JasperServer Localization Guide

Version 2.0.1

Table of Contents

1 Introduction

You can localize JasperSoft, including translating JasperSoft BI Suite into a different language by updating its labels and messages. This document describes the process and discusses a few cases that may require further configuration.

Note: The resource bundles described in this document consist of locale-specific ASCII text files (*.properties files).

This document describes steps for creating a locale and configuring JasperSoft to use it.

Creating a locale includes these tasks:

- Translating labels and messages.
- Changing date formats.
- Changing format masks.

Configuring JasperSoft to use a new locale includes these tasks:

- Adding the locale to JasperServer.
- Adding time zones.
- Changing character encoding.
- Working with fonts.

1.1 Documentation Set

JasperServer is described in the following documentation:

- *JasperServer User Guide*. Describes usage and maintenance of the software.
- *JasperServer Build and Developer Guide.* Describes building the source code and developing with JasperSoft.
- *JasperServer Installation Guide*. Describes the installation process that relies on running the executable installation programs. Also provides instructions for deploying a WAR (Web Application Archive) file into your application server.
- *JasperServer Web Services Guide*. Describes the web services and their WSDL to help you integrate JasperServer using XML request and response documents.
- *JasperServer Localization Guide* (this document). Describes the steps you must take to make JasperServer available in a new language and locale: create the locale and configure JasperServer to take advantage of it.

2 UTF-8 Configuration

JasperSoft uses UTF-8 (8-bit Unicode Transformation Format) character encoding. If your database server or application server use a different character encoding form, you may have to configure them to support UTF-8. This section provides information for configuring the character encoding on the Tomcat application server and the MySQL and PostgreSQL databases. If you use a different application server or database, and its default character encoding isn't UTF-8, you may need to make similar updates to support certain locales. For more information, refer to the documentation associated with your software.

2.1 Tomcat

By default, Tomcat uses ISO-8859-1 (ISO Latin 1) character encoding for URIs, which is sufficient for Western European locales, but does not support many locales in other parts of the world.

If you plan to support locales that Latin 1 does not support, you must change Tomcat's URI encoding format.

To configure Tomcat to support UTF-8:

1. Open the conf/server.xml file and locate the section that reads:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector port="8080" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" redirectPort="8443" acceptCount="100"
        connectionTimeout="20000" disableUploadTimeout="true" />
```

2. At the end of this section, insert the following line before the closing marker:

```
URIEncoding="UTF-8" />
```

3. For example, after your changes, the section might read:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector port="8080" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" redirectPort="8443" acceptCount="100"
        connectionTimeout="20000" disableUploadTimeout="true"
        URIEncoding="UTF-8" />
```

4. Save the file.

5. Restart JasperServer.

2.2 MySQL

MySQL uses a default character encoding of ISO-8859-1, referred to as "latin1" in MySQL documentation. JasperServer requires MySQL to use UTF-8 encoding to support the Unicode character set for the repository or report data sources. The simplest way to do this is to create the database with a character set of "utf8". An example of a command for creating a database follows:

```
create database jasperserver character set utf8;
```

In addition, the MySQL JDBC driver requires extra parameter settings to support UTF-8, which are "useUnicode=true" and "characterEncoding=UTF-8". These can be added to the end of the URL. If this is a JNDI data source, such as the repository database, the parameters can be added to the JDBC URL is entered in WEB-INF/context.xml:

```
<Resource name="jdbc/jasperserver" auth="Container"
type="javax.sql.DataSource"

    maxActive="100" maxIdle="30" maxWait="10000"

    username="root" password="password"
    driverClassName="com.mysql.jdbc.Driver"

    url="jdbc:mysql://localhost:3306/jasperserver?
    useUnicode=true&amp;characterEncoding=UTF-8"/>
```

If this is a JDBC data source, the JDBC URL can be changed by editing the datasource in the JasperServer repository. Here is an example of the JDBC URL (note that the ampersand isn't escaped):

```
jdbc:mysql://localhost:3306/foodmart_ja?
useUnicode=true&characterEncoding=UTF-8
```

2.3 PostgreSQL

PostgreSQL's default character encoding is called "SQL_ASCII", but a database can be created with a UTF-8 encoding as follows:

```
create database jasperserver encoding='utf8';
```

PostgreSQL doesn't require any changes to the JDBC URL to support UTF-8.

3 Creating a Locale

Translation is only one aspect of localization. You can also configure format masks and date formats.

The tasks in this section require you to edit these files:

File Name	Location	Purpose of Edits
*.properties files	WEB-INF\bundles	Translating labels and messages
jasperserver_config.properties	WEB-INF\bundles	Changing date formats

3.1 About Properties Files

JasperServer and JasperAnalysis resource bundle files are found in the `WEB-INF/bundles` directory. JasperSoft provides a default locale (for example, jasperserver_messages.properties), which is written in the U.S. English dialect.

A resource bundle consists of the *.properties files that contain all the labels and messages used in the JasperServer and JasperAnalysis, customized for a particular locale. The default resource bundle includes these files:

Component	Default File Name	Description
JasperServer	jasperserver_messages.properties	Core labels and messages used in JasperServer.
JasperServer	jasperserver_config.properties	Configuration properties for dates and date times.
JasperAnalysis	jpivot_messages.properties	Labels and messages used in JasperAnalysis.

iReport and the iReport Plugin have their own resource bundles, which are only installed if you install iReport or the plugin. These files include:

Component	Default File Name	English File Name	Description
iReport	ireport.properties	ireport_en.properties	Labels and messages used in iReport.
iReport plugin	irplugin.properties	irplugin_en.properties	Labels and messages used in the JasperServer iReport plugin.

JasperSoft recommends using JasperBabylon to localize iReport and the iReport plugin resource bundles. For more information, refer to section 6 "JasperBabylon" on page 11.

3.2 Creating a Resource Bundle

3.2.1   Translating Properties Files

Create a resource bundle by copying each *.properties file to a new name. The file name pattern for these files is `<default_file_name>_<locale>.properties`, where <default_file_name> is the name of the default version of the properties file and <locale> is a Java-compliant locale identifier.

For example, consider the core JasperServer resource bundle. For various locales, it might be named:

File Type	File Name
Default resource bundle	jasperserver_messages.properties
English	jasperserver_messages_en.properties
French	jasperserver_messages_fr.properties
French in Switzerland	jasperserver_messages_fr_CH.properties

For a list of Java-compliant locales, please refer to Sun's Java web site.

Note: When working in these files, you must use Unicode characters rather than using the native file format.

The calendar control used throughout the application can be localized by translating the labels from `calendar.properties` to the desired language. The messages are often accompanied by an explanation of their meaning.

To create a new JasperSoft resource bundle:

1. Copy each of the properties files (keeping them in the same directory as the originals) and rename them according to your locale.

2. Translate each *.properties file's labels and messages into the new language.

   Some of the strings in the properties files may not seem like English. These cases are typically date formats and format masks that may need to be edited for the new locale. For more information, refer to section 3.3 "Changing Date and Datetime Formats" on page 6.

3. Save the files.

4. If the new locale requires specific character encoding or fonts, ensure that JasperServer and the third party software it relies on are configured to support them. For more information, refer to section 5 "Advanced Configuration" on page 8.

5. Restart JasperServer, and log into the web application to test your translation. Reviewing the translated strings in context can help you improve your word choices.

Note: This locale will not be available in JasperSoft until you follow the steps described in section 4.1 "Specifying Additional Locale" on page 7.

3.3 Changing Date and Datetime Formats

Each locale may have its own format masks and rules for displaying dates and datetime values. This section describes the steps you must take to update these options for your new locale.

System date and datetime formatting is controlled by four patterns that are specified in the jasperserver_config_<locale>.properties file associated with a particular locale.

For example in the English resource bundle, the four entries are:
```
date.format=dd-MM-yyyy
datetime.format=dd-MM-yyyy HH:mm
calendar.date.format=%d-%m-%Y
calendar.datetime.format=%d-%m-%Y %H:%M
```

The first two keys are used to parse and format dates and datetime values using an internal `java.util.DateFormat` object across the whole application. These patterns should be non-localized date patterns, in accordance with the Java Development Kit (JDK) syntax.

The other two keys are used by the calendar control, which formats the user-selected date and datetime values in accordance with its own pattern syntax.

To change system date and datetime formatting for a new locale, edit the strings specified by these keys.

4 Configuring JasperServer to Offer a Locale

After creating a locale, you must configure JasperSoft to offer it to your users. The tasks in this section require you to edit these files:

File Name	Location	Purpose of Edits
applicationContext-security.xml	WEB-INF	Specifying additional locales
jasperserver-servlet.xml	WEB-INF	Specifying additional time zones

4.1 Specifying Additional Locales

By default, JasperSoft appears in the locale selected in the end-users browser. The login page allows users to specify the locale they want to use in JasperSoft. The list of locales from which they choose is defined in applicationContext-security.xml. Edit this file to add a new locale.

To add a new locale:

1. Edit the applicationContext-security.xml file and locate the bean named `userLocalesList`. For example:

```
<bean id="userLocalesList" class="com.jaspersoft.jasperserver.war.com-
mon.LocalesListImpl">
        <property name="locales">
            <list>
                <value type="java.util.Locale">en</value>
                <value type="java.util.Locale">fr</value>
                <value type="java.util.Locale">it</value>
                <value type="java.util.Locale">de</value>
                <value type="java.util.Locale">ro</value>
                <value type="java.util.Locale">ja</value>
                <value type="java.util.Locale">zh_TW</value>

            </list>
        </property>
    </bean>
```

2. Add the new locale to the end of the list. For example, after you added Dutch (Java's nl_NL locale), the bean would be similar to the following:

```
<bean id="userLocalesList" class="com.jaspersoft.jasperserver.war.com-
mon.LocalesListImpl">
        <property name="locales">
            <list>
                <value type="java.util.Locale">en</value>
                <value type="java.util.Locale">fr</value>
                <value type="java.util.Locale">it</value>
                <value type="java.util.Locale">de</value>
                <value type="java.util.Locale">ja</value>
                <value type="java.util.Locale">zh_TW</value>
                <value type="java.util.Locale">nl_NL/value>

            </list>
        </property>
```

```
</bean>
```

3. Save the file.

4. Restart JasperServer.

For a list of Java-compliant locales, please refer to Sun's Java web site.

4.2 Specifying Additional Time Zones

By default, JasperSoft assumes the user's time zone is the same as the time zone of the computer hosting JasperServer. The login page allows users to specify their time zone. The list of time zones from which they choose is defined in jasperserver-servlet.xml file.

To add a time zone:

1. Open the jasperserver-servlet.xml file and locate the `userTimeZonesList` bean. For example:

```
<bean id="userTimeZonesList" class="com.jaspersoft.jasperserver.war.com-
mon.JdkTimeZonesList">
    <property name="timeZonesIds">
      <list>
        <value>GMT</value>
        <value>PST</value>
        <value>CET</value>
        <value>EET</value>
        <value>GMT+4</value>
      </list>
    </property>
  </bean>
```

2. Add the new time zone to the bottom of the list. JasperSoft recommends specifying each time zone as its GMT (Greenwich Mean Time) off-set. For example, the GMT offset for Pacific Standard Time (PST) is GMT-8.

   For example, after adding Tokyo's time zone (GMT + 9), the bean would be similar to the following:

```
<bean id="userTimeZonesList" class="com.jaspersoft.jasperserver.war.com-
mon.JdkTimeZonesList">
    <property name="timeZonesIds">
      <list>
        <value>GMT</value>
        <value>PST</value>
        <value>CET</value>
        <value>EET</value>
        <value>GMT+4</value>
        <value>GMT+9</value>
      </list>
      </property>
  </bean>
```

3. Save the file.

4. Restart JasperServer.

For more information about Java-complaint time zones, please refer to Sun's Java web site.

5 Advanced Configuration

Depending on the third party software you use and the locales you support, you may also have to configure JasperSoft and its host. The steps described in this section are only necessary under certain circumstances, such as if you plan to use a character encoding form that UTF-8 cannot handle, or if you need to change the font options for JasperAnalysis charts.

The tasks in this section require you to edit these files:

File Name	Location	Purpose of Edits

applicationContext.xml	WEB-INF	Changing character encoding
jpivot_internal_messages.properties	WEB-INF/internal	Specifying chart fonts for JasperAnalysis Open Source
userConfig.xml	WEB-INF/jpivot/print	Embedding fonts in PDF

5.1 Changing Character Encoding

To use a character encoding form other than UTF-8, you must configure JasperServer, your application server, and your database server.

5.1.1 Configuring JasperSoft

To configure JasperSoft to use a different encoding form, you must edit the applicationContext.xml.

1. Open the applicationContext.xml file and locate the following bean:

```
<bean id="encodingProvider" class="com.jaspersoft.jasperserver.api.common.u-
til.StaticCharacterEncodingProvider">
        <constructor-arg value="UTF-8"/>
</bean>
```

2. Change UTF-8 to the encoding type your database server and application server use. For example:

```
<bean id="encodingProvider" class="com.jaspersoft.jasperserver.api.com-
mon.util.StaticCharacterEncodingProvider">
        <constructor-arg value="UTF-16"/>
</bean>
```

3. Save the file.

4. Restart JasperServer.

5.1.2 Configuring the Application Server and Database Server

If you want to use a character encoding form other than UTF-8, you may need to configure the third party software that JasperSoft relies on. For more information, refer to the documentation associated with your application server and database server. For Tomcat, you can specify a different character encoding by following steps similar to those described in section 2 "UTF-8 Configuration" on page 4.

Note: This step is only necessary if you plan to support locales that requires a different character encoding, such as UTF-16. In addition to this change, your application server and database must be configured to use the character encoding you require. For more information, refer to the documentation associated with your third party software.

5.1.3 Configuration for Localized Analysis Schemas

If you plan to use localized analysis views, you must take additional steps to configure JasperServer:

1. Every Unicode database that JasperServer interacts with (whether it is the metadata database for the repository or a database accessed through a data source defined in JasperServer) must be created to support UTF-8. For example, to create the Foodmart database in MySQL, you might give a command similar to:

```
create database foodmart_ja character set utf8;
```

2. The URL of the any OLAP database JasperServer accesses must be properly configured in the /jasperserver/META-INF/context.xml file. For example, the URL definition for the Foodmart sample database might be similar to the following:

```
<Resource name="jdbc/MondrianFoodMart_ja" auth="Container"
type="javax.sql.DataSource"

      maxActive="100" maxIdle="30" maxWait="10000"

      username="root" password="password"
      driverClassName="com.mysql.jdbc.Driver"

      url="jdbc:mysql://localhost:3306/foodmart _ja?
```

```
useUnicode=true&amp;characterEncoding=UTF-8"/>
```

3. Encoding options must be added to the JDBC connection string for any data source that points to an OLAP database. For example, when creating a data source in JasperServer that points to an OLAP database, use the following connection string:

```
jdbc:mysql://localhost:3306/foodmart_ja?
useUnicode=true&characterEncoding=UTF-8
```

5.2 Working with Fonts

While the fonts JasperSoft uses are generally dictated by the JRXML files that define your reports, some font configuration is required for special circumstances. For example, you can configure JasperAnalysis to offer different options in the **Chart Default Font** field in the Chart Options window.

This section describes some of the steps you may need to take, depending on the functionality you use and the locales you support.

Note: In order to use a font in JasperSoft, the font must be available in the host's operating system.

5.2.1   Configuring Options for Chart Default Fonts

If you implement JasperAnalysis and support a locale with special font requirements, you can configure JasperAnalysis to offer different options in the **Chart Default Font** field in the Chart Options window of the analysis view. This may be necessary if you implement locales that Latin 1 doesn't support.

An analysis view's Chart Options window includes the **Chart Default Font** field, which allows users to select the font to use in charts. The default options are SanSerif, Serif, and MonoSpaced. JasperServer reads these values from a properties file and attempts to map them to fonts available in the JasperSoft host's operating system. You can configure JasperSoft to offer different fonts if these fonts don't support the locales you implement.

To change the **Chart Default Font** field's options:

1. Copy the jpivot_internal_messages.properties file, and copy it to a new name that reflects the new locale. For example, for Japanese, the new file would be called jpivot_internal_messages_ja.properties.

2. Open the new file and locate the following keys:
```
jsp.wcf.chart.sansserif=SansSerif
jsp.wcf.chart.serif=Serif
jsp.wcf.chart.monospaced=Monospaced
```

3. Change one or more of the strings to the name of a font available in the host's operating system. For example, if you wanted to change the SansSerif font to the SimHei font, edit the value specified by `jsp.wcf.chart.sansserif`. For example:
```
jsp.wcf.chart.sansserif=SimHei
```

4. Save the file.

5. Restart JasperServer.

5.2.2   Embedding Fonts in PDF Output

Note: By default, JasperServer can create PDF (Portable Document Format) files with many different fonts. However, if you experience font problems in the PDF output of your reports, you may need to take the steps described in this section to make the fonts available to JasperSoft's XSL Formatting Object (XSL-FO) processor.

When users save reports in PDF format, JasperServer generates the PDF output using Apache FOP (Formatting Objects Processor). In order for FOP to render fonts properly, you must install the font itself (for example, a TTF file) on the JasperServer host, create a font metrics file using Apache's `org.apache.fop.fonts.apps.TTFReader` utility), and update the userConfig.xml file to associate the font with its metrics. For more information, please refer to the documentation associated with FOP, which is available at:
http://xmlgraphics.apache.org/fop/0.20.5/embedding.html

You can embed any Unicode font using this procedure, though larger font files may have significantly larger memory footprints. In order to keep memory requirements small, JasperSoft recommends that you use the smallest font file you can, such as SimHei to support Chinese, Japanese, and Korean.

Note: You must have the distribution rights to a font in order to embed it in a PDF file.

6 JasperBabylon

JasperBabylon allows JasperSoft's open source community to edit and share locale resource bundles. The repository is public, and includes translations for several applications (notably, iReport, iReport plugin, and the open source version of JasperServer). You can access the repository by: pointing your browser to the JasperBabylon web site: http://www.jasperforge.org/jasperbabylon

More information, including usage instructions, is available on the JasperBabylon web site. To make updates on this site, you must sign up for a contributor ID, which is free.

7 Contacting JasperSoft

If you have the evaluation license and would like to purchase the commercial license, please contact JasperSoft Sales at 415-348-2319.

If you have purchased JasperSoft Professional, and have questions, suggestions, or problems, please contact JasperSoft Technical Support:

Phone: 415-677-2245
Toll-free Phone (within the U.S.): 877-600-5767
Email: support@jaspersoft.com
Web site: http://www.jaspersoft.com/ss_overview.html

JasperSoft encourages all JasperSoft users to participate in the forums we host at:
http://www.jasperforge.org/