

# **Sterling Field Sales**

---

## **Localization Guide**

**Release 9.1**



# 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 17.

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

Localizing the IBM Sterling Field Sales Application: An Overview .....	5
Localize Telephone Number Format .....	6
Localizing UI Branding .....	7
Localizing Themes .....	7
Localizing Icons .....	7
Localizing Resource Bundles .....	9
Classification of Resource Bundles .....	9
Localize Client-Side Literals .....	10
Localize Server-Side Literals .....	10
Localize Server-Side Application Add-In Literals .....	11
Localize New bundle js Entries .....	11
Localizing the Factory Setup Data .....	14
Postlocalization Tasks .....	15
Postlocalization Tasks for Server Side Literals .....	15
Postlocalization Tasks for Client Side Literals .....	15
<b>Notices</b> .....	<b>17</b>
Trademarks .....	19
<b>Index</b> .....	<b>21</b>

---



---

## Localizing the IBM 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 IBM® 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 "[Localizing the 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 the localization of the Sterling Field Sales components, refer to the following guides:

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

Additionally, you can localize the JasperReport reports. For more information about localizing the server-side Application Add-in literals, refer to the sub-topic "[Localize Server-Side Application Add-In Literals](#)".

**Note:** It is recommended that along with Sterling Field Sales localization, you perform the browser localization as well.

---

## Localize Telephone Number Format

You can localize the telephone number format in Sterling Field Sales. The telephone number format is defined in the `localizable_formatter.js` file located in the `<INSTALL_DIR>/repository/eardata/sfs/war/localization/sfs/appcommon/core/format` folder.

To localize the telephone number format, perform the following steps:

1. Create the `<INSTALL_DIR>/repository/eardata/sfs/war/localization/<language_code>/<country_code>/sfs/appcommon/core/format` folder and copy the `formatter.js` file to this folder.
2. Modify the `formatter.js` file that you copied in Step 1, as required.  
**Note:** Based on the modifications, you must override the `format` and `deformat` method of the `sc.app.common.core.format.PhoneNumerFormat` class defined in the `formatter.js` file.
3. After the localization is completed, the JavaScript files must be minified. For more information about minifying JavaScript files, refer to the *Sterling Selling and Fulfillment Foundation: Customizing the Web UI Framework Guide*.
4. Re-create the `smcfs.ear` file. This file is the Sterling Selling and Fulfillment Foundation EAR package. For more information about creating the EAR file, refer to the *Sterling Field Sales: Deployment Guide*.

---

## Localizing UI Branding

As part of localizing the user interface (UI) branding, you can localize both themes and icons.

### Localizing Themes

The look-and-feel (which includes themes, images, and icons) of the UI in the Web UI Framework is CSS file-driven.

You can add, override, or modify CSS entries by providing the locale-specific CSS files. The default theme CSS files are present in the following directory:

```
<INSTALL_DIR>/repository/eardata/sfs/war/sfs/resources/default/css
```

**Note:** To override the CSS entries irrespective of the dependency on the existing CSS file, add the custom themes. For more information about adding custom themes, refer to the *Sterling Field Sales: Customization Guide*.

To provide CSS entries for a locale:

1. Create a new file with the same name in the following directory:

```
<INSTALL_DIR>/repository/eardata/sfs/war/localization/<locale>/sfs/resources/default/css
```

where <locale> is in the format <language\_code>/<country\_code>/<variant>. Note that <variant> is optional.

For example,

```
<INSTALL_DIR>/repository/eardata/sfs/war/localization/en/US/EST/sfs/resources/default/sfs.css.
```

2. Add CSS entries in the new file.

For example, to localize the theme of the title in the screen title panel, add the following entry:

```
.sc-screen-title .sfs-titlebar label {  
    background:none repeat scroll 0 50% # FFFF66;  
    color #7A991A;  
    display:block;  
    font-family: Comic Sans MS;  
    font-size: 15pt;  
}
```

3. Rebuild the EAR file.

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

### Localizing Icons

Icons are displayed in the user interface using the CSS entries. To localize the icons, you must override the appropriate CSS entries such that they point to the localized version of the icons.

To localize the icons for a locale:

1. Create a new file with the same name in the following directory:

```
<INSTALL_DIR>/repository/eardata/sfs/war/localization/<locale>/sfs/resources/default/css
```

where <locale> is in the format <language\_code>/<country\_code>/<variant>. Note that <variant> is optional.

For example,

```
<INSTALL_DIR>/repository/eardata/sfs/war/localization/en/US/EST/sfs/resources/default/css/sfs-images.css
```

2. Add CSS entries in the new file.

For example, to localize the application logo in the application header, add the following entry:

```
.sc-panel-logo {  
    background: transparent url(../icons/app_logo.png) no-repeat scroll 0 0;  
    height: 50px;  
    width: 175px;  
}
```

3. Copy the localized icon to the relative path specified in the background property.

For example,

```
<INSTALL_DIR>/repository/eardata/sfs/war/localization/en/US/EST/sfs/resources/default/icons/apps_logo.png
```

4. Rebuild the EAR file.

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



---

## Localizing Resource Bundles

The user interface (UI) 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.

### 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. As part of server side localization, you can localize entities such as the home page options.

The client-side bundles pertain to the data that is localized at the client-side. Following are some of the UI components that you can localize as part of client-side localization:

- 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"`.

## Localize Client-Side Literals

To localize the client-side literals, you must run the JavaScript bundle collector utility by performing the following steps:

1. If you are localizing the Sterling Field Sales application, you must generate the `bundle-index.properties` file. Therefore, run the following script in `<INSTALL_DIR>/bin` to extract the client-side literals in 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
```

where `-Dindexdir` is the output directory.

### 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
```

where `-Dindexdir` is the output directory.

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

```
├── opts  
├── context-index  
└── bundle-index.properties
```

2. Send the `bundle-index.properties` file for translation.
3. After localization, save the localized `bundle-index.properties` file as `bundle-index_<language_code>_<country_code>.properties` based on the locale in the `<INSTALL_DIR>/repository/eardata/sfs/localization_index` folder.

For example, if you want to localize the application for France in French language, save the localized `bundle-index.properties` as `bundle-index_fr_FR.properties`.

**Note:** If you have not localized the Web UI Framework client-side literals already, you must localize the Web UI Framework client-side literals by running the JavaScript bundle collector utility. For more information about running the bundle collector utility to localize platform files, refer to *Sterling Selling and Fulfillment Foundation: Localizing the Web UI Framework Guide*.

4. Perform the postlocalization tasks on the client side. For more information about the postlocalization tasks, refer to the topic "[Postlocalization Tasks](#)".

## Localize Server-Side Literals

To localize the server-side literals, perform the following steps:

1. Send the `sfsbundle.properties` file located in the `<INSTALL_DIR>/resources` folder for translation.
2. Save the localized `sfsbundle.properties` file as `sfsbundle_<language_code>_<country_code>.properties` based on the locale in the `<INSTALL_DIR>/resources` folder. For example, if you localize the application for France in

French language, save the localized `sfsbundle.properties` file as `sfsbundle_fr_FR.properties`.

3. 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 Server-Side Application Add-In Literals

To localize the server-side Application Add-in literals:

1. Send the `ycdbundle.properties` file located in the `<INSTALL_DIR>/resources` folder for translation.
2. Save the localized `ycdbundle.properties` file as `ycdbundle_<language_code>_<country_code>.properties` based on the locale in the `<INSTALL_DIR>/resources` folder. For example, if you localize the application for France in French language, save the localized `ycdbundle.properties` file as `ycdbundle_fr_FR.properties`.
3. Perform the postlocalization tasks on the server side. For more information about the postlocalization tasks, refer to the topic "[Postlocalization Tasks](#)".

## Localize New bundle js Entries

This section describes the steps required to update your localized bundle JavaScript files with the new entries that have been added in the English `bundle.js` files after localization of the application is completed.

To update the `bundle.js` files after installing the changes in the English `bundle.js` files:

1. Run the `jsUtil` tool to generate a fresh `bundle-index.properties` file to obtain the newly added English bundle entries.

Navigate to the `<INSTALL_DIR>/bin` folder and run the following command:

### 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  
where -Dindexdir is the output directory.
```

### 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  
where -Dindexdir is the output directory.
```

The `bundle-index.properties` file is generated in the `localization_index` directory.

2. Compare the `bundle-index.properties` file generated in Step 1, and your localized `bundle-index_<language_code>_<country_code>.properties` file to determine the newly added entries. Translate the newly added entries and add them in your localized `bundle-index_<language_code>_<country_code>.properties` file.
3. Run the `jsUtil` tool in `map` mode to regenerate the localized `bundle.js` files from the updated `bundle-index_<language_code>_<country_code>.properties` file.

Navigate to the `<INSTALL_DIR>/bin` folder and 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_<language_code>_<country_code>.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_<language_code>_<country_code>.properties -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.properties` file is `bundle-index_fr_FR.properties`, you must 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
```

4. Regenerate the minified localized bundle js files.

Navigate to the <INSTALL\_DIR>/bin folder and 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
```

5. Rebuild the EAR and re-deploy the application. For more information about creating the EAR file, refer to the *Sterling Field Sales: Deployment Guide*.

For more information about generating minified JavaScript source files, refer to *Sterling Field Sales: Customization Guide*.

---

## Localizing the Factory Setup Data

Besides storing your transactional data, the database also stores configuration data, such as error codes. This means that the database may need 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. Sterling Field Sales enables you to localize the factory default XML and user-configured attributes for multiple locales. For more information about localizing factory setup, refer to the *Sterling Selling and Fulfillment Foundation: Localization Guide*.

**Note:** If the Sterling Call Center and Sterling Store and/or Sterling Web applications are installed, do not rerun the localization steps.

---

## 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 Sterling 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 for Server Side Literals

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

1. Navigate to the `<INSTALL_DIR>/bin` 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 Sterling Selling and Fulfillment Foundation EAR package.

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

### Postlocalization Tasks for Client Side Literals

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

1. Navigate to the `<INSTALL_DIR>/bin` folder:
2. To generate localized JavaScript source files from the localized `bundle-index.properties` file, run the following command:

**For Windows:**

```
sci_ant.cmd -f jsUtil.xml bundle.map  
-D sourcedir=<INSTALL_DIR>/repository/eardata/sfs/war  
-D indexedir=<INSTALL_DIR>/repository/eardata/sfs/localization_index  
-D indexfile=<INSTALL_DIR>/repository/eardata/sfs/localization_index/bundle-  
index_<language_code>_<country_code>.properties -Dwebcontentdepth=0
```

where `-D indexedir` is the output directory.

**For Linux/UNIX:**

```
./sci_ant.sh -f jsUtil.xml bundle.map  
-D sourcedir=<INSTALL_DIR>/repository/eardata/sfs/war  
-D indexedir=<INSTALL_DIR>/repository/eardata/sfs/localization_index
```

```
-Dindexfile=<INSTALL_DIR>/repository/eardata/sfs/localization_index/bundle-  
index_<language_code>_<country_code>.properties -Dwebcontentdepth=0
```

For example, if you have localized for 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  
-Dsrcdir=<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
```

where -Dindexdir is the output directory.

**For Linux/UNIX:**

```
./sci_ant.sh -f jsUtil.xml bundle.map  
-Dsrcdir=<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
```

where -Dindexdir is the output directory.

3. To generate minified JavaScript source files from the localized bundle-index.properties file, 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 Sterling Selling and Fulfillment Foundation EAR package. For more information about creating the EAR file, refer to the *Sterling Field Sales: Deployment Guide*.

For more information about generating minified JavaScript source files, refer to *Sterling Field Sales: Customization 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.



---

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

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

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

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

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

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.

## C

client-side literals  
localizing 10

## F

factory setup  
localizing 14

## L

localizing Sterling Field Sales 5

## P

postlocalization  
performing client-side tasks 15  
performing server-side tasks 15  
postlocalization tasks 15

## R

resource bundles  
classifying 9  
localizing 9

## S

server-side Add-in literals  
localizing 11  
server-side literals  
localizing 10