modifying 17
viewing details 15
opportunity life cycle 9

P

payment and shipment terms
modifying 50
preconfigured item 47
primary quote
marking 33
proposal
generating 59

Q

quote 10
abandoning 69
accepting 63
adjusting price 37
converting to order 66
creating 29

- managing 27
- presenting to customer 62
- rejecting 64
- reworking 65
- viewing details 31

quote approval 55

quote life cycle 10

quote line

- configuring or reconfiguring 47

quote name

- modifying 34

R

related item

- adding to a quote 44

S

saved search

- managing 70

U

user preferences

- setting 11