

Localizing the Application



Contents

Localizing the Application	2
Character Encoding	2
Translating the Interface	2
International Characters	3
Testing Your Translation	3

Localizing the Application

Use this guide to learn about how the application is internationalized, and how to localize it.

Jive SBS is an internationalized application to support localization. Internationalization is the process of changing the format of dates and numbers to obey rules unique to your region and translating user-interface text to your language. For example the number 1,234.00 in an English locale would be formatted as 1.234,00 in the locale for France. Similarly, user-interface text like "Your Name" in English would be translated to "Ihr Name" for German locales.

Character Encoding

The character encoding affects what languages users will be able to post messages in. The default of "ISO-8859-1" is suitable for Western languages, but won't work for Eastern languages such as Chinese or Japanese. For sites that want to support a broad variety of languages, we recommend using "UTF-8" (Unicode) as your character encoding. With the admin console, you can change the encoding used by the application.

Your database must support the character encoding that you choose. Specifically, many database must be specially configured to support Unicode. Although MySQL doesn't normally support Unicode, you can use a JDBC driver workaround by adding the following to the <database> section of jive_startup.xml:

```
<mysql>
  <useUnicode>true</useUnicode>
</mysql>
```

If you're not a MySQL user, consult your database documentation for information about enabling Unicode support in your database.

Note: Settings for mod_jk, the Apache Tomcat redirector module, can cause problems with localized sites. Be sure to use +ForwardURICompatUnparsed rather than +ForwardURICompat. Here's an example:

```
jkOptions +ForwardKeySize +ForwardURICompatUnparsed -ForwardDirectories
```

For more on the connector, see *its documentation* at the Apache web site.

Translating the Interface

All text in the default skin is stored in resource bundles. A resource bundle is a file containing key/value pairs. The default skin loads text using keys and the correct values are retrieved based on locale settings (English values are used for English locales, French values for French locales, etc). Key/value pairs in the English resource bundle might look like the following:

```
skin.yes=Yes
skin.no=No
skin.topic=Topic
skin.message=Message
```

The German resource bundle would contain the same keys, but different values:

```
skin.yes=Ja
skin.no=Nein
skin.topic=Thema
skin.message=Beitrag
```

Making your own translation involves copying the English resource bundle, renaming the file, then translating its contents.

To start, extract the "jive_i18n.properties" file from the sbs-dist.jar (where "dist" is either "employee" or "public," depending on the distribution) file, which is part of the distribution. To extract the file, use the "jar" tool (part of the standard Java distribution):

```
jar -xf sbs-dist.jar jive_i18n.properties
```

Next, you'll need to rename the file to be specific to your locale. For example, the German resource bundle is named "jive_i18n_de.properties" because "de" is the language code for German. For French, the file would be called "jive_i18n_fr.properties". A list of language codes can be found at: <http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>.

International Characters

When translating from English, you may need to use special characters for your language. Unfortunately, all resource bundle files must be saved in ASCII format, which doesn't allow for many international characters. Try working on your translation in a text editor that supports all characters in your language. After finishing your translation, use the `native2ascii` tool (built in to Java) to convert international characters to the ASCII format.

Here's the syntax for the `native2ascii` tool:

```
native2ascii -encoding XXX <input_file> <output_file>
```

Here's an example:

```
native2ascii -encoding XXX my_translation.properties jive_i18n_YY.properties
```

The "-encoding XXX" parameter is optional. If you don't specify it, Java will use the default encoding value, taken from System property "file.encoding". If you do specify an encoding (XXX), it must be taken from the first column of the table of supported encodings in the Supported Encodings document.

Testing Your Translation

To test your translation, the first step is to have a working installation. Save your translated file (example, `jive_i18n_de.properties`) to your webapp `WEB-INF/classes` directory. (If you do not have the application installed as a webapp, your properties file must be in the appserver's classpath.) Next, restart your appserver then log-in to the admin tool. Click the tab "Global Settings" then choose "Locale Settings" from the sidebar. Under the "Change Locale Settings" section, scroll through the list of available locales until you see your language. For example, to test the German resource bundle, you should choose any of the locales starting with "German". Save your settings then load up the default skin. If you see all user interface buttons, column headers, etc, translated to your language then you've correctly loaded your resource bundle. If you still see English text you may have named your bundle incorrectly or your appserver wasn't able to load it.