



Localization Guide

Release 7.5 SP1

May 2006



Copyright Notice

Localization Guide, Release 7.5 SP1

Copyright © 2002 - 2006
Yantra Corporation
ALL RIGHTS RESERVED

WARNING: ANY UNAUTHORIZED DUPLICATION OF THIS DOCUMENTATION SHALL BE AN INFRINGEMENT OF COPYRIGHT

Trade Secret Notice

This documentation, the software it describes, and the information and know-how they contain constitute the unpublished, proprietary, confidential and valuable trade secret information of Yantra Corporation, its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Yantra entity. This documentation and the software it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Yantra Corporation copyright notice.

This documentation and the software it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Yantra license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Yantra entity reserves the right to revise this documentation from time to time and to make changes in content hereof without the obligation to notify any person or entity of such revisions or changes.

Yantra does not warrant or represent that use of the software described in this documentation will insure compliance with the U.S. Department of Transportation's (DOT) Hazardous Materials Regulations (HMR) found in Title 49 of the Code of Federal Regulations (49 CFR), and users of this software should consult independent legal counsel and technical support to insure compliance with the HMR and other legal requirements.

U.S. Government Restricted Rights. This documentation and the software it describes are each and collectively a "commercial item" as defined in 48 C.F.R. 2.101. Consequently, as and when provided to any agency or instrumentality of the U.S. Government or to a U.S. Government prime contractor or a subcontractor at any tier ("Government Licensee"), the terms and conditions of the customary Yantra commercial license agreement are imposed on Government Licensees per 48 C.F.R. 12.212 or § 227.7201, as applicable, or through 48 C.F.R. § 52.244-6.

Yantra and the Yantra logo are marks of Yantra Corporation. All other services and products or company names are used for identification purposes only and may be marks of their respective owners.

Yantra Corporation
One Park West
Tewksbury, MA 01876
1-978-513-6000

Copyright © 2002 -
2006

Contents

Preface

Intended Audience	v
Structure	v
Yantra 7x Documentation	vi
Conventions	vii

1 Internationalization in Yantra 7x

1.1 Localizable Components and Data	1
1.1.1 Literals and Data	2
1.1.2 Multi-Byte Character Sets	2
1.1.3 Date and Time Formats	3
1.1.4 Time Zones	3
1.1.5 Numeric Formats	4
1.1.6 Currency	4
1.1.6.1 Currency Precision	5
1.1.6.2 Currency Conversion	5
1.1.6.3 Currency Conversion Scenario	6
1.1.7 Units of Measure	7
1.2 Non-Localizable Components and Data	7
1.3 Localizing Yantra 7x using Single or Multiple Language Packs	7

2 Localizing Yantra 7x

2.1 Before You Begin	11
2.2 Setting Up Character Encoding	13

2.2.1	Specifying the JRE Settings.....	13
2.2.2	Latin-1 Character Encoding.....	13
2.2.3	UTF-8 Character Encoding.....	14
2.3	Database Overview.....	15
2.3.1	Oracle Database Setup.....	16
2.3.2	SQL Server Database Setup.....	16
2.3.3	DB2 Database Setup.....	17
2.4	Data.....	17
2.4.1	Localization for a Multi-Language Installation.....	17
2.4.1.1	Extracting Error Cause and Error Action for Localization.....	19
2.4.1.2	Extending Default Factory-Shipped Translations.....	20
2.4.1.3	Switching the Yantra Base Language.....	21
2.4.1.4	Full Export to Back-up Existing Localization Literals.....	24
2.4.1.5	Deleting Unused Localization Literals.....	24
2.5	Directory Structure and Filenames.....	25
2.6	User Interface Themes.....	25
2.6.1	Themes.....	26
2.7	Literals.....	27
2.7.1	Resource Bundles.....	28
2.8	Date and Number Validation.....	31
2.8.1	Date and Time Validation.....	31
2.8.2	Number Validation.....	32
2.9	Templates.....	33
2.9.1	E-mail Templates.....	33
2.9.2	Exception Alert Templates.....	33
2.10	Calendar and Message Popups in the Console.....	34
2.11	Images.....	35
2.11.1	Application Consoles Images.....	35
2.11.2	Configurator Images.....	35

A Localizable XML Attributes

Index

Preface

This manual provides an outline for localizing the Yantra 7x application suite.

Intended Audience

This manual is intended to provide a general localization procedure for users responsible for localizing the Yantra 7x application suite for new languages, as well as languages that are provided by Yantra 7x. Prior to reading this document, be sure that you have read the *Yantra 7x Installation Guide* for knowledge and understanding of the Yantra 7x database installation.

Structure

This manual contains the following sections:

Chapter 1, "Internationalization in Yantra 7x"

This chapter briefly lists and describes how the Yantra 7x application has been internationalized for localization and lists the components that cannot be localized.

Chapter 2, "Localizing Yantra 7x"

This chapter describes the Yantra 7x components that can be localized and provides general guidelines on how to localize each.

Appendix A, "Localizable XML Attributes"

This chapter lists the localizable factory setup XML attributes.

Yantra 7x Documentation

For more information about the Yantra[®] 7x components, see the following manuals in the Yantra[®] 7x documentation set:

- *Yantra[®] 7x Release Notes*
- *Yantra[®] 7x Installation Guide*
- *Yantra[®] 7x Upgrade Guide*
- *Yantra[®] 7x Performance Management Guide*
- *Yantra[®] 7x High Availability Guide*
- *Yantra[®] 7x System Management Guide*
- *Yantra[®] 7x Localization Guide*
- *Yantra[®] 7x Customization Guide*
- *Yantra[®] 7x Integration Guide*
- *Yantra[®] 7x Product Concepts*
- *Yantra[®] 7x Warehouse Management System Concepts Guide*
- *Yantra[®] 7x Platform Configuration Guide*
- *Yantra[®] 7x Distributed Order Management Configuration Guide*
- *Yantra[®] 7x Supply Collaboration Configuration Guide*
- *Yantra[®] 7x Inventory Synchronization Configuration Guide*
- *Yantra[®] 7x Product Management Configuration Guide*
- *Yantra[®] 7x Logistics Management Configuration Guide*
- *Yantra[®] 7x Reverse Logistics Configuration Guide*
- *Yantra[®] 7x Warehouse Management System Configuration Guide*
- *Yantra[®] 7x Platform User Guide*
- *Yantra[®] 7x Distributed Order Management User Guide*
- *Yantra[®] 7x Supply Collaboration User Guide*
- *Yantra[®] 7x Inventory Synchronization User Guide*
- *Yantra[®] 7x Logistics Management User Guide*
- *Yantra[®] 7x Reverse Logistics User Guide*

- *Yantra® 7x Warehouse Management System User Guide*
- *Yantra® 7x Mobile Application User Guide*
- *Yantra® 7x Analytics Guide*
- *Yantra® 7x Javadocs*
- *Yantra® 7x Glossary*
- *Yantra® 7x Carrier Server Guide*
- *Yantra® 7x Application Server Installation Guide* (for optional component)

Conventions

The following conventions may be used in this manual:

Convention	Meaning
. . .	An ellipsis represents information that has been omitted.
< >	Angle brackets indicate user-supplied input.
mono-spaced text	Mono-spaced text indicates a file name, an API name, or a code example.
/ or \	Slashes and backslashes are file separators for Windows, UNIX and LINUX operating systems. The file separator for the Windows operating system is "\" and the file separator for Unix and Linux systems is "/". The Unix convention is used unless otherwise mentioned.

Internationalization in Yantra 7x

This chapter provides an overview of concepts regarding localization and internationalization. It also explains how they apply to the Yantra 7x application.

1.1 Localizable Components and Data

This section describes the internationalized components and data that can be localized in your system.

Every user created within Yantra 7x can have an associated preference for number formatting, date layout, and language. These preferences are called a locale and are identified as a pairing of a language code and a country code. Examples include `en_US` (English, US), `fr_CA` (French, Canada), `fr_FR` (French, France). A specific locale definition includes the following information:

- Country
- Language
- Date and time format
- Time zones
- Numeric Formats
- Currency
- Unit of measure for dimensions, volume, and weight

To associate data such as dates, times, and strings with locale-specific formats, Yantra 7x associates a locale with each user profile. This allows each user to interact with a locale-specific version of the product.

The locale definition associated with any organization defined in Yantra 7x is used to determine currency and unit of measure only. Date, time, and time zone information is strictly related to each user.

1.1.1 Literals and Data

All user interface and exception message literals are retrieved from a set of external files or database tables.

The Yantra 7x application retrieves images and literals from locale-specific files. In order to provide a single installation, multi-lingual solution, Yantra 7x stores multiple instances of the literals for a screen. Each instance is identified by a specific country and language pairing.

1.1.2 Multi-Byte Character Sets

Multi-byte character sets are appropriately and thoroughly taken into consideration in the database, application server, and browser tiers of the Yantra 7x application. To represent all of the characters in a language it is sometimes necessary to use 2 (double-byte) or 3 (multi-byte) bytes for each character. The longer character representations can represent a space and transmission challenge for application development.

Double-byte Character Set (DBCS): One of a number of character sets defined for representing Chinese, Japanese, or Korean text (for example, JIS X 0208-1990). These character sets are often encoded in such a way as to allow double-byte character encoding to be mixed with single-byte character encoding.

Multi-byte Character Set (MBCS): A character set encoded with a variable number of bytes for each character. Many large character sets have been defined as multi-byte character set in order to keep strict compatibility with the standards of the ASCII subset, the ISO and IEC 2022.

Note: In the CUI Mobile Terminal, if you localize DB and MB character set, the characters will not be clearly visible.

The Yantra 7x application architecture ensures that:

- All data is stored in the database using a standard compression algorithm known as UTF-8.

- The application is coded in Java which can handle multi-byte character sets without any special changes.
- All communication between the database and the application server is through Java Database Connectivity (JDBC) which transforms the UTF-8 database representation of data to and from the multi-byte character set.
- All communication between the application server and the client is through UTF-8, which minimizes data transmission volume.
- All clients are expected to receive and send data using the UTF-8 algorithm.

1.1.3 Date and Time Formats

The Yantra 7x application can present stored dates and times in any valid date or time format. Date and time fields in Yantra 7x are expected to be entered relative to the locale where the Yantra 7x database resides. Some typical date formats are as follows:

- MM/dd/yyyy
- dd/MM/yyyy
- yyyy/MM/dd

Note: In each date format, the month can be expressed as a word instead of a numeral as long as the entire date does not exceed ten characters in length.

Some typical time formats are as follows:

- HH:mm:ss
- HH:mm

1.1.4 Time Zones

Besides being sensitive to local time zone considerations, Yantra 7x is aware of worldwide time zones. For example, if an order is entered in Germany for fulfillment in the United States and the order is not filled in time, the software considers the differences in time zones in order to raise an exception at the appropriate hour in the United States.

The following characteristics guarantee sensitivity to time zones:

- When a date and time value is stored, it is converted to the locale of the database. For example, assume that a database is in New York and a customer service representative is in London. The customer service representative enters an order. When the date and time of the order is stored in the database, which resides in New York, the values are converted to Eastern Standard Time.
- When a date and time field is displayed through the user interface, Yantra 7x performs time zone calculations based on the current locale of the user and presents the time accordingly. For example, when a customer service representative in London displays an order, which resides in a New York database, the date and time are converted from Eastern Standard Time to Greenwich Mean Time.

Note: When a date field does not contain a time component, the time is assumed to be 12 AM. Such fields are not adjusted for time zone when viewed from various locales with different time zones.

- Yantra 7x APIs output the date and time as stored in the database. For time-sensitive fields the time zone difference from the Universal Time Coordinate (UTC) is appended to the date and time in the output.

1.1.5 Numeric Formats

Numeric formats are dependent on the country and language set up in the locale definition.

1.1.6 Currency

Yantra 7x allows each order to be processed in the preferred currency of the customer. This currency is known as the "transactional currency".

A currency is also associated with the locale that is associated with each enterprise. This currency is known as the "enterprise currency".

Order management capabilities provide order handling in multiple currencies between buyer and seller. These capabilities include a

multi-currency view of an order's value as well as currency conversion procedures and rate tables.

Data structures hold flexible charge and tax taxonomies for order and invoice entities. Yantra 7x has deliberately *not* taken the approach of building complex taxation rules into the application. Integration with sophisticated tax calculation programs, such as Taxware or Vertex, complement our solution and provide you with a complete taxation system. Yantra 7x optionally provides a standard integration to the Taxware product line.

Note: All charges and taxes display in the user interface in the order (transactional) currency. Users can switch to view payment information of an order in either the transactional or enterprise currency.

1.1.6.1 Currency Precision

The Yantra 7x application has the following currency precision characteristics:

- Unit price and unit cost are entered and stored to a maximum of six decimal places.
- Totals are entered and stored to a maximum of two decimal places.
- Totals are rounded with traditional rounding (round up any digit of 5 or more).

1.1.6.2 Currency Conversion

Yantra 7x has the following currency conversion characteristics:

- Currency conversion rates are defined between two currencies and are bound by effective dates. For example:

Table 1–1 Currency Conversion Rates

From	To	From	To	Rate
British Pound	US Dollar	1/1/2004	1/31/2004	1.51
British Pound	US Dollar	2/1/2004	2/28/2004	1.56
British Pound	US Dollar	3/1/2004	3/31/2004	1.62

A conversion rate definition is understood to imply a 1:x relationship. Thus, if one British Pound is equal to 1.51 US dollars during January 2004, two British pounds are equal to 3.02 US dollars during January 2004.

- Rate definitions are not reciprocal. A conversion from US dollars to British pounds cannot use the inverse of the British pound to US dollar exchange rate. A rate for converting US dollars to British pounds must be available or Yantra 7x reports an error. The only exception to this restriction is that a reciprocal relation does exist between the euro and its member currencies.
- Yantra 7x provides the `updateConversionRates` API that allows users to import exchange rates and maintain the exchange rates in their database. Using this API, new exchange rates can be loaded every few hours if desired. You can also set up conversions using the Yantra 7x Configurator.
- Yantra 7x provides a user exit at the point of performing a currency conversion. This user exit allows users to perform currency conversions outside of Yantra 7x. This allows real-time application of the current exchange rate instead of using the last updated exchange rate.
- Conversion rates can be entered and maintained in the Yantra 7x Configurator.

1.1.6.3 Currency Conversion Scenario

The following scenario illustrates how currency is displayed for an order. Suppose that a customer enters an order in France with an English company. The customer pays in francs (the transactional currency). Yantra 7x converts the amount from francs to pounds using triangulation through the euro.

Because the order is placed in francs, if a company employee in England displays the order, the order is displayed in francs.

To allow auditing or reviewing of currency conversions, Yantra 7x stores the exchange rates for each order as an order attribute. When an order is entered, the software stores the exchange rate used between the transactional currency and the enterprise currency. If a conversion involves triangulation, Yantra 7x stores the composite exchange rate

used between the euro, the transactional currency, and the enterprise currency.

1.1.7 Units of Measure

Yantra 7x users can select different units of measure (UOM) from various places in the Yantra 7x User Interface.

For supported carriers, pack dimensions entered by a shipper can be converted to different dimensions needed by the carrier. For example, if a shipper enters pack dimensions in kilos, kilos can be converted to pounds for a carrier. To accomplish this conversion for other carriers, custom coding is needed.

1.2 Non-Localizable Components and Data

Yantra does not support the localization of components that are explicitly technical in nature. These components are typically used by professionals who are working with a Yantra employee to perform installation, tuning tasks, and so forth. The components include documentation and the following tools:

- Configuration Deployment Tool
- Portlet Generator

1.3 Localizing Yantra 7x using Single or Multiple Language Packs

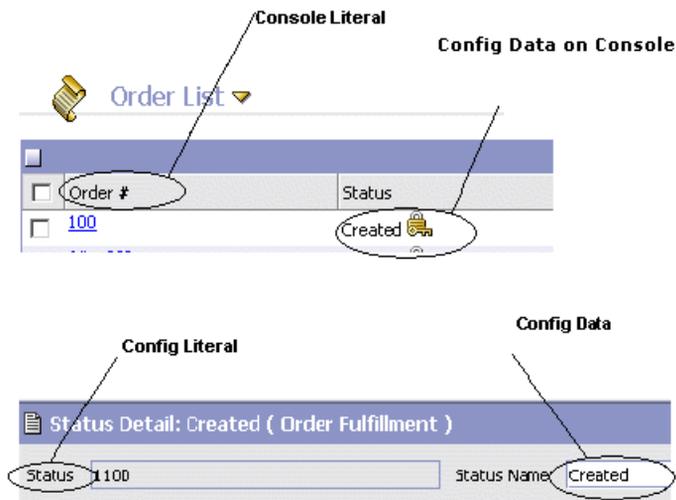
You can localize Yantra 7x in a single base language with one or multiple language packs. The base language refers to the display language of the factory setup data in the configurator. For more information on switching the base language refer to [Section 2.4.1.3, "Switching the Yantra Base Language"](#).

Note: The languages listed here are for informational purposes only. For actual available languages please contact Yantra Corporation's sales division at 1-978-513-6000.

The following list provides a description of the terms used in the tables.

- Base Language - Refers to the language for displaying factory setup data in the Yantra 7x Configurator.
- Config Literals - Refers to the labels on the Yantra 7x Configurator screen.
- Console Literals - Refers to the labels on the Yantra 7x Application Consoles screen.
- Config Data - Refers to the factory setup data. A specific list of fields can be translated. For example, the status description can be localized.
- Config data on Console - Refers to the display of the config data on the console user interface. For example, Status description is config data that appears on the console screen.

A schematic diagram with the above discussed terms is given below:



The following sections provide more information on the localized literals and data if you have purchased single language CD or multiple language CDs.

Single Language CD

The [Table 1–2, "Localization Options for Single Language CD"](#) provides two different options for the localization information if you get a single language pack along with a standard base language.

Table 1–2 Localization Options for Single Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 1: Base Language English with Japanese Language Pack	Japanese	Japanese	English	Japanese	Japanese
	English	English	English	English	English
Option 2: Base Language Japanese with English Language Pack	Japanese	Japanese	Japanese	Japanese	Japanese
	English	English	Japanese	English	English

Multiple Language CDs

Each language pack is shipped as a separate CD, if you have purchased a Japanese and Chinese language packs you would have 2 language CDs.

The [Table 1–3, "Localization Options for Multiple Language CD"](#) provides three different options for the localization information if you get multiple language packs along with a standard base language.

Note: Each of the below specified option is supported as part of your implementation design. However, you need to decide the best possible option for your business needs.

Table 1–3 Localization Options for Multiple Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 1: Base Language Japanese with Chinese and English Language Pack	Japanese	Japanese	Japanese	Japanese	Japanese
	English	English	Japanese	English	English
	Chinese	Chinese	Japanese	Chinese	Chinese

Table 1–3 Localization Options for Multiple Language CD

Options	User Locale	Config Literals	Config Data	Console Literals	Config Data on Console
Option 2: Base Language Chinese with Japanese and English Language Pack	Japanese	Japanese	Chinese	Japanese	Japanese
	English	English	Chinese	English	English
	Chinese	Chinese	Chinese	Chinese	Chinese
Option 3: Base Language English with Japanese and Chinese Language Pack	Japanese	Japanese	English	Japanese	Japanese
	English	English	English	English	English
	Chinese	Chinese	English	Chinese	Chinese

Localizing Yantra 7x

This chapter provides steps for localizing the Yantra 7x application for both languages that are not yet provided and languages that are supported by Yantra 7x out of the box.

2.1 Before You Begin

Before you begin localizing Yantra 7x, keep the information in this section in mind.

Rules Governing Localization

Some of the localization rules that Yantra 7x follows are governed by standards and rules external to Yantra 7x, such as the Java JVM, JRE, operating systems, and external applications. The Java programming language specifies the implementation of the locale logic. For example, see

<http://java.sun.com/j2se/1.4.2/docs/api/java/util/ResourceBundle.html>.

For more information on the logic around resource bundles. For information on specifying a Java JRE, see [Chapter 2.2.1, "Specifying the JRE Settings"](#).

Note: If possible, Yantra recommends that an exact match is used for all localization files (`_<lang>_<country>` in the file name), to ensure that all files are correctly returned.

International Standards Organization (ISO) Codes

Yantra 7x uses locale-related codes as specified by the ISO. Throughout this guide, locale is represented by `<language>_<country>`. Locale is

comprised of <language>, a 2-character lower-case ISO-639 code, and <country>, a 2-character upper-case ISO-3166 code.

Character Encoding

When planning the localization of your Yantra 7x application, ensure that your operating system default character encoding is set to the one specified by the `yfs.ui.defaultEncoding` property in the `yfs.properties` file during product installation.

When deploying on WebSphere, in order to enable entering any characters other than the ASCII set (for example Asian characters and vowels with accents), ensure that the WebSphere JVM `client.encoding.override` property is set to UTF-8. For more information about required WebSphere JVM settings, see the *Yantra 7x Installation Guide*.

Literals External to Yantra 7x

External applications used in conjunction with Yantra 7x may display on-screen information. For example, WebLogic displays a status message during startup. Non-Yantra 7x literals of this type may need to be localized. For instructions on how to localize these types of messages, see the documentation supplied by that software provider.

Windows Clients

When localizing the Configurator to run on Windows, ensure that client computers have regional options set appropriately within the Windows Control Panel.

For example, if the client computer is running Windows and you want to display the title "Yantra 7x Configurator" in Japanese, the client Windows regional options must be set to Japanese.

When localizing the Yantra 7x Application Consoles to run on Windows, ensure that client computers can correctly display Unicode characters by configuring the display appearance within Control Panel > Display > Appearance.

Mobile Device Clients

The `ycpapibundle.properties` file contains literals that apply specifically to mobile devices. The key for these mobile device literals are

preceded with "Mobile_". When localizing Yantra 7x to run on a mobile device, localize the literals associated with these keys.

2.2 Setting Up Character Encoding

For your implementation of Yantra 7x, use the guidelines specified within this section when using Latin-1 and UTF-8 encoding.

2.2.1 Specifying the JRE Settings

In a multi-language deployment, install the international version of the JRE onto each client computer.

To install the plugin, download the JRE from <http://java.sun.com/j2se/1.4.2/download.html> and install it on each client computer. For clients that must display non-English literals, choose the "Windows (all languages, including English)" JRE.

2.2.2 Latin-1 Character Encoding

ISO 8859-1 is the ISO standard Latin-1 character set and encoding format. CP1252 is what Microsoft defined as the superset of ISO 8859-1, so there are approximately 27 extra characters that are not included in the standard ISO 8859-1.

When using Latin-1 character encoding, Yantra strongly recommends that you also refer to the following documents:

- For information on "Basic Latin" and the "Latin-1 Supplement" that together comprise the "Latin-1" (or ISO 8859-1) character set and encoding, see <http://www.unicode.org/charts/>.
- For an official chart download, see [http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28245&ICS1=35&ICS2=40&ICS3= \(\)](http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28245&ICS1=35&ICS2=40&ICS3= ()).
- For a Microsoft site that lists both ISO 8859-1 and CP1252, see <http://www.microsoft.com/globaldev/reference/WinCP.msp>.

2.2.3 UTF-8 Character Encoding

Encoding must be specified in order for international characters to display correctly on the following UI components:

- Inventory Graph
- Delivery Map
- Shipnode in Inventory Console
- Item ID in Inventory Console

On Unix, the UTF-8 character encoding must be specified on your Unix application server.

To set up a UTF-8 environment:

1. Modify all process startup scripts (for the application server, time-triggered transactions, and so forth) to include the `-Dfile.encoding=UTF-8` parameter for the Java command.
-OR-
2. On the Unix application server, determine whether or not it has UTF-8 capability by running the following command:

```
locale -a
```

- If the command returns *any* line that indicates UTF-8, go to [Step 1](#).

For example:

```
POSIX
common
en_US.UTF-8
C
iso_8859_1
```

- If the command returns no lines that indicate UTF-8, go to [Step 3](#).
3. From the international language option pack appropriate to your Unix operating system, install at least one language that has the UTF-8 character set, then return to [Step 2](#) to test its correct installation.

2.3 Database Overview

Data Encoding Format

Yantra 7x is tested and shipped using the UTF-8 transformation format. If you use a different transformation encoding format, the number of characters that you can store in standard database sizes diminishes. In this case, you must ensure that you review and modify the Yantra 7x database creation process to size the database fields accordingly.

Character Set

Use a character set appropriate for your desired localization language. For example, single-byte language character sets typically require UTF-8, while a multi-byte language may require a UTF-16 character set.

The character set you choose may impact field sizes. For example, a Varchar(40) field can only store 40/3 Japanese characters using the UTF-8 character set. This has implications on the table field sizes at the time of creation. Table creation scripts must be changed to ensure that the field lengths are correct.

When deciding the appropriate character set to use for your desired language, see the documentation specific to your database server.

Factory Defaults

Factory defaults are limited to one language. This means that in order to switch from one language to another, you must load new factory defaults to your database.

The UI literals are not part of the factory defaults. They can be switched from one language to another as needed, as long as they have been translated appropriately.

When using an Oracle database, it is possible to use Japanese characters in English factory defaults. In fact, it could be any character in valid Unicode and UTF-8 range. When using a SQL Server database, it is not possible to use more than one encoding and code page, based on the database collation chosen at time of the database creation. Therefore, if the database is created with a collation that is not Japanese, the factory defaults cannot contain Japanese character collation, such as Latin1 (SQL_Latin1_General_CP1_CI_AS).

2.3.1 Oracle Database Setup

In this release of Yantra 7x, multi-lingual (including a multi-byte character set) environments with Oracle databases have been tested and certified using Oracle database created with a UTF-8 character set. See the *Yantra 7x Installation Guide* for more information about installing Oracle and the settings for the UTF-8 character set.

2.3.2 SQL Server Database Setup

SQL Server is limited to the collation and code page with which the database is created to store characters. Because of this, during database creation, you must carefully consider collation issues regarding storage of data with non-English international characters, including supported Asian code pages.

Yantra recommends that SQL Server requires case-insensitive collation (specified using the CI argument). For example, to choose a case-insensitive collation for the Japanese language, one of the valid collation choices is Japanese_CI_AI.

The localizable factory setup is limited to the selected collation and code page as well, since they are stored in database.

SQL Server permits one locale for each database.

To set up SQL Server:

1. From the SQL Server collation name drop-down menu, select the collation name that supports the language you want to specify. Be sure to select a collation that is case insensitive. This creates your database.
2. Edit the `yfs.properties` file and specify the `charset` value for the JDBC URL property to match the code page of the collation name selected in SQL Server as shown below:

Jdbc url format is

```
"jdbc:inetdae7: <hostName>?charset=<yourCharset>"
```

3. Install the application tables and other components, and then load the database factory defaults as described in the *Yantra 7x Installation Guide*.

2.3.3 DB2 Database Setup

DB2 database can be localized by passing the codeset as UTF-8. For more information on DB2 database setup see the *Yantra 7x Installation Guide*.

2.4 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 need to store values in a language-specific format. If these database literals are not localized, screen literals display inconsistently, with some displaying in the localized language and others displaying in English.

You can store item descriptions in your database in multiple languages. For more information on item descriptions see the *Yantra 7x Product Management Configuration Guide*.

Note: The localized item description is only available for Order Details and Order Line Details screens.

2.4.1 Localization for a Multi-Language Installation

The database factory default values can be localized for a multi-language installation, to use one or more locales in addition to the installed locale.

A list of localizable factory setup defaults

To localize factory default XML and user-configured attributes for multiple locales:

1. Create the necessary locale using the Yantra 7x Configurator. For more information about creating a locale, see the *Yantra 7x Platform Configuration Guide*.
2. In the `<YFS_HOME>/bin` directory, configure the required parameters in your `yantra.properties.sample` file. The parameters to be configured include:

```
YFS_HOME=<directory>
DB_DRIVER_CLASS=<JDBC driver class>
DB_URL=<url>
DB_USER=<User ID>
```

```
DB_PASSWORD=<Password>
ANT_OPTS=-ms96m -mx512m
log4j.configuration=/resources/log4jconfig.xml
```

Save the modified file as `yantra.properties` in the same directory.

3. Run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in EXPORT mode as follows:

```
ant -f localizedstringreconciler.xml export -Ddestdir=<YFS_
HOME>/database/FactorySetup/XMLS
```

This processes the entity XMLs and identifies the missing literals for the localizable columns in the `YFS_LOCALIZED_STRINGS` table.

The data is then exported to one or more `properties` files depending on the number of locales present in the `YFS_LOCALE` table.

To export for a specific locale, use optional parameters `language`, and `country`. This usage is as follows:

```
ant -f localizedstringreconciler.xml export -Ddestdir=<YFS_
HOME>/database/FactorySetup/XMLS -Dlanguage=fr -Dcountry=FR
```

Note: In an instance where the `YFS_LOCALE` table contains two entries other than the base locale, two files are created.

For example, if `fr_FR` (representing French, and France) is the locale, a file named `en_US_ycpmissinglocalizedstrings_fr_FR.properties` is created in the destination folder specified (`destdir`).

4. Edit the `aa_BB_ycpmissinglocalizedstrings_xx_YY.properties` file for the relevant locale, in the `<YFS_HOME>/database/FactorySetup/XMLS/<xx_YY>` directory, where `aa` is the language code for the 'from' locale, `BB` is the country code for the 'from' locale, `xx` is the language code for the 'to' locale, and `YY` is the country code for the 'to' locale.

The `aa_BB_ycpmissinglocalizedstrings_xx_YY.properties` file, contains entries in the following format:

```
Acceptance_Process=
```

```
Add_Service=
Line=
```

Add relevant transaction values to the entries. An example (for French literals) is:

```
Acceptance_Process=Processus d'acceptation
Add_Service=Ajouter service
Line=Ligne
```

5. Save the modified file as `aa_BB_ycplocalizedstrings_xx_YY.properties`, where `aa` is the language code for the 'from' locale, `BB` is the country code for the 'from' locale, `xx` is the language code for the 'to' locale, and `YY` is the country code for the 'to' locale.
6. Run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in `IMPORT` mode as follows:

```
ant -f localizedstringreconciler.xml import -Dsrc=<YFS_
HOME>/database/FactorySetup/XMLS
```

This inserts the values specified in the `properties` file into the database.

2.4.1.1 Extracting Error Cause and Error Action for Localization

The `export` mode does not extract error cause and error action literals. The error cause and action can be extracted from the database.

To extract error cause and error action literals from the database for **localization**:

1. In the `<YFS_HOME>/bin` directory, configure the required parameters in your `yantra.properties.sample` file. The parameters to be configured include:

```
YFS_HOME=<directory>
DB_DRIVER_CLASS=<JDBC driver class>
DB_URL=<url>
DB_USER=<User ID>
DB_PASSWORD=<Password>
ANT_OPTS=-ms96m -mx512m
log4j.configuration=/resources/log4jconfig.xml
```

2. Save the modified file as `yantra.properties` in the same directory.

3. Run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in `EXTRACT_ERRORCAUSE_DB` mode as follows:

```
ant -f localizedstringreconciler.xml extract_errorcause_
db-Ddestdir=<YFS_HOME>/database/FactorySetup/XMLS
```

4. Provide the location of the file and filename as follows:

```
database/FactorySetup/XMLS/nonlocalized_<Country>_
<Language>
ycplocalizedstrings_<Country>_<Language>.properties
```

5. Localize the file to the desired locale.
6. Save the modified file as `aa_BB_ycplocalizedstrings_xx_YY.properties`, where `aa` is the language code for the 'from' locale, `BB` is the country code for the 'from' locale, `xx` is the language code for the 'to' locale, and `YY` is the country code for the 'to' locale.
7. Run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in `IMPORT` mode as follows:

```
ant -f localizedstringreconciler.xml import -Dsrc=<YFS_
HOME>/database/FactorySetup/XMLS
```

This inserts the values specified in the `properties` file into the database.

2.4.1.2 Extending Default Factory-Shipped Translations

The factory default translations can be extended to create custom translations for the localization literals.

To modify the default factory-shipped translations with custom localization literals:

1. Create a new `extn` directory in the `<YFS_HOME>/database/FactorySetup/XMLS/<language>_<country>` directory.

Note: For example, if `fr_FR` is the factory-shipped translated locale, the `extn` directory should be created in the `<YFS_HOME>/database/FactorySetup/XMLS/fr_FR` directory.

2. Copy the `<baselanguage>_<basecountry>_ycplocalizedstrings_<language>_<country>.properties` file from the `<YFS_HOME>/database/FactorySetup/XMLS/<language>_<country>` directory to the newly created `extn` folder.
3. Edit the translations in the `properties` file in the `extn` directory:
 - Modify the translations for existing localization literals with the new translations.
 - Add new localization literals and their translations, if needed.
 - Remove any obsolete or unwanted translations that are not overridden for localization literals. This enables you to retain the Yantra 7x provided translations during an upgrade.

Note: It is recommended that you manage the extended translations during an upgrade.

4. Run the `LocalizedStringReconciler` tool in `IMPORT` mode as follows:

```
ant -f localizedstringreconciler.xml import -Dsrc=<YFS_HOME>/database/FactorySetup/XMLS
```

This tool first inserts the values specified in the `properties` file present in the `<YFS_HOME>/database/FactorySetup/XMLS/<language>_<country>` directory into the database.

This entry is then replaced with the values specified in the `properties` file in the `<YFS_HOME>/database/FactorySetup/XMLS/<language>_<country>/extn` directory.

2.4.1.3 Switching the Yantra Base Language

The base language is the language that all translations are in relation to. For example, when a new common code is added, the description is in the base language, and there are translations from this language to other languages.

Note: The base language can only be switched **once**. Switching a base language more than once may result in a loss of data or other potential errors.

By performing this switch the following processes occur:

- Every translation to the desired base language from your current base language is changed to a translation from your desired base language to your current base language.
- Every translation from your current base language to another language is changed to a translation from your desired base language to another language.
- Every localizable column is converted from your current base language to your desired base language.

For example, your Yantra install has English as a base language, and you have setup translations to French and German. The translations are interpreted as follows:

- Translations from English to French
- Translations from English to German

By switching your base language to French, your translations will be interpreted as follows:

- Translations from French to English
- Translations from French to German

Note: In order to successfully perform a base language switch, all localizable descriptions from the current base language to the desired base language must exist. Furthermore, there must be at least one entry in the YFS_LOCALE table corresponding to the desired base language.

Considerations When Switching the Base Language

The following points need to be taken into consideration prior to performing a base language switch.

- Multiple 'from' strings in a language may translate to the same 'to' string in another language. During the switch, only one of these records will be retained, and ambiguous records will be removed.

For example, ABC and AAC in English may both translate to FABC in French. When the base language is switched from English to French, only one record with a 'from' string of FABC will be retained.

- If a locale does not provide a translation, Yantra will fall back to the base language.

For example, there are three locales, en_US, en_GB, and fr_FR. Currently en_US is the base language. fr_FR has full translations, but en_GB only has a few translations, such as *color* to *colour*. Using the en_GB locale, the en_GB translation for *apple* does not exist, therefore Yantra falls back to the en_US translation of *apple*.

When the base language is switched to French, the fall back language becomes French. Now when the en_GB translation of *apple* does not exist, Yantra falls back to the fr_FR translation, *pomme*.

It is recommended that *all* translations be provided for *all* locales.

Switching the Base Language

1. To validate that there are no issues in making a base language switch, run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in SWITCHTEST mode as follows:

```
ant -f localizedstringreconciler.xml switchtest -Dlanguage=xx
-Dcountry=YY
```

where xx and YY are the desired base language and country. For example, fr and FR for French and France, respectively. This will simulate the switch and report any errors it finds, without making any changes to the database.

2. To perform the base language switch, run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in SWITCH mode as follows:

```
ant -f localizedstringreconciler.xml switch -Dlanguage=xx
-Dcountry=YY
```

where xx and YY are the desired base language and country. For example, fr and FR for French and France, respectively.

2.4.1.4 Full Export to Back-up Existing Localization Literals

To create a full export to back-up existing localization literals:

1. Run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in `EXTRACT` mode as follows:

```
ant -f localizedstringreconciler.xml extract
-Ddestdir=<YFS_HOME>/database/FactorySetup/XMLS
```

This exports the literals that are currently defined in the `YFS_LOCALIZED_STRINGS` table.

To extract existing localized strings for a specific locale, use the optional parameters `locale`, and `country`. The usage is as follows:

```
ant -f localizedstringreconciler.xml extract
-Ddestdir=<YFS_HOME>/database/FactorySetup/XMLS
-Dlanguage=fr -Dcountry=FR
```

For example, if there are three locales that have been localized for factory default data in the database, one file for each locale is created in the following format:

```
<baselanguage>_<basecountry>_ycpdblocalizedstrings_
<language>_<country>_db.properties
```

Each file is saved within the `extn` directory within the respective locale directory. For instance, for the `fr_FR` locale, the file `en_US_ycpdblocalizedstrings_fr_FR_db.properties` is created and saved to the `<YFS_HOME>/database/FactorySetup/XMLS/fr_FR` directory.

2.4.1.5 Deleting Unused Localization Literals

To delete unused localization literals, run the `LocalizedStringReconciler` tool from `<YFS_HOME>/bin` in `EXPORT` mode as follows:

```
ant -f localizedstringreconciler.xml export -Ddestdir=<YFS_
HOME>/database/FactorySetup/XMLS -Ddelete=true
```

This deletes the unused literals, while exporting the literals that are currently defined in the `YFS_LOCALIZED_STRINGS` table.

2.5 Directory Structure and Filenames

The Yantra 7x directory structure must be maintained based on the language and country localization. File names are limited to one locale and can only contain characters a through z, 0 (zero) through 9, and the underscore (_).

Likewise, since many files are created using a transaction ID, this rule also applies to transaction names and transaction IDs. Similarly, you should adhere to these limitations when localizing the template filename for transaction events (`TransactionID.On_SUCCESS.xml`).

Yantra 7x follows the hierarchy of lookups given below:

1. When a certain locale is selected in the console it searches for some files with that particular suffix specified in the locale. For example, if the locale `ja_JP` is chosen in the console, the system searches for `ycpapibundle_ja_JP.properties`, `validation_ja_JP.properties` and `sapphire_ja_JP.css` files.
2. If these files are not found, the systems looks at the default locale of the operating system that the application server is running. If it is `en_US`, the system tries to look for the above files with `en_US` suffix. For example, `ycpapibundle_en_US.properties`.
3. Only if the above two steps fails, the default files (for example, `ycpapibundle.properties`) are used.

The method of defaulting to the standard files occurs for the files that can have the locale suffix such as `alertmessages.js` and `yantraiconsbe.jar` and so on.

2.6 User Interface Themes

The user interface theme files specify screen colors and display fonts to use. Display fonts are dependent on what languages need to be supported. Some fonts may not support all languages. For example, Tahoma (the Yantra 7x default font) does not support Japanese, for the best results when localizing Yantra 7x to Japanese use the font MS Gothic. When setting up a theme, choose a font that displays the specific language you need. For example, when setting up a Japanese locale, customize the theme to use a font that displays Japanese characters such as Hiragana.

Note: If you use a font that is bigger than the Yantra 7x default font (Tahoma), it may be necessary to customize the `YFSDataTypes.xml` file to increase the user interface size of data types that are used for input fields in the Yantra 7x Application Consoles. In particular, the user interface size of the "Date" data type should be increased.

2.6.1 Themes

When localizing your Yantra 7x Configurator user interface themes, you modify the theme-specific XML file, when localizing your Yantra 7x Application Consoles user interface themes, you modify the theme-specific CSS file. For example, the following files must be localized for themes:

```
<YFS_HOME>/template/api/<theme>.xml and <YFS_
HOME>/webpages/css/<theme>.css
```

These files are localized by appending the language and country codes in the file name. For example, if you are using the sapphire theme in a French locale, you must localize the following files:

```
<YFS_HOME>/template/api/sapphire_fr_FR.xml
<YFS_HOME>/webpages/css/sapphire_fr_FR.css
```

The following themes are distributed with the Yantra 7x application:

- Earth (`<YFS_HOME>/template/api/earth.xml` and `<YFS_HOME>/webpages/css/earth.css`)
- Jade (`<YFS_HOME>/template/api/jade.xml` and `<YFS_HOME>/webpages/css/jade.css`)
- Sapphire (`<YFS_HOME>/template/api/sapphire.xml` and `<YFS_HOME>/webpages/css/sapphire.css`)

To localize a theme:

1. Copy the `<YFS_HOME>/template/api/<theme>.xml` file and save it as `<YFS_HOME>/template/api/<theme>_<language>_<country>.xml`.
2. Copy the `<YFS_HOME>/webpages/css/<theme>.css` file and save it as `<YFS_HOME>/webpages/css/<theme>_<language>_<country>.css`.

You must edit the `<YFS_HOME>/webpages/css/<theme>_<language>_<country>.css` file to change the display font for the Yantra 7x Application Consoles. In addition, the font name and size for the graph displayed in Inventory Summary screen in the Inventory Console is configured in the `<YFS_HOME>/template/api/<theme>_<language>_<country>.xml` file.

For example, in the default `sapphire.xml` file, the graph font is configured as:

```
<!-- Font for Inventory Graphs(Axis Titles & Lables) -->
<Font Name="InvGraphFont" FontName="Tahoma" FontSize="12" />
<!-- Font for Inventory Graphs -->
```

To localize for double-byte languages such as Japanese, Yantra 7x recommends that you edit the `<YFS_HOME>/template/api/<theme>_<language>_<country>.xml` file to use either the serif or sansserif font as below.

```
<!-- Font for Inventory Graphs(Axis Titles & Lables) -->
<Font Name="InvGraphFont" FontName="serif" FontSize="12" />
<!-- Font for Inventory Graphs -->
```

Note: If your application server is running on Unix, the valid fonts you can use are stored in the `<JAVA_HOME>/jre/lib/font_properties.<file.encoding>` file.

2.7 Literals

All Yantra 7x components use a common resource bundle that contains literals displayed on the screens. Yantra 7x enables you to customize and localize resource bundles as needed.

In addition, literals used in customized screens have their own resource bundle and should also be considered during the localization process. For information about localizing your customizations to Yantra 7x, see the *Yantra 7x Customization Guide*.

Note: Literals cannot be localized in the following places:

- Condition Builder
 - Order/Shipment Monitor
 - Hard-coded literals in APIs
-
-

For a complete list of resource bundle literals, along with the screens on which those literals appear, see the `<YFS_HOME>/documentation/resource_mapping.html` file.

2.7.1 Resource Bundles

Yantra always releases complete resource bundles in the `ycpapibundle.properties` file with the localized versions of the Yantra 7x application. Incremental updates are not provided. If you localize the Yantra 7x application, it is your responsibility (or that of your 3rd-party localization company) to compare and validate the differences between the resource bundles shipped with the product to those you have localized.

The resource bundles of the Yantra 7x application are located in the `<YFS_HOME>/resources/ycpapibundle.properties` file.

To localize the resource bundles:

1. Copy the `<YFS_HOME>/resources/ycpapibundle.properties` file and save it as `<YFS_HOME>/resources/ycpapibundle_<language>_<country>.properties`.
2. Each resource bundle contains a `<key>=<value>` pair where key is the resource key and value is the literal displayed for the corresponding locale. Replace `<value>` with the translated value.
 - When localizing menus in the Configurator, by default the accelerator key is the first character in a menu item. To specify any other character to be the accelerator key, insert an ampersand (&) just before that character.
 - When localizing the console UI, be aware that changing the height or width of the text in the application may affect the layout of the screens. It may be necessary to customize certain screens to achieve optimal layout after the other localization steps are

complete. For example, if the resource bundle contains translated literals that are lengthy, you may need to increase the width of the screen in order to accommodate the larger size of the translated literal.

- Some of the literals that need to be translated in the resource bundles contain data place holders. These data place holders indicate that the literal is displayed with one or more data values within the literal. For example, the application displays the error message "Priority should be greater than X" where X could be any number. Since the location of the "X" within the literal can be different for different languages, the resource bundle uses a place holder that can be placed anywhere in the literal during translation. The resource bundle entry looks like this:

```
PRIORITY_ERROR_MESSAGE=Priority should be greater than {0}
```

Notice how the "{0}" place holder indicates where the dynamic data value appears in the literal. This "{0}" can be placed anywhere in the literal. For example,

```
PRIORITY_ERROR_MESSAGE=A number greater than {0} should be entered
```

or

```
PRIORITY_ERROR_MESSAGE={0}: Priority entered should be greater than this
```

are valid possibilities. This gives you the flexibility to translate the literal in any way that the language dictates. Note that multiple place holders may appear in the literal as well. For example, {0}, {1}, {2}, and so forth. Each place holder must exist somewhere in the corresponding translated literal.

- When using literals that contain data place holders, you cannot use single quotation marks. If a single quotation mark is used in conjunction with a place holder, the single quotation mark is not displayed and the place holder is not replaced with its replacement value. In order to avoid this situation, enter two single quotes wherever a single quote is required. For example, the following is invalid:

```
PRIORITY_ERROR_MESSAGE=The primary organization's name is {0}
```

but the following is valid:

```
PRIORITY_ERROR_MESSAGE=The primary organization''s name is {0}
```

- Files should be returned in native format with UTF-8 encoding.
 - Properties should be returned in escaped Unicode format with UTF-8 encoding.
3. The default font used is Tahoma. Therefore if you want to display or type Unicode characters, you should localize the theme. This is done by changing the font to Unicode in the theme-specific XML files.
 4. Save the modified file. If the file is in UTF-8 format, convert it to ASCII by running the native2ascii command as follows:

```
native2ascii -encoding UTF-8 <source file> <target file>
```

Note that files localized in Latin1 languages do not require this conversion.

The file should be returned in the following format:

```
<filename>_<2 letter code for language as given by ISO 639>_<2 letter code for territory as given by ISO 3166>.<file extension>
```

For example, `ycpapibundle.properties` should be returned as `ycpapibundle_fr_FR.properties`.

5. If you are customizing Yantra 7x, save the extended resource bundles as `<YFS_HOME>/resources/extn/extnbundle_<language>_<country>.properties`.

For example, `ycpapibundle.properties` should be saved as `ycpapibundle_fr_FR.properties`.

6. For extended tag attributes add the following bundle entry in `extnbundle.properties` for each extended tag attribute:

```
Item_Tag_<TagName>=<Tag Name>
```

7. Copy the `ycpapibundle_<language>_<country>.properties` file to the `<YFS_HOME>/webpages/yfscommon` directory.
8. Copy the `extnbundle_<language>_<country>.properties` file to the `<YFS_HOME>/webpages/extn` directory.

Note: The `Custom_Code_Prefix` and `Custom_Code_Postfix` properties in the `ycpapibundles.properties` file are used to prefix or append identifying literals or extensions to your newly created custom transaction IDs, supply types, demand types, or document types. When you create a new transaction ID, supply type, demand type, or document type, the value specified for these properties is prefixed or appended to each of these types of literals when displayed in the user interface. The default value for the `Custom_Code_Prefix` is "" (blank) and the `Custom_Code_Postfix` is ".ex". You may change this value if it does not suit your needs.

2.8 Date and Number Validation

Date and number validations are performed using JavaScript. By default, Yantra 7x provides validation for the `en_US` locale (English for the United States).

2.8.1 Date and Time Validation

Dates can be stored in a standard format but displayed according to the required format. If the date is entered on one of the screens it has to be verified by some means. Date format is specified in the Yantra 7x Configurator Locale Details screen. For more information, see the *Yantra 7x Platform Configuration Guide*.

To localize date and time validation:

1. Save the `<YFS_HOME>/webpages/yfcscripts/Validation.js` file as `Validation_<language>_<country>.js` (in UTF-8 encoding format) and make modifications to the new file as indicated in the following steps.
2. Ensure that date and time values match the entries specified in the Locale fields in the Yantra 7x Configurator. For detailed information about the Locale fields and their suggested syntax, see the *Yantra 7x Platform Configuration Guide*.

3. Change the `[yfcDateFormat]` (MM/dd/yyyy), `[yfcTimeFormat]` (hh:mm:ss) and `[yfcDateTimeFormat]` (MM/dd/yyyy hh:mm:ss) variables to contain the correct date and time format.

These date and time formats are according to the United States English version. When you create a `Validation.js` file for another language, these formats change accordingly.

Table 2–1 Date and Time Variable Formats

Date Element	Description
yyyy	Four-digit year - for example, 2002
MM	Two-digit month - for example, 05 for May
dd	Two-digit day - for example 01 for the first day of the month
mm	Minutes
hh	Hours
ss	Seconds

4. Localize the popup messages and calendar formats as described in [Section 2.10, "Calendar and Message Popups in the Console"](#) on page 34.

2.8.2 Number Validation

By default, Java displays the localized number format based on what is specified for `<language>_<country>` in the Configurator. In order for the Yantra 7x UI to validate numeral values, it reads in values as specified in the `Validation_<language>_<country>.js` file. This means that the validation JavaScript file must contain validation logic that is specific to the number format to be used for validation for each locale.

To localize number validation:

1. Save the `<YFS_HOME>/webpages/yfcscripts/Validation.js` file as `Validation_<language>_<country>.js` (in UTF-8 encoding format).
2. Change the decimal separator and grouping separator as needed. The grouping separator indicates how to format numbers visually; it has no impact on the actual value of the number.

For example, en_US uses a comma (",") and fr_FR uses a non-breaking space (" "). When specifying the separator, use the Unicode literal.

For example, the non-breaking character for fr_FR would be specified as shown below:

```
yfcGroupingSeparator = "\u00a0";
```

3. Localize the exception messages for invalid number format as described in [Section 2.7, "Literals"](#) on page 27.

2.9 Templates

Among the templates used by Yantra 7x, you can localize e-mail and exception templates.

2.9.1 E-mail Templates

You can store e-mail templates using any character encoding format, but the encoding format must be set with the

`yfs.email.template.encoding` property in the `yfs.properties` file.

2.9.2 Exception Alert Templates

Exception alert templates enable you to supply additional text to alerts raised. This enables you to make error message more descriptive and easy to understand. They also provide a means of supplying a hyperlink to the resolution screens from the Alert Console.

For example: for any alert created for an order, shipment, or load document type, a hyperlink is created and displays in the "Created For" column on the Alert List screens. You localize the literals displayed in this column by translating them in the `DefaultListTemplate.xsl` file located in your `<YFS_HOME>\ProductFiles\template\exception_console` directory.

You can store exception alert templates using any character encoding format, but the encoding format must be set with the `yfs.file.encoding` property in the `yfs.properties` file. If these properties are not explicitly set in the `yfs.properties` file, Yantra 7x uses UTF-8 character encoding.

2.10 Calendar and Message Popups in the Console

Calendar and message popup windows contain literals that must be localized.

To localize date formats for calendar popup windows in the Console:

Note: Only the literals in calendar popup windows can be localized. The order of the days cannot be localized. Therefore, the week always starts with Sunday.

1. Copy the `<YFS_HOME>/webpages/common/alertmessages.js` file and save it as `<YFS_HOME>/webpages/common/alertmessages_<language>_<country>.js` (in UTF-8 encoding format).
2. Translate the date-related literals using this following as described below:
 - `monthArray` - contains the series of literals to use for months of the year
 - `weekdayList` - contains the series of literals to use when displaying the literals for days of the week
 - `weekdayArray` - contains the series of literals to use to when displaying the shortened literals for days of the week
 - `todayString` - displays the word "Today"

To localize exception messages:

1. Copy the `<YFS_HOME>/webpages/common/alertmessages.js` file and save it as `<YFS_HOME>/webpages/common/alertmessages_<language>_<country>.js`, (in UTF-8 encoding format).
2. Edit your exception messages file as needed.

2.11 Images

Images and icons are stored in JAR files. A separate JAR file can be used for each locale. The Configurator and Application Consoles both use their own mechanisms for reading in images.

2.11.1 Application Consoles Images

To localize images used in the Application Consoles:

1. Copy the `<YFS_HOME>/webpages/yfscommon/yantraiconsbe.jar` file to a new `<YFS_HOME>/webpages/yfscommon/yantraiconsbe_<language>_<country>.jar` file.
2. Within your custom JAR file, add or remove images as needed.
3. If you are customizing and localizing Yantra 7x, your custom images are read in from the `<YFS_HOME>/webpages/extn/icons/yantraiconsbe.jar` file. Therefore, you must copy the images from the `<YFS_HOME>/webpages/extn/icons/yantraiconsbe.jar` file and make the necessary locale-specific changes in the images.
4. Save the changed images in `<YFS_HOME>/webpages/extn/icons/yantraiconsbe_<language>_<country>.jar` file.

Note: The warning message that displays when you click on the Yantra icon in the top right-hand corner of the Yantra 7x Application Consoles user interface is text included in an image file. You can localize this message by following the instructions in this section to localize the `/console/icons/about_box_back.gif` file.

2.11.2 Configurator Images

To localize images used in the Yantra 7x Configurator:

1. Copy the images from the `<YFS_HOME>/webpages/yfscommon/yantraicons.jar` file and make the necessary locale-specific changes to the images.

2. Save the changed images in `<YFS_HOME>/webpages/yfscommon/yantraicons_<language>_<country>.jar` file.

Note: The warning message that displays on the bottom of the Yantra 7x Configurator About box when you select Help > About from the Yantra 7x Configurator menu is text included in an image file. You can localize this message by following the instructions in this section to localize the `/com/yantra/ycp/ui/icons/about.gif` file.

A

Localizable XML Attributes

The [Table A–1, "Localizable Factory Setup XML Attributes"](#) table below enables you to determine the XML file name associated with the specific XML attributes. This table lists the factory default XML attributes that can be localized along with the associated database table name. The full name of the XML file can be derived from the database table name using the following procedure.

To derive a list of XML files:

1. Find the database table that corresponds with the XML attribute you want to translate.
2. Using the database table name, append .xml to it and prepend the appropriate prefix for the module to which it belongs. Module prefixes are as follows:
 - INV - Inventory Synchronization
 - OMD - Order Management, Supply Collaboration, and Reverse Logistics
 - OMP - Order Management Platform
 - REF - Reference implementation
 - VAS - Value Added Services
 - WMS - Warehouse Management Services
 - YCM - Product Management
 - YCP - Platform
 - YDM - Delivery Management

For example, the Actionname attribute corresponds with the YFS_ACTION database table, which uses the YCP_YFS_ACTION.xml file.

Table A-1 Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_ACTION	Actionname
YFS_ACTIVITY	Description
YFS_ACTIVITY_CONSTRAINT	ConstraintDescription
YFS_ADAPTER	AdapterDescription
YFS_ADJUSTMENT_REASON	Description
YFS_ALLOCATION_RULE	Description
YFS_ANSWER_OPTION	Description
YFS_ATP_RULES	AtpRuleName
YFS_BARCODE_TRANSLATION	Description
YFS_BASE_ACTIVITY_GROUP	ActivityGroupName
YFS_BASE_DOCUMENT_PARAMS	Description
YFS_BASE_EVENT	EventName
YFS_BASE_PACK_DO_NOT_MIX	FieldNameDesc
YFS_BASE_PROCESS_TYPE	ProcessTypeName
YFS_BASE_PURGE_CRITERIA	PurgeCodeDescription
YFS_BASE_SHIP_CONSTRAINTS	FieldNameDesc
YFS_BASE_TRANSACTION	BaseTranname
YFS_BUSINESS_DOCUMENT	DocumentDescription DocumentName
YFS_CALENDAR	CalendarDescription
YFS_CARRIER_SERVICE	CarrierServiceDesc
YFS_CATEGORY	Description
YFS_CATEGORY_DOMAIN	ShortDescription Description
YFS_CHARGE_CATEGORY	Description

Table A–1 Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_CHARGE_NAME	Description
YFS_CLASSIFICATION_PURPOSE	ClassificationPurposeDesc
YFS_COMMON_CODE	CodeLongDescription CodeShortDescription Note: For entries where CODE_TYPE starts with CODE_TYPE_LIST, you must change the CODE_DESCRIPTION property in the ycpapibundle.properties file.
YFS_CONDITION	ConditionID ConditionName
YFS_COUNT_STRATEGY	Description
YFS_CURRENCY	CurrencyDescription
YFS_DATE_TYPE	Description
YFS_DEVICE_SUB_TYPE	Description
YFS_DEVICE_TYPE	Description
YFS_DOCUMENT_FORMAT	FormatDescription
YFS_DOCUMENT_PARAMS	Description
YFS_ENTERPRISE	Enterprisename
YFS_EQUIPMENT_TYPE	Description
YFS_ERROR_CAUSE_ACTION	Cause Action
YFS_ERROR_CODE	ErrorMessage
YFS_EVENT	EventName
YFS_EXECUTION_EXCEPTION	ExceptionShortDescription ExceptionLongDescription
YFS_FREIGHT_TERMS	Description ShortDescription
YFS_HOLD_TYPE	HoldTypeDescription
YFS_INVENTORY_DEMAND_TYPE	Description

Table A–1 Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_INVENTORY_STATUS	Description
YFS_INVENTORY_SUPPLY_TYPE	Description
YFS_ITEM_UOM_MASTER	Description
YFS_LOCALE	LocaleDescription
YFS_LOCATION_SIZE	Description
YFS_MONITOR_TYPE	Field1Name, Field2Name, Field3Name, Field4Name, Field5Name, Field6Name, Field7Name, Field8Name, Field9Name, Field10Name
YFS_NEGOTIATION_RULE	NegotiationRuleId
YFS_ORDER_LINE_TYPE	LineTypeDesc
YFS_ORGANIZATION	OrganizationName Note: OrganizationCode and OrganizationKey cannot be localized.
YFS_PAYMENT_TYPE	PaymentTypeDescription
YFS_PICK_STRATEGY	Description
YFS_PIPELINE	PipelineDescription PipelineId
YFS_PLA_ACTIVITY_DETER	Description
YFS_PLA_ZONE_SET	Description
YFS_PRINT_DOCUMENT	PrintDocumentDescription
YFS_PROCESS_TASK_TYPE	ProcessTaskTypeDesc
YFS_PROCESS_TYPE	ProcessTypeName Description
YFS_PRODUCTIVITY_TYPE	Description ProductivityType
YFS_PURGE_CRITERIA	PurgeCodeDescription ErrFileName LogFileName

Table A–1 Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_QUESTION	QuestionText
YFS_QUEUE	QueueDescription Note: QueueKey and QueueId cannot be localized.
YFS_REGION_LEVEL	AddressFieldAlias
YFS_REGION_MATCH_PREF	AddressFieldAlias
YFS_RETRIEVAL_STRATEGY	Description
YFS_RETURN_DISPOSITION	Description
YFS_ROLE	RoleDescription Note: RoleID and RoleKey cannot be localized.
YFS_ROLE_DOCUMENT	DocumentName DocumentDescription
YFS_RULES	RuleSetFieldDescription
YFS_SCAC	ScacDesc Note: Scac and ScacKey cannot be localized.
YFS_SCAC_AND_SERVICE	ScacAndService ScacAndServiceDesc
YFS_SERVICE_TYPE	Description
YFS_SHIPMENT_GROUP	Description
YFS_SPECIAL_SERVICES	SpecialServicesDescription
YFS_STATUS	Description StatusName
YFS_STATUS_MODIFICATION_TYPE	ModificationLevelScreenName ModificationTypeScreenName
YFS_TASK_TYPE	TaskTypeName
YFS_TRANSACTION	Tranname

Table A-1 Localizable Factory Setup XML Attributes

Database Table Name	XML Attribute Name
YFS_UOM	UomDescription Note: If the YFS_UOM.xml is changed, the YFS_UOM_CONVERSION.xml should also have corresponding changes.
YFS_USER	Username
YFS_USER_EXIT_IMPL	ImplementationNotes
YFS_USER_GROUP	UsergroupName
YFS_WAVE_SIZE_CONSTRAINT	Description
YFS_ZONE	Description
YFS_COUNT_PROGRAM	CountProgramName
YFS_COUNT_PROGRAM_COND	Description

B

base language
switching, 23

C

calendars, 34
character encoding, 12
Latin-1, 13
setting up, 13
specifying JRE settings, 13
UTF-8, 14
setting up environment, 14
character sets, 2, 15
charges and taxes, 5
components and data
localizable, 1
conversion rates
definitions, 6
currency, 4
conversion, 5
conversion rates, 6
conversion scenario, 6
precision, 5
storing exchange rates, 7

D

data
overview, 17
data encoding format, 15
databases, 15, 17

oracle database setup, 16
SQL Server database setup, 16
date and number validations, 31
date and time validations, 31
date formats, 3
dates with no time component, 4
date layout, 1
DBCS (Double-byte Character Set), 2
DB2 database setup, 17
double-byte character set. See DBCS

E

e-mail templates, 33
exception alert templates, 33
exception messages, 34
exchange rates
storing, 7
external literals, 12

F

factory defaults, 15
setup attributes, 37
formats
date, 3
time, 3

I

images, 35
Application Consoles, 35
Yantra 7x Configurator, 35

internationalization, 1
ISO (International Standards Organization)
codes, 11

J

java database connectivity. See JDBC, 3
JDBC (Java Database Connectivity), 3

L

literals
overview, 27
locale
country code, 1
definition, 1
language code, 1
locale definition, 1
localizable
data, 2
literals, 2
localizable components, 1
localizable data, 1
localization
multi-language installation, 17
back-up existing localization literals, 24
delete, 24
deleting unused localization literals, 24
export mode, 18
extending factory-shipped translations, 20
full export, 24
import mode, 19, 20
localization rules, 11
lookup hierarchy, 25

M

MBCS (Multi-byte Character Set), 2
message popups, 34
mobile device clients, 12
multi-byte character set. See MBCS
multi-language installation, 17

N

non-localizable components, 7
non-localizable data, 7
number validations, 32
numeric formats, 4
numeric formatting, 1

R

resource bundles, 28
localizing, 28

T

templates, 33
themes, 26
time formats, 3
time zones, 3
sensitivity, 3
sensitivity characteristics, 4
transitional currency, 4

U

units of measure. See UOMs
universal time coordinate. See UTC
UOMs (Units of Measure), 7
user interface theme files, 25
UTC (Universal Time Coordinate), 4
UTF-8, 3

W

Windows clients, 12