

Business Center™

Localization Guide

Release 8.5

Last updated in HF22

August 2010

Sterling Commerce
An IBM Company

© Copyright 2009 Sterling Commerce, Inc. All rights reserved.

Additional copyright information is located on the documentation library:
<http://www.sterlingcommerce.com/Documentation/MCSF85/CopyrightPage.htm>

Contents

Localizing the Business Center Application: An Overview	5
Localize Resource Bundles	7
Localize the <code>bundle-index</code> File	8
Localize the <code>sbcbundle.properties</code> File	8
Localizing New <code>bundle.js</code> Entries	9
Localize Factory Setup	12
Localizing UI Branding	13
Localizing Themes	13
Localizing Icons	13
Localizing Certain Descriptions and Values in Item Administration	15
Localizing the Sentences in the Coupons and Pricing Rules Screens	17
Post-Localization Tasks	19
Post-Localization on the Server Side	19
Post-Localization on the Client Side	19
Index	22

Localizing the Business Center Application: An Overview

The Business Center™ application can be used by users from different locales to perform tasks such as creating items, viewing the price of items, and so on. In such a scenario, it is important to localize the user interface (UI) components of the application in the language and format that is specific to a user's locale.

Following are the Business Center UI components that can be localized:

Category	Components	Applicable/Localizable	For more details, refer to the
Framework Localization		Yes	<i>Selling and Fulfillment Foundation: Localizing the Web UI Framework Guide</i>
Locale-Specific Formats			
	Date/Time	Yes	<i>Selling and Fulfillment Foundation: Localizing the Web UI Framework Guide</i>
	Numeric Group Separator	No	
	Numeric Decimal Separator	Yes	<i>Selling and Fulfillment Foundation: Localizing the Web UI Framework guide</i>
	E-mail	No	
	Phone Number	No	
	Credit Card Number	No	
	Currency	Yes	<i>Selling and Fulfillment Foundation: Application Platform Configuration Guide</i>
	Unit of measure for dimensions, volume, and weight	No	
	Time Zones	Yes	<i>Selling and Fulfillment Foundation: Localization Guide</i>
Literals and Data			
	Resource Bundles	Yes	<i>Selling and Fulfillment Foundation: Localizing the Web UI Framework Guide</i> and Localize Resource Bundles topic

Category	Components	Applicable/Localizable	For more details, refer to the
	Factory Setup	Yes	<i>Selling and Fulfillment Foundation: Localization Guide</i> and Localize Factory Setup topic
	Master Data - Item related	Yes	Localizing Certain Descriptions and Values in Item Administration topic
UI Branding			
	Structure of Panel Components	Yes	<i>Selling and Fulfillment Foundation: Localizing the Web UI Framework Guide</i> and Localizing the Sentences in the Coupons and Pricing Rules Screens topic
	Localizing Themes / Cascading Style Sheets (CSS)	Yes	Localizing UI Branding topic
	Localizing Icons	Yes	Localizing UI Branding topic

Note: When you localized `validation.js` for a locale, ensure that the date/time you configure in the `validation.js` is same as the one you have configured as part of the locale configuration.

Localize Resource Bundles

The user interface components in the Business Center™ application use resource bundles that contain all the static literals displayed throughout the application. After a third-party performs 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 also translated as part of localization. Each field in the user interface has a key associated with it. To display the translated literal for 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 Business Center application:

◆ `bundle-index`

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
- ◆ Sentences displayed in the user interface. For more information about localizing sentences, refer to the topic, *Localizing the Sentences in the Coupons and Pricing Rules Screens*.
- ◆ 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 this case, if the maximum character 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"`.

◆ `sbcbundle.properties`

You can use the `sbcbundle.properties` file to localize entities such as menu entries, related tasks, and advanced search criteria. The `sbcbundle.properties` file is located in the `<INSTALL_DIR>/resources/` folder.

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 at the application server, and are stored in the `sbcbundle.properties` file. For example, the options displayed in the menu 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 `bundle-index` file. For example, the labels on the user interface are localized by the client-side resource bundles.

Localize the bundle-index File

To localize the `bundle-index` file:

1. If you are customizing the Business Center application, you must generate the `bundle-index` file. Therefore, run the following script in `<INSTALL_DIR>/bin` to extract the literals from the bundle javascript files to the `bundle-index` file:

For Windows:

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

where `-Dindexdir` is the output directory.

For Linux/UNIX:

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

where `-Dindexdir` is the output directory.

A new folder, `localization_index`, containing the following files is created in the `-Dindexdir` directory:

- ◆ `bundle-index`
- ◆ `duplicate-keys-map`

For more information about the generated files, refer to the *Selling and Fulfillment Foundation: Localizing the Web UI Framework*.

2. Copy `bundle-index` to the `bundle-index_<language>_<country>` file pertaining to the locale. For example, if you want to localize the application for France in French language, copy `bundle-index` to the `bundle-index_fr_FR` file.

3. Edit the `bundle-index_<language>_<country>` file with the translations, and save the file.

The `bundle-index` file is localized. After this, perform the post-localization steps on the client side. For more information about the post-localization steps, refer to the topic, *Post-Localization Tasks*.

Note: Ensure that you run the script in the `bundle.index` mode for the Web UI Framework files.

For more information about the bundle collector utility, refer to the *Selling and Fulfillment Foundation: Localizing the Web UI Framework*.

Localize the sbcbundle.properties File

To localize the `sbcbundle.properties` file:

1. If you are customizing the Business Center application, save the extended resource bundles as `<INSTALL_DIR>/resources/extn/<Extn_SBC_Bundle>_<language>_<country>.properties` where `<Extn_SBC_Bundle>` is the bundle file containing the extended resource bundles for Business Center.

2. Copy the `sbcbundle.properties` file to the `sbcbundle_<language>_<country>.properties` file pertaining to the locale.
For example, if you want to localize the application for France in French language, copy `sbcbundle.properties` to the `sbcbundle_fr_FR.properties` file.
3. Edit the `sbcbundle_<language>_<country>.properties` file with the translations, and save the file.
The `sbcbundle.properties` file is localized. After this, perform the post-localization steps on the server side. For more information about the post-localization steps, refer to the topic, [Post-Localization Tasks](#).

Localizing New `bundle.js` Entries

This section describes the steps required to update your localized `bundle.js` 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` 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/sbc/war
-Dindexdir=<INSTALL_DIR>/repository/eardata/sbc/localization_index
```

For Linux/UNIX:

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

where `<INSTALL_DIR>` is the directory in which Business Center is installed.

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

2. Compare the `bundle-index` file generated in Step 1, and your localized `bundle-index_<language>_<country>` file to determine the newly added entries. Translate the newly added entries and add them in your localized `bundle-index_<language>_<country>` file.

3. Run the jsUtil tool in map mode to regenerate the localized bundle js files from the updated bundle-index_<language>_<country> 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/sbc/war
-Dindexdir=<localization directory containing the localized bundle-index
file>
-Dindexfile=<localization directory containing 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/sbc/war
-Dindexdir=<localization directory containing the localized bundle-index
file>
-Dindexfile=<localization directory containing 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/sbc/localization_index, and the localized bundle-index file is bundle-index_fr_FR, you must run the following command:

For Windows:

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

For Linux/UNIX:

```
./sci_ant.sh -f jsUtil.xml bundle.map
-Dsourcedir=<INSTALL_DIR>/repository/eardata/sbc/war
-Dindexdir=<INSTALL_DIR>/repository/eardata/sbc/localization_index
-Dindexfile=<INSTALL_DIR>/repository/eardata/sbc/localization_index/bundle-
index_
fr_FR -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/sbc/war
-DdestDir=<INSTALL_DIR>/repository/eardata/sbc/war -Dminify=true
-DcreateIndividualFile=false
-DjsbDir=<INSTALL_DIR>/repository/eardata/sbc/war/builder/sbc
```

For Linux/UNIX:

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

5. Rebuild the EAR and re-deploy the application.

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

Localize Factory Setup

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.

For more information about localizing factory setup, refer to the *Selling and Fulfillment Foundation: Localization Guide*.

Localizing UI Branding

Localizing Themes

The Look and Feel (which include 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 provided by Business Center™ are present in:

```
<INSTALL_DIR>/repository/eardata/sbc/war/sbc/resources/core/css
```

To provide CSS entries for a locale:

1. You need to place a new file with same name in the following directory:

```
<INSTALL_DIR>/repository/eardata/sbc/war/localization/<locale>/sbc/resources/core/css
```

where <locale> is in the format <language_code>/<country_code>/<variant>.

For example,

```
<INSTALL_DIR>/repository/eardata/sbc/war/localization/en/US/EST/sbc/resources/core/css/sbc.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 .sbc-titlebar label {  
    background:none repeat scroll 0 50% # FFFF66;  
    color #7A991A;  
    display:block;  
    font-family: Comic Sans MS;  
    font-size: 15pt;  
}
```

3. Rebuild the Enterprise ARchive (EAR) file.

For more information about rebuilding the EAR, refer to the *Selling and Fulfillment Foundation: Installation 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. You need to place a new file with same name in the following directory:

```
<INSTALL_DIR>/repository/eardata/sbc/war/localization/<locale>/sbc/resources/core/css
```

where <locale> is in the format <language_code>/<country_code>/<variant>.

For example,

```
<INSTALL_DIR>/repository/eardata/sbc/war/localization/en/US/EST/sbc/resources/core/css/sbc-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/sbc/war/localization/en/US/EST/sbc/resources/core/icons/apps_logo.png
```

4. Rebuild the Enterprise ARchive (EAR) file.

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

Localizing Certain Descriptions and Values in Item Administration

Business Center™ can be used by administrator users of an organization to localize certain attributes of item administration. A user may have to create translated descriptions for entities such as items so that item descriptions can appear in different languages for users from different locales. In the case of item descriptions, the translated descriptions are stored in the YFS_ITEM_LOCALE table. In the Business Center user interface (UI), only if more than one locale is configured in the system, fields that can be localized have the localization icon next to them. One of the Related Tasks, **Localize**, can also be clicked to localize the Business Center fields.

For example,  image displays the Short Description field with the localization icon next to it. When a user clicks the  icon, the localization dialog box containing the corresponding base values is displayed. A user can enter the localized values for each locale defined in the system.

The fields listed below the screen names can be localized:

- ◆ Category Management screen
 - ◆ Long Description
 - ◆ Short Description
- ◆ Classification Management screen
 - ◆ Long Description
 - ◆ Short Description
- ◆ Item and Service Primary Information screen
 - ◆ Long Description
 - ◆ Short Description
 - ◆ Extended Description
- ◆ Assets screen for item, service, catalog, category classification domain, classification value, attribute domain, attribute group, attribute, and attribute value
 - ◆ Label
 - ◆ Description
 - ◆ Content Location
 - ◆ Content ID
- ◆ Attribute Group screen
 - ◆ Long Description
 - ◆ Short Description
- ◆ Attribute Domain Details panel
 - ◆ Long Description

- ◆ Short Description
- ◆ Attribute Details screen
 - ◆ Long Description
 - ◆ Short Description
- ◆ Text Attribute Properties screen
 - ◆ Short Description
- ◆ Derived Attribute Ranges screen
 - ◆ Description

Localizing the Sentences in the Coupons and Pricing Rules Screens

Business Center™ enables users to localize the sentences that are displayed in the screens pertaining to coupons and pricing rules. However, the following must be considered when localizing a sentence that is displayed in the user interface (UI):

- ◆ Translation of the literals in a sentence. Each literal has a corresponding bundle key in the `sbcbundle.js` file.
- ◆ Sequence of the components in a sentence. The sequence of the components in a sentence is defined by the language of the corresponding locale such that the sentence is grammatically correct. The corresponding bundle keys are placed in the `sbcsentence_bundle.js` file.

For example, consider the way in which the sequence of components pertaining to a sentence exist for a Combination type coupon under the Conditions tab in the Coupon Details screen:



Label 1 Radio Button Label 2

When this sentence is localized, the order of the labels and the radio button may change for example, **Label 2**, may be followed by **Label 1**, which may in turn be followed by **Radio button**. In such a situation, the correct sequence of this sentence is fetched from the `sbcsentence_bundle.js` file. The bundle entries in the `sbcbundle.js` file corresponding to this example are defined as:

```
"b_OrderContains": "The order contains",  
"b_any": "any",  
"b_all": "all",  
"b_followingItems": "of the following items:"
```

The out-of-the-box sentence bundle entry corresponding to this sentence is as follows:

```
"b_OrderItemKey": "{%sbcOrderContains}{%sbcpanelComb}{%sbcFollItems}"
```

Here, `b_OrderItemKey` is the key and `{%sbcOrderContains}{%sbcpanelComb}{%sbcFollItems}` is the value of this key.

The values in the `sbcsentence_bundle.js` file are the unique identifiers (`sciId` properties) of the sentence components in a particular sequence. The following table lists the `sciId` properties defined for the screen components corresponding to this sentence:

Screen Component	sciId Property
The order contains	sbcOrderContains

Screen Component	sciid Property
Combo box	sbcpanelComb
of the following items:	sbcFollItems

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

To localize this sentence in the language that is specific to a required locale, perform the following tasks:

- ◆ Modify the value of the `b_OrderItemKey` key in the `sbcsentence_bundle.js` file.
- ◆ Translate the values of the `b_OrderContains`, `b_any`, `b_all`, and `b_followingItems` keys in the `sbcbundle.js` file.

Post-Localization Tasks

The localized resource bundles are not automatically used by the Business Center™ application. To ensure that the localized bundles are used by the Business Center application, you must re-create the Selling and Fulfillment Foundation EAR package.

The sections that follow describe the post-localization activities on the server side and client side.

Post-Localization on the Server Side

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

1. Navigate to the <INSTALL DIR>/bin folder.
2. To create the resource jar, run the following command:

For Windows:

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

For Linux/UNIX:

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

3. Rebuild the Enterprise ARchive (EAR) file.

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

Post-Localization 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. To generate localized JavaScript source files from the localized bundle-index file, run the following command:

For Windows:

```
sci_ant.cmd -f jsUtil.xml bundle.map
-Dsourcedir=<INSTALL_DIR>/repository/eardata/sbc/war
-Dindexdir=<INSTALL_DIR>/repository/eardata/sbc/localization_index
-Dindexfile=<INSTALL_DIR>/repository/eardata/sbc/localization_index/
bundle-index_<lang>_<country> -Dwebcontentdepth=0
```

For Linux/UNIX:

```
./sci_ant.sh -f jsUtil.xml bundle.map
-Dsourcedir=<INSTALL_DIR>/repository/eardata/sbc/war
-Dindexdir=<<INSTALL_DIR>/repository/eardata/sbc/localization_index
-Dindexfile=<INSTALL_DIR>/repository/eardata/sbc/localization_index/bundle-
index_<lang>_<country> -Dwebcontentdepth=0
```

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

For Windows:

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

For Linux/UNIX:

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

3. To generate the minified files after localization, run the following command:

For Windows:

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

For Linux/UNIX:

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

4. Rebuild the Enterprise ARchive (EAR) file.

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

Note: Ensure that you run the script in the `bundle.map` and `minify-js` modes for the Web UI Framework files.

For more information about the bundle collector utility, refer to the *Selling and Fulfillment Foundation: Localizing the Web UI Framework*.

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

L

localization
user interface components 5