Sterling Field Sales

Localization Guide

Release 9.0

March 2010

Sterling Commerce
An IBM Company

Contents

Localizing the Sterling Field Sales Application: An Overview

Localization is the process of adapting an application to a particular locale to enable a user in that locale to use the application. All user interface literals of the Sterling Field Sales application are stored in the application resource bundles and the database.

For more information about localizing resource bundle files, refer to the topic "Localizing Resource Bundles". For more information about localizing database literals, refer to the topic "Localize Factory Setup Data".

You can localize the following components of the Sterling Field Sales application:

- Literals
- Date and Time Formats
- Time Zones
- Numeric Formats
- Currency
- Cascading Style Sheets (CSS)

For more information about understanding the concept of localization and localization of components, refer to the following guides:

- *Selling and Fulfillment Foundation: Localization Guide*
- *Selling and Fulfillment Foundation: Localizing the Web UI Framework*
- *Business Center: Localization Guide*

Localizing Resource Bundles

The user interface components in Sterling Field Sales use resource bundles that contain the literals displayed throughout the application. After a third party carries out translation in multiple languages, a folder corresponding to each of the locale, and comprising the corresponding bundle files, is generated. For example, if a locale has been created for Japan in Japanese language, a folder called `ja_JP` is generated. In addition, the literals that are used in customized screens and have their own resource bundles, are also considered during the localization process.

A resource bundle is a file that comprises resource bundle keys and corresponding values. The values pertaining to these resource bundle keys are translated as part of localization. Each field in the user interface has a key associated with it. To display the translated literal pertaining to a field in the user interface, the localized value of the key associated with the field is fetched from the bundle file.

Use the following files to localize the Sterling Field Sales application:

- `bundle-index.properties`

  You can use this file to localize the following:

  - Error and warning messages displayed on the user interface.
  - Descriptions pertaining to labels, panels, and headings in the user interface.
  - Dynamic data within a literal. For example, the user interface may have to display a literal that informs a user that an input value cannot exceed a certain number of characters, and that the number of characters is dynamic. In such a scenario, if the maximum length for a description is set at 428 characters, 428 is the value of the parameter. The corresponding bundle entry is defined as "b_MaxCharLengthExceeded":"The value cannot exceed {0} characters".

  The `bundle-index.properties` file is located in the `/Files/resources/` folder in the localization kit.
- `sfsbundle.properties`

  You can use this file to localize entities such as the home page options. The `sfsbundle.properties` file is located in the `/Files/resources/` folder in the localization kit.

Classification of Resource Bundles

The resource bundles can be categorized as server-side bundles and client-side bundles. The server-side bundles pertain to the data that is localized in the application server, and stored in the `sfsbundle.properties` file. For example, the options displayed on the home page are localized by the server-side bundles.

The client-side bundles pertain to the data that is localized at the client-side. The client-side resource bundles are stored in the `bundle-index.properties` file. For example, the labels on the user interface are localized by the client-side resource bundles.

Localize the bundle-index.properties File

To localize the `bundle-index.properties` file:

1. If you are customizing the Sterling Field Sales application, you must regenerate the `bundle-index.properties` file. Therefore, run the following script in `<INSTALL_DIR>/bin` to extract the `bundle-index.properties` file:

   For Windows:

   ```
   ./sci_ant.cmd -f jsUtil.xml bundle.index
   -Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
   -Dindexdir=<INSTALL_DIR>/repository/eardata/sfs/localization_index
   ```

   For Linux/UNIX:

   ```
   ./sci_ant.sh -f jsUtil.xml bundle.index
   -Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
   -Dindexdir=<INSTALL_DIR>/repository/eardata/sfs/localization_index
   ```

   `<INSTALL_DIR>` is the directory in which Sterling Field Sales is installed.

   A new folder, `localization_index`, containing the following files is created:

   - `opts`
   - `context-index`
   - `bundle-index.properties`

2. Copy the `bundle-index.properties` and rename it as `bundle-index_<language_code>_<country_code>.properties` based on the locale.

   For example, if you want to localize the application for France in French language, copy the `bundle-index.properties` and rename it as `bundle-index_fr_FR.properties`.

3. Edit the `bundle-index_<language_code>_<country_code>.properties` file with the translations.

4. Save the `bundle-index_<language_code>_<country_code>.properties` file.

   The `bundle-index.properties` file is localized.

   After this, perform the postlocalization tasks on the client side. For more information about the postlocalization tasks, refer to the topic "Postlocalization Tasks".

Localize the sfsbundle.properties File

To localize the `sfsbundle.properties` file:

1. If you are customizing the Sterling Field Sales application, save the extended resource bundles as `<Extn_SFS_Bundle>_<language_code>_<country_code>.properties` in the `<INSTALL_DIR>/resources/extn/` directory.

   `<Extn_SFS_Bundle>` is the bundle file containing the extended resource bundles pertaining to Sterling Field Sales and `<INSTALL_DIR>` is the directory in which Sterling Field Sales is installed.

2. Copy the `sfsbundle.properties` and rename it as `sfsbundle_<`*`language_code`*`>_<`*`country_code`*`>.properties` based on the locale.

   For example, if you want to localize the application for France in French language, copy the `sfsbundle.properties` file and rename it as `sfsbundle_fr_FR.properties`.

3. Edit the `sfsbundle_<`*`language_code`*`>_<`*`country_code`*`>.properties` file with the translations.

4. Save the `sfsbundle_<`*`language_code`*`>_<`*`country_code`*`>.properties` file.

   The `sfsbundle.properties` file is localized.

   After this, perform the postlocalization tasks on the server side. For more information about the postlocalization tasks, refer to the topic "Postlocalization Tasks".

Note: It is recommended that when you localize the literals, you keep the length of the literals short in order to avoid them getting truncated.

Localize Factory Setup Data

Besides storing your transactional data, the database also stores configuration data, such as error codes and item descriptions of various attributes. This means that the database may have to store values in a language-specific format. If these database literals are not localized, screen literals are displayed inconsistently, with some being displayed in the localized language, and others being displayed in English.

The `literals2translate.properties` file, which is available as part of the localization kit, contains all the Sterling Field Sales factory setup literals that have to be localized.

To localize the factory setup data, perform the following steps:

1. Localize the `literals2translate.properties` file.
2. Rename the localized file as `en_US_sfslocalizedstrings_<`*`language_code`*`>_<`*`country_code`*`>.properties`.

   For example, to localize the factory setup data for the fr_FR (French, France) locale, you must rename the localized `literals2translate.properties` file as `en_US_sfslocalizedstrings_fr_FR.properties`.
3. Copy the `en_US_sfslocalizedstrings_<`*`language_code`*`>_<`*`country_code`*`>.properties` file to the `<`*`INSTALL_DIR`*`>/repository/factorysetup/complete_installation/XMLS/<`*`language_code`*`>_<`*`country_code`*`>` folder.
4. Load the language pack translations by running the Localized String Reconciler tool.

   For more information about loading the language pack translations, refer to the *Sterling Selling and Fulfillment Suite: Applications Installation Guide*.

Postlocalization Tasks

The localized resource bundles are not automatically used by the Sterling Field Sales application. To ensure that the localized bundles are used by Sterling Field Sales, you must re-create the Selling and Fulfillment Foundation Enterprise ARchive (EAR) package.

The following sections describe the postlocalization activities on the server side and the client side.

Postlocalization Tasks on the Server Side

After you have localized the bundles, perform the following steps on the server side:

1. Copy the localized file to the `<INSTALL_DIR>/resources` folder.
2. Create the resource JAR using the following command:

   For Windows:

   ```
   deployer.cmd -t resourcejar
   ```

   For Linux/UNIX:

   ```
   deployer.sh -t resourcejar
   ```

3. Re-create the `smcfs.ear` file. This file is the Selling and Fulfillment Foundation EAR package.

   For more information about creating the EAR file, refer to the *Sterling Field Sales: Deployment Guide*.

Postlocalization Tasks on the Client Side

After you have localized the bundles, perform the following steps on the client side:

1. Navigate to the `<INSTALL_DIR>/bin` folder:
2. Run the following command:

   For Windows:

   ```
   ./sci_ant.cmd -f jsUtil.xml bundle.map
   -Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
   -Dindexdir=<localization directory having the localized bundle-index file>
   -Dindexfile=<localization directory having the localized bundle-index
   file>/<localized bundle-index file> -Dwebcontentdepth=0
   ```

   For Linux/UNIX:

   ```
   ./sci_ant.sh -f jsUtil.xml bundle.map
   -Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
   -Dindexdir=<localization directory having the localized bundle-index file>
   ```

```
-Dindexfile=<localization directory having the localized bundle-index
file>/<localized bundle-index file> -Dwebcontentdepth=0
```

For example, if you have localized to the French locale, and the localization directory is `<INSTALL_DIR>/repository/eardata/sfs/localization_index`, and the localized bundle-index file is `bundle-index_fr_FR.properties`, run the following command:

For Windows:

```
./sci_ant.cmd -f jsUtil.xml bundle.map
-Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
-Dindexdir=<INSTALL_DIR>/repository/eardata/sfs/localization_index
-Dindexfile=<INSTALL_DIR>/repository/eardata/sfs/localization_index/bundle-
index_fr_FR.properties -Dwebcontentdepth=0
```

For Linux/UNIX:

```
./sci_ant.sh -f jsUtil.xml bundle.map
-Dsourcedir=<INSTALL_DIR>/repository/eardata/sfs/war
-Dindexdir=<INSTALL_DIR>/repository/eardata/sfs/localization_index
-Dindexfile=<INSTALL_DIR>/repository/eardata/sfs/localization_index/bundle-
index_fr_FR.properties -Dwebcontentdepth=0
```

3. Run the following command:

   For Windows:

   ```
   ./sci_ant.cmd -f jsUtil.xml minify-js
   -DsrcDir=<INSTALL_DIR>/repository/eardata/sfs/war
   -DdestDir=<INSTALL_DIR>/repository/eardata/sfs/war -Dminify=true
   -DcreateIndividualFile=false
   -DjsbDir=<INSTALL_DIR>/repository/eardata/sfs/war/builder/sfs
   ```

   For Linux/UNIX:

   ```
   ./sci_ant.sh -f jsUtil.xml minify-js
   -DsrcDir=<INSTALL_DIR>/repository/eardata/sfs/war
   -DdestDir=<INSTALL_DIR>/repository/eardata/sfs/war -Dminify=true
   -DcreateIndividualFile=false
   -DjsbDir=<INSTALL_DIR>/repository/eardata/sfs/war/builder/sfs
   ```

4. Re-create the `smcfs.ear file`. This file is the Selling and Fulfillment Foundation EAR package.

   For more information about creating the EAR file, refer to the *Sterling Field Sales: Deployment Guide*.

Note: After localizing the application, you may want to verify that the application has been localized. However, because resource bundles are cached in a Web browser, it is recommended that you clear the Web cache before verifying the localization.

Index