Sterling Field Sales

Customization Guide

Release 9.0

Last updated in HF1

May 2010

Sterling Commerce
An IBM Company

Contents

Overview of Sterling Field Sales Customization

The Sterling Field Sales application can be customized based on your business requirements. This topic provides an overview of the types of customizations possible in Sterling Field Sales.

You can customize the Web UI Framework components and extend the database. Additionally, you can customize the following:

- Related Tasks: You can customize the related tasks that are available in a particular screen. For more information about customizing related tasks, refer to the topic "Customize Related Tasks".
- Product Search and Product Details Screens: You can customize the Product Search screen and the Product Details screen that can be accessed from the home page and the Quote Details screen. For more information about customizing the Product Search screen and the Product Details screen, refer to the topic "Customize Product Search and Product Details Screens".

Setting Up the Extensibility Environment

To customize Sterling Field Sales, you must use the `sfsdev.war` file. The `sfsdev.war` file contains additional folders and web.xml entries that are required for extensibility. In production, you must deploy the `sfs.war` file with the required extensions created using the `sfsdev.war` file.

To customize the Sterling Field Sales application, you must first create the `sfsdev.war` file.

To create the `sfsdev.war` file, run the following command:

```
.\buildwar.sh (.cmd for Windows) -Dappserver=<application server>
-Dwarfiles=sfsdev
```

After running this command, you must perform the following tasks:

1. Deploy the application.
2. Start the application server by passing the following argument:

   `-Dwufdevmode=true`

Web UI Framework Extensibility

You can customize the user interface of an existing out-of-the-box installation of the Sterling Field Sales application using the Extensibility Workbench. It allows you to customize the existing installation at runtime without recompiling or changing the original source code. The Sterling Field Sales application is built on the Ext JS JavaScript framework, version 3.0. The JavaScript API Documentation contains information about the JavaScript utility and classes used by the application. The JavaScript API Documentation can be accessed using the following URL:

```
http://<server>:<port>/<context_root>/yfscommon/jsdocs/30/index.html
```

where context_root is the context root of the `smcfsdocs.war` file, which is used for the context-sensitive help. For more information about deploying the context-sensitive help, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

You can perform the following customizations:

- Differential Screen Extensions
- Override screen extensions
- Mashups
- Struts
- Customizing the web.xml File for Development
- Customizing the web.xml File for Deployment
- Resource Bundles
- Datatype handling
- Validating Fields

For more information about extensibility in the Web UI Framework, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Differential Screen Extensions

Differential extensibility enables you to customize parts of a screen. You can add a new field or hide an existing field in the user interface. Moreover, you can also modify the properties of the out-of-the-box fields. The Java Script Builder (JSB) files are located in the <INSTALL_DIR>/repository/eardata/sfs/war/builder/sfs directory.

For more information about differential extensibility, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Override Screen Extensions

Override extensibility enables you to replace an existing screen completely with a new screen. For more information about override extensibility, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Mashups

You can customize the input XML template and the output XML template of an API call. In addition to this, you can define new mashup layers. The Sterling Field Sales mashups are located in the <INSTALL_DIR>/repository/eardata/sfs/war/mashupxmls/sfs directory. You must place the extended mashups in the <INSTALL_DIR>/extensions/sfs/webpages directory.

The Sterling Field Sales mashups use the following mashup types:

- XAPI (for XAPI calls)
- AggregateXAPI (for multiple mashups)
- SFSAPI (for application-specific mashups)

For more information about extending mashups, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Struts

You can define new Struts and override the existing Struts. The Sterling Field Sales struts files are located in the `<INSTALL_DIR>/repository/eardata/sfs/war/WEB-INF/classes` directory. The `struts.xml.sample` and `struts.properties.sample` files are located in the `<INSTALL_DIR>/repository/eardata/sfs/extn` directory.

For more information about creating and extending a Struts XML file, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Customizing the web.xml File for Development

During development, you can customize the `web.xml` file to view your customizations.

To customize the `web.xml` file during development:

1. Run the buildwar utility to create the `sfsdev.war` file.
2. Copy the web.xml.sample file from the `<INSTALL_DIR>/repository/eardata/sfsdev/extn` directory to the same directory with the file name "web.xml".
3. Modify the newly created `web.xml` files as needed.
4. If you need to add a new servlet or filter, package it in a jar file and run the `<INSTALL_DIR>/bin/install3rdParty.sh` script to include this jar file in a classpath.
5. Run the buildwar utility to create the `sfsdev.war` file.

Customizing the web.xml File for Deployment

After performing the customizations, you must customize the `web.xml` file for deployment.

To customize the `web.xml` file for deployment:

1. Run the buildear or buildwar utility to create the `sfs.war` file.
2. Copy the web.xml.sample file from the `<INSTALL_DIR>/repository/eardata/sfs/extn` directory to the same directory with the file name "web.xml".
3. Modify the newly created web.xml files as needed.
4. If you need to add a new servlet or filter, package it in a jar file and run the `<INSTALL_DIR>/bin/install3rdParty.sh` script to include this jar file in a classpath.
5. Run the buildear or buildwar utility to create the `sfs.war` file.

Resource Bundles

You can define new bundle entries and override the out-of-the-box bundle entries. The client-side bundle files are located in the `<INSTALL_DIR>/repository/eardata/sfs/war/sfs/resources` directory. To load the client-side bundles, the sfsBundle target is used. This target is defined in the `sfscore.jsb` file located in the `<INSTALL_DIR>/repository/eardata/sfs/war/builder/sfs` directory.

For more information about changing bundle files, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Creating a Menu Entry

You can use the Configurator to create a new menu entry in the Sterling Field Sales application. For more information about creating a new menu entry, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Data Handling

In Sterling Field Sales, you can customize data types, field validations, and the verifications performed when changes are made on a screen.

Datatype handling

You can add new data types and modify existing data types. For more information about datatype handling, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Validating Fields

You can validate fields against certain standards, using the default validation system or your own validation system. You can also change the field values so that they are displayed in the required format.

By default, Sterling Field Sales provides validation for the following types of information in the en_US locale:

- E-mail address (using the international accepted standard)
- Telephone number format (locale-specific)

By default, the Regular Expression that is provided for e-mail address is:

```
/^[a-zA-Z][\w\.-]*[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\
.]*[a-zA-Z]$/i
```

By default, the format that is provided for telephone number is (123)-456-7890. However, you can customize this format by overriding the format and deformat method of the sc.app.common.core.format.PhoneNumerFormat class defined in the `formatter.js` file. The `formatter.js` file is located in the `<INSTALL_DIR>/repository/eardata/sfs/war/sfs/appcommon/core/format` directory.

Database Extensibility

You can extend the database based on your requirements. For example, you can modify an existing table by adding a new column. For more information about extending the database, refer to the *Selling and Fulfillment Foundation: Extending the Database Guide*.

Deploying Web UI Framework Extensions

After you perform the customizations, you must place all the extensions in the `<INSTALL_DIR>/extensions/sfs/webpages` directory. You must place all the Java Script files in the `<INSTALL_DIR>/extensions/sfs/webpages/extn` directory. For more information about deploying the Web UI Framework extensions, refer to the *Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.

Customize Related Tasks

In Sterling Field Sales, users can perform tasks from a particular screen using the **Tasks** panel. The **Tasks** panel comprises a **Next Steps** category that contains the tasks that can be performed next from the current screen, and an **Other Tasks** category that contains the other related tasks that can be performed from the current screen. The **Tasks** panel can be customized by adding or removing a related task category and adding or removing a related task within a category. All the customizations pertaining to the related tasks must be performed by placing the corresponding related task metadata file in the `<INSTALL_DIR>/extensions/sfs/webpages/sfs/metadata/relatedtasks` folder.

The related task metadata files are located under the `<INSTALL_DIR>/repository/eardata/sfs/war/sfs/metadata/relatedtasks` folder. The following related task metadata XML files are provided out of the box:

- opportunity.xml
- quote.xml

A related task metadata file contains the following elements:

- Category: You can add, delete, or modify a related task category by defining the `Action` attribute in the Category element in the related task metadata file.

  You can perform the following tasks pertaining to a related task category in the corresponding related task metadata file:

  - Add a new related task category in the corresponding related task metadata file: The new related task category is added at the end of the list in the corresponding **Tasks** panel.
  - Delete a related task category from the corresponding related task metadata file: All the related tasks under a given category are deleted from the corresponding **Tasks** panel.
  - Modify a related task category in the related task metadata file corresponding to a related task category: You can add a new related task, remove an existing related task, and modify an existing related task. Modifying a related task includes modifying only the action that occurs when a user clicks a related task under a given category.
- IncludeCategory: If a related task metadata file contains this element, it indicates that this XML consists of multiple related task metadata files corresponding to various categories. The path to these related task metadata files are defined in the corresponding File attribute of the IncludeCategory element. You can add or delete the IncludeCategory element, which in turn adds or deletes a category if it is present in the File attribute. However, you cannot set the value of the `Action` attribute to MODIFY in the IncludeCategory element.
- Task: You can add, delete, or modify a related task by defining the `Action` attribute in the Task element. However, you need to first set the `Action` attribute to Modify in the corresponding Category element.

Following is a sample related task metadata XML file:

```
<RelatedTasks>
   <Categories>
      <Category CategoryName="b_NextSteps" CategoryID="NextStepsQuoteTask"
SeqNo="10">
         <Tasks>
```

```
            <Task ActionID="SendForApprovalTaskAction" TaskName="b_SendForApproval"
TaskID="SFSQTE00090" SeqNo="10"/>
         </Tasks>
      </Category>
      <Category CategoryName="b_OtherTask" CategoryID="OtherTask" SeqNo="20">
         <Tasks>
            <Task ActionID="QuoteSummaryTaskAction" TaskName="b_ViewQuoteDetails"
TaskID="SFSQTE00120" SeqNo="10"/>
         </Tasks>
      </Category>
   </Categories>
</RelatedTasks>
```

The following table describes the attributes for the Category element as shown in the sample metadata XML file:

Attribute	Description
CategoryID	The unique identifier of the category.
CategoryName	The bundle entry for the category name. This bundle entry should be defined in the client JavaScript bundle file.
SeqNo	The sequence number of the category. The order in which the categories are displayed in the user interface is based on the `SeqNo` attribute.

The following table describes the attributes for the Task element as shown in the sample metadata XML file:

Attribute	Description
ActionID	The unique identifier of the action that will be executed when a user clicks a task on the user interface. The action should be defined on the client side.
TaskName	The bundle entry for the task name. This bundle entry should be defined in the client JavaScript bundle file.
TaskID	The unique identifier of the task.
SeqNo	The sequence number of the task. The order in which the tasks are displayed under a category in the user interface is based on the `SeqNo` attribute.

Add a Related Task Category

You can add a related task category to the list of existing categories in the **Tasks** panel.

To add a related task category:

1. Create a corresponding related task metadata file in the `<INSTALL_DIR>/extensions/sfs/webpages/sfs/metadata/relatedtasks` folder.
2. Create a root element, RelatedTasks.
3. Add a child element, Categories, under the RelatedTasks element.
4. Add a child element, Category, under the Categories element.

5. Perform the following steps under the Category element:
   a. Set the value of the XML attribute, `Action`, to ADD.
   b. Set the value of the XML attribute, `CategoryID`, as required. This value acts as a unique identifier for the related task category.
   c. Set the value of the XML attribute, `CategoryName`, as required. This value indicates the bundle key of the related task category that is displayed in the corresponding **Tasks** panel.

   The related task category is added at the end of the list of categories in the corresponding **Tasks** panel.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after adding a new related task category or related task.

For more information about creating the EAR file, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Modify a Related Task Category

You may want to modify a related task category in order to add, delete, or modify a task under it.

To modify a related task category:

1. Create a corresponding related task metadata file if it does not already exist in the <INSTALL_DIR>/extensions/sfs/webpages/sfs/metadata/relatedtasks folder.
2. Create a root element, RelatedTasks.
3. Add a child element, Categories, under the RelatedTasks element.
4. Add a child element, Category, under the Categories element.
5. Perform the following steps under the Category element:
   a. Set the value of the XML attribute, `Action`, to MODIFY.
   b. Set the value of the XML attribute, `CategoryID`, as required. This value acts as a unique identifier for the related task category.
   c. Perform any or all of the following tasks:

      - Add a Related Task

      - Modify a Related Task

      - Delete a Related Task

Add a Related Task

If you have created a new screen, you can add a new related task in the **Tasks** panel, which, when clicked, displays the corresponding screen.

To add a related task:

1. Add an element, Tasks, as a child element of the Category element.
2. Add an element, Task, as a child element of the Tasks element.

3. Perform the following steps under the Task element:
   a. Set the value of the corresponding XML attribute, `Action`, to ADD.
   b. Set the value of the XML attribute, `TaskID`, as required. This value acts as a unique identifier and resource identifier for the related task.
   c. Set the value of the XML attribute, `TaskName`, as required. This value indicates the bundle key of the related task that is displayed in the corresponding **Tasks** panel.
   d. Set the value of the XML attribute, `PermissionID`, as required. This indicates the resource that controls the permission to this task.
   e. Set the value of the XML attribute, `ActionID`, as required. This value indicates the action that will return the corresponding screen in which the related task can be performed.

   The related task is added to the related task category in the corresponding **Tasks** panel.

Modify a Related Task

You can modify the `ActionID` attribute of the Task element so that when the corresponding related task is clicked in the **Tasks** panel, a new screen linked to the modified `ActionID` attribute is displayed. For example, if you have created a new screen, you can modify the existing related task in the **Tasks** panel, which, when clicked, displays the corresponding screen.

To modify a related task:

1. Add an element, Tasks, as a child element of the Category element.
2. Add an element, Task, as a child element of the Tasks element.
3. Perform the following steps under the Task element:
   a. Set the value of the XML attribute, `TaskID`, as required. This value acts as a unique identifier and resource identifier for the related task.
   b. Define the value of the corresponding XML attribute, `ActionID`, as required. This value indicates the action that will return the corresponding screen in which the related task can be performed.

   The related task is modified. When a user clicks the related task in the corresponding **Tasks** panel, the screen to which the value in the `ActionID` attribute refers, is displayed.

Delete a Related Task

To delete a related task:

1. Add an element, Tasks, as a child element of the Category element.
2. Add an element, Task, as a child element of the Tasks element.
3. Perform the following steps under the Task element:
   a. Set the value of the XML attribute, `TaskID`, as required. This value acts as a unique identifier and resource identifier for the related task.
   b. Set the value of the corresponding XML attribute, `Action`, to DELETE.

   The related task is deleted from the related task category in the corresponding **Tasks** panel.

Delete a Related Task Category

You can delete a related task category from the **Tasks** panel.

To delete a related task category:

1. Create a corresponding related task metadata file in the `<INSTALL_DIR>/extensions/sfs/webpages/sfs/metadata/relatedtasks` folder.
2. Create a root element, RelatedTasks.
3. Add a child element, Categories, under the RelatedTasks element.
4. Add a child element, Category, under the Categories element.
5. Perform the following steps under the Category element:
   a. Set the value of the XML attribute, `CategoryID`, as required. This value acts as a unique identifier for the related task category.
   b. Set the value of the corresponding XML attribute, `Action`, to DELETE.

   The related task category along with the related tasks under that category are deleted from the **Tasks** panel.

   Note: Ensure that you rebuild the Enterprise ARchive (EAR) after deleting a related task category or related task.

   For more information about creating the EAR file, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Customize Product Search and Product Details Screens

You can customize the Product Search screen and the Product Details screen that can be accessed from the home page and the Quote Details screen. The Product Search screen and the Product Details screen are built using the Ext JS JavaScript framework templates.

To customize the Ext JS templates used for the Product Search screen and the Product Details screen:

1. Create a folder structure as follows:

   `<INSTALL_DIR>/extensions/sfs/webpages/sfs/itembrowsing/templates`

2. Under the `templates` folder, create a template file with the same name as the one you want to override.
3. Perform the required changes in the newly created template file.

Note: Ensure that you rebuild the Enterprise ARchive (EAR) after customizing the Ext JS templates.

For more information about creating the EAR file, refer to the *Selling and Fulfillment Foundation: Installation Guide*.

Index