



Globalization in WebSphere Commerce Suite 5.1



What We are Talking About

- **Multilingual Support**
 - ▶ program that supports the use of multiple languages and strives to handle data in a way that is not dependent on a particular language or writing system
- **Multicultural Enablement**
 - ▶ the capability to provide a Web Site shopping experience whereby catalogs and other information are presented in a language of customer choice
- **National Language Support**
 - ▶ The features of a product that accommodate a specific region, its language, script, local conventions and culture. Your product can achieve NLS internationalization and localization

What We are Talking About cont'd.

- Internationalization
 - ▶ Designing and developing a software product to function in multiple locales. This process involves identifying the locales that must be supported, designing features which support those locales, and writing code that functions equally well in any of the supported locales
- Localization
 - ▶ Modifying or adapting a software product to fit the requirements of a particular locale. This process includes: translating the user interface, documentation and packaging, changing dialog box geometries, customizing features, and testing the translated product to ensure that it works

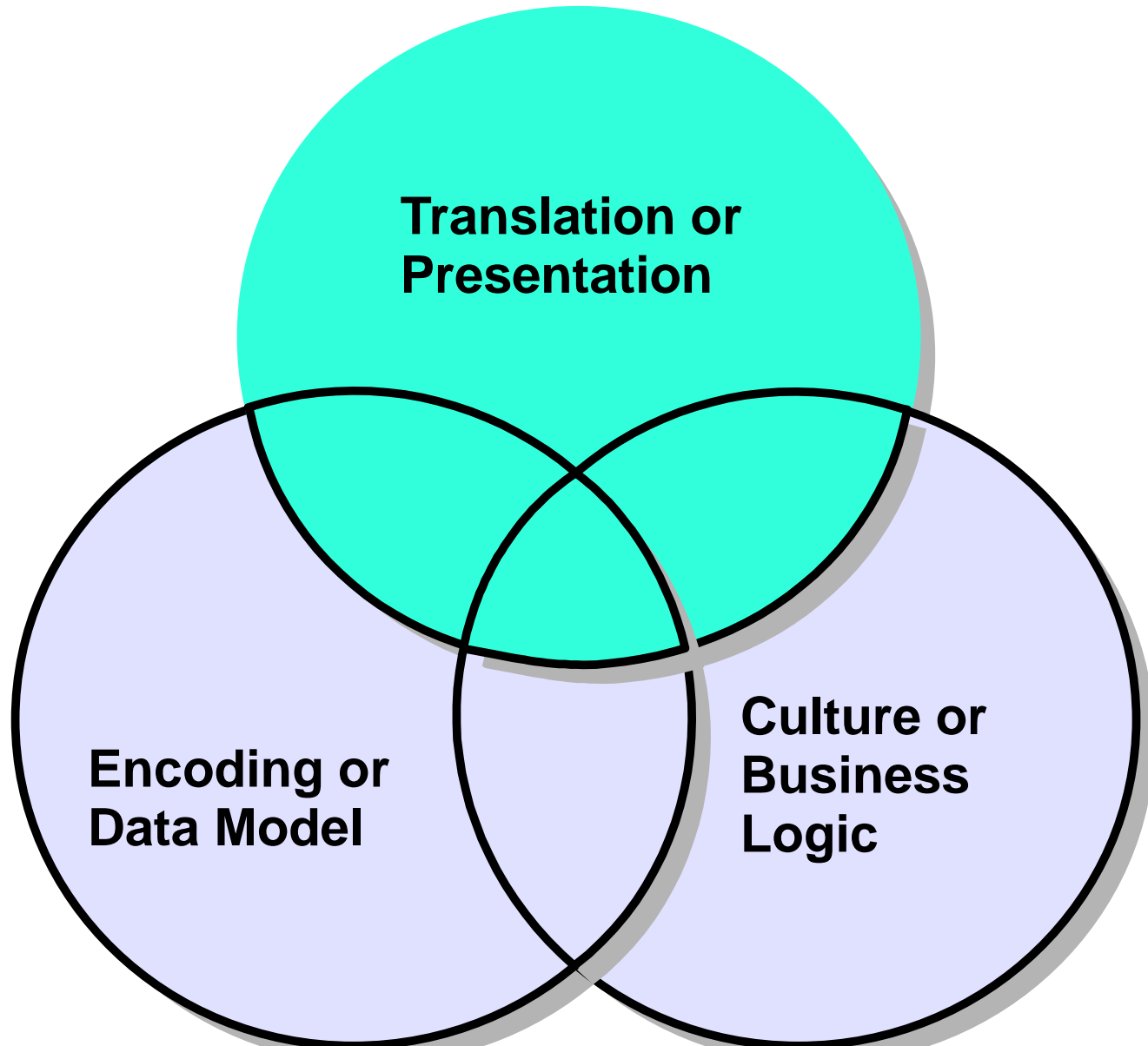
What We are Talking About cont'd.

- Globalization
 - ▶ is the proper design and execution of systems, software, services, and procedures to allow users to work throughout the world, and among people with different languages, business requirements, and cultural requirements.

Why These Issues Are Important

- To establish rapport with a wide customer base
- To build customer loyalty
- To increase the opportunities for sales
- A single server supports applications in different languages, and applications that support multiple languages simultaneously
- Design networks and deploy servers based on load and resources rather than on language support requirements
- The cost of support, maintenance and training is reduced

Globalization Support



Translation

- Considerations
 - ▶ translation is not the automatic replacement of one word of the source language with its equivalent in the target language; it is the re-expression of the same concept in the target language
 - ▶ terminology/technical terms need to be defined and acceptable translations made without inconsistencies
 - ▶ all machine readable information (MRI) must be isolated from code so it and only it gets translated
 - ▶ messages and other MRIs must be complete sentences; they cannot be constructed from parts of sentences only

Translation cont'd.

- Activities
 - ▶ provide for effective presentation of MRI after the expansion that results from translation
 - ▶ ensure that functions that depend on the location of user interface elements are not inhibited by display position changes, caused by MRI expansion
 - ▶ in addition to messages, dialog boxes, online manuals, animations and help text, get audio output, tutorials, diagnostics, clip art and icons translated

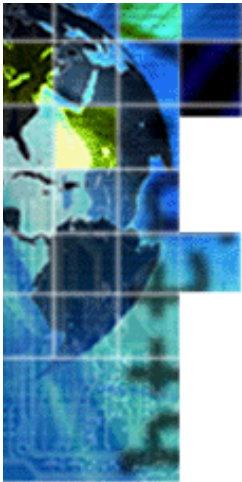
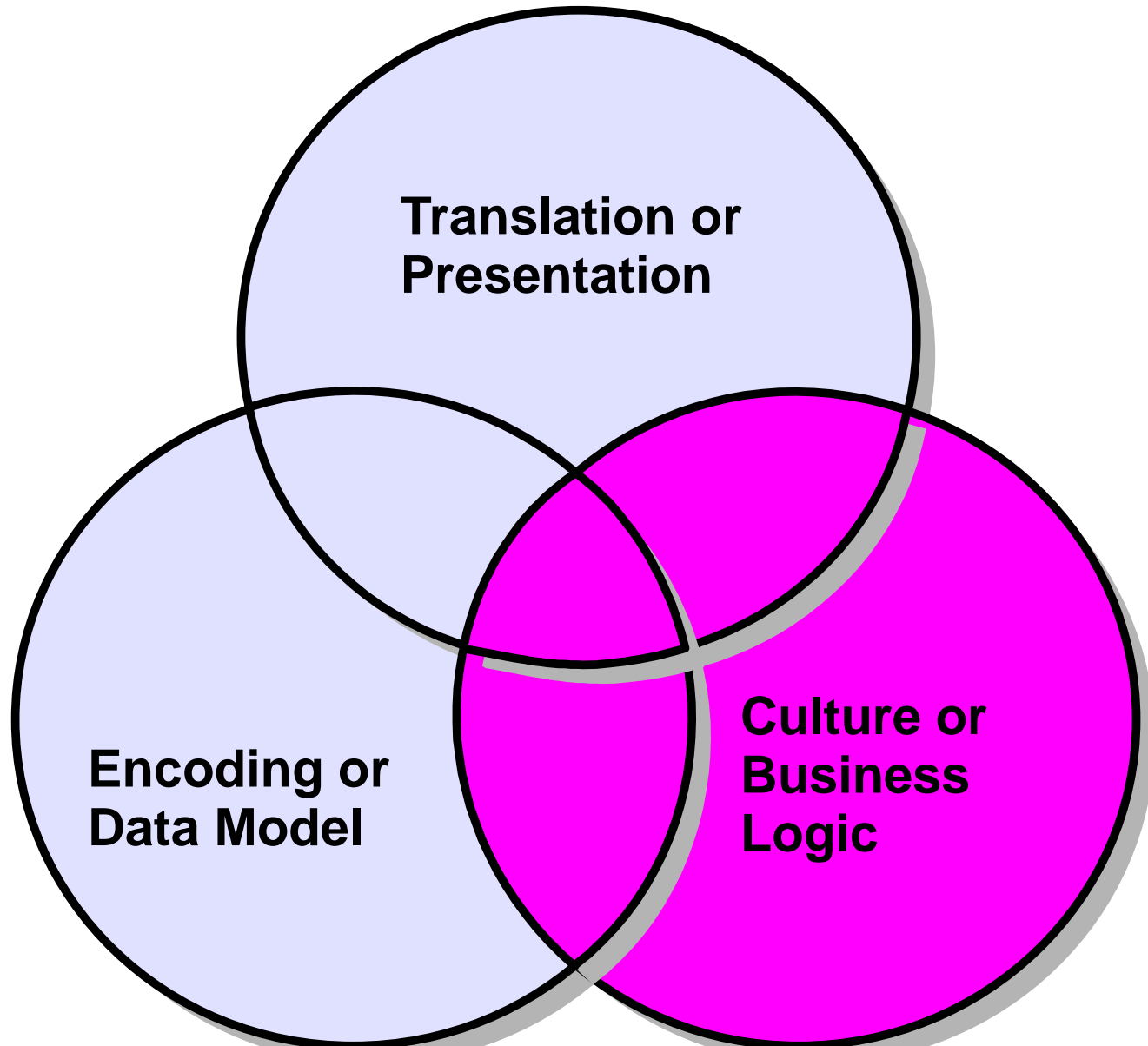
Example of a Translation Problem

An icon that shows a horizontal hand (with pointing index finger) is used to indicate an indexing facility on a dialog box.



When the dialog box is translated into an European language, the translator warns that users may interpret that icon as an obscene gesture. The dialog box must be redesigned entirely because the icon is hard coded into the dialog box.

Globalization Support



Cultural Considerations

- Name and address formatting
- Currency units and formatting
- Measurement systems
- Date/time formatting
- Decimal and thousands separators
- Use of colour
- Taxation and shipping calculation methods
- Various payment methods

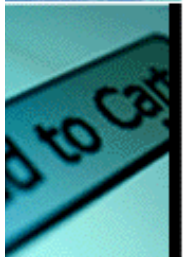
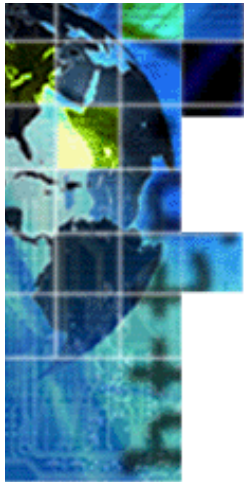
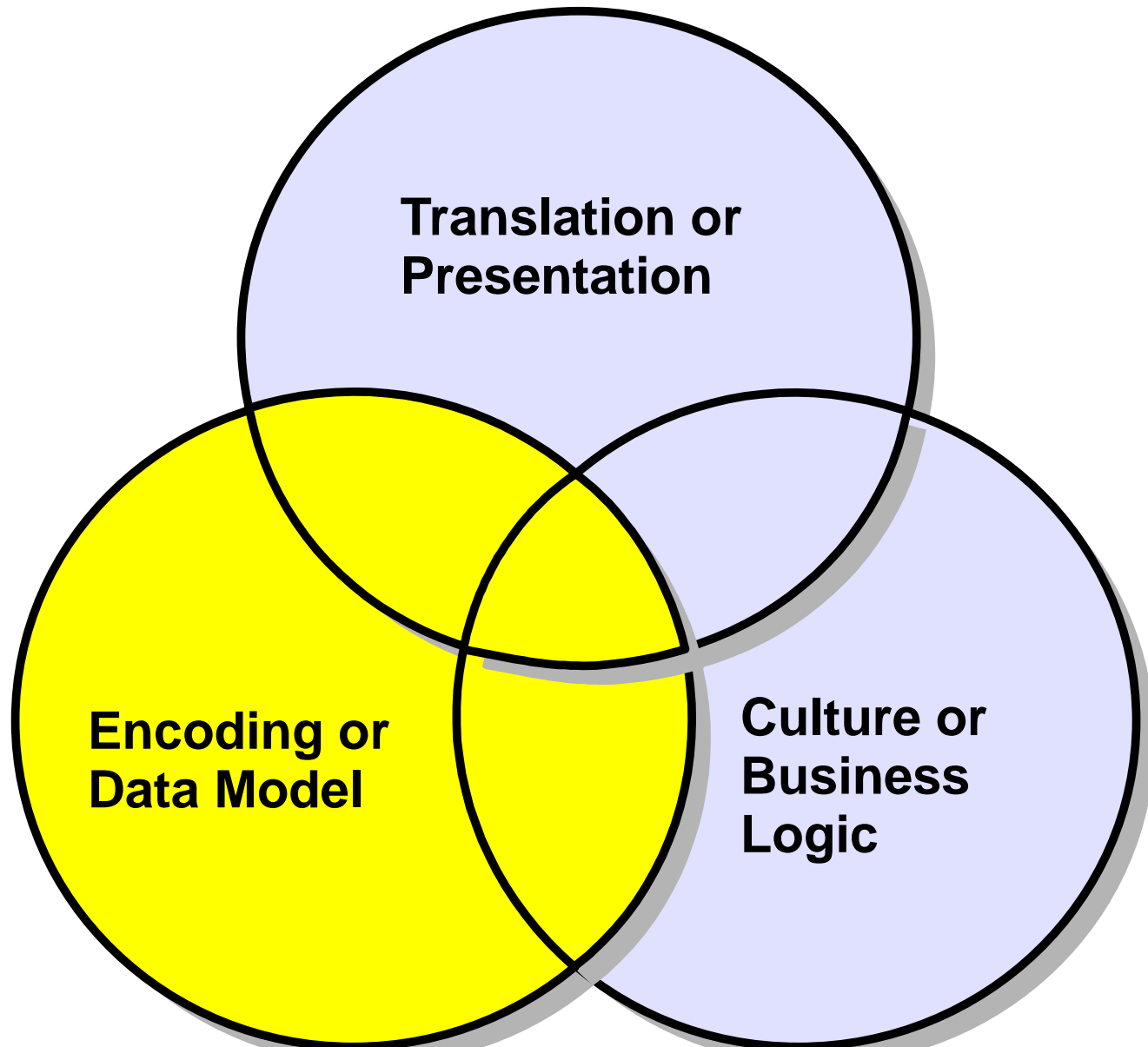
Cultural -Date Formatting

Country	Preferred	Alternate	Month & Date Leading
Brazil	dd/mm/yyyy	yyyy.mm.dd	Y
Italy	dd/mm/yy	dd/mm/yyyy	Y
France	dd/mm/yyyy	dd.mm.yy	Y
Germany	dd.mm.yyyy	dd.mm.yy	N
Japan	yyyy-mm-dd	mm/dd/yy	Y
Korea	yyyy.mm.dd	yy/mm/dd	Y
China	yyyy.mm.dd	none	Y
Taiwan	yy/mm/dd	none	Y

Cultural - Currency Formatting

Country	Currency Name	Positive Formats	ISO 4217 Code
Brazil	real	R\$ 123,45	BRC
China	yuan renminbi	Y 123	CNY
France	franc	123,45 F	FRF
Germany	Mark	123,45 DM	DEM
Italy	lira	L. 123	ITL
Japan	yen	Y 123	JPY
Korea	won	W 123	KRW
Spain	peseta	123 Pts	ESP
Taiwan	new Taiwan dollar	NT\$ 123.4	TWD

Globalization Support



Encoding/Data Model

- A new universal encoding system has been devised called Unicode
- Unicode recognizes and provides for the conventions of different languages
- A standard for the storage and interchange of textual data
- A Unicode font is usually a subset of this larger standard and contains "glyphs"

Unicode

- Enables a system to input, output, store, display, print, and send data in all the world's major languages
- Allows storage and access to all Unicode characters without corruption
- Has the ability to present sorted results based on locale-specific multilevel collation rules

Database Preparation

- Create the database from a UTF-8 client
- Set DB2CODEPAGE environment variable on the client to 1208 to create a UCS-2 database
- Explicitly specify "UTF-8" as the CODESET name and use any valid two letter TERRITORY code supported by DB2 UDB
- Consider byte size of ASCII and non-ASCII characters
- Consider different storage requirements of character string UTF-8 and graphic string UCS-2
- Specify indexes carefully
- Remember that pattern matching is slightly different in a UCS-2 database from existing MBCS databases

Globalization Challenges

- Integration with operating systems, other applications
- Input/Output
- Text Layout
- Fonts
- Legacy data and applications

Integration with Operating Systems and Other Applications

- To achieve globalization with operating systems and other applications:
 - ▶ single body of globally executable code per platform should be installable and work for and in any supported locale
 - ▶ operating systems must supply common APIs
 - ▶ applications, middleware and other parts of the operating system must use common APIs for locale sensitive operations
 - ▶ programs that are part of an e-business system and that process textual user data should use Unicode internally to avoid corruption problems
 - ▶ use application messaging services
 - ▶ use business process integration and workflow services
 - ▶ use component integration services

Input/Output

- Full globalization of input and output includes:
 - ▶ use of input method editors (IMEs)
 - ▶ use of Unicode in IMEs for communicating text
 - ▶ IMEs that provide for both Unicode and legacy character set APIs
 - ▶ provide enough glyphs to represent all the required composite characters for the supported languages
 - ▶ provide for scripts that do not have a 1-1 mapping between characters in memory and visible shapes on the screen or printer

Text Layout

- The process for determining which sequence of glyphs corresponds to a given sequence of characters
- Includes mapping back and forth between ranges in character space and metrics in glyph space
- Can be handled by OpenType

Fonts

- To support complex output in a globalized site, the system needs:
 - ▶ a range of separate fonts that covers the supported languages
 - ▶ a single unified font such as WorldType Fonts

Legacy Data and Applications

- Connectors might be enhanced to allow the combining of non-Unicode, unilingual or limited multi-lingual output data from multiple application invocations into a single multilingual Web page presented to an end user
- Other connectors support Java programming access to legacy applications via APIs delivered as part of the connector implementation. Globalization is limited generally by the capabilities of the legacy end

Legacy Data and Applications con't.

- Present connectors and the associated programming might become more globalized in the following ways:
 - ▶ pervasive use of Unicode, especially in client/server interactions
 - ▶ support for single executable with pluggable country packs
 - ▶ support for text processing functions such as transformation from logical to visual order of bi-directional data and locale-sensitive normalization and collation

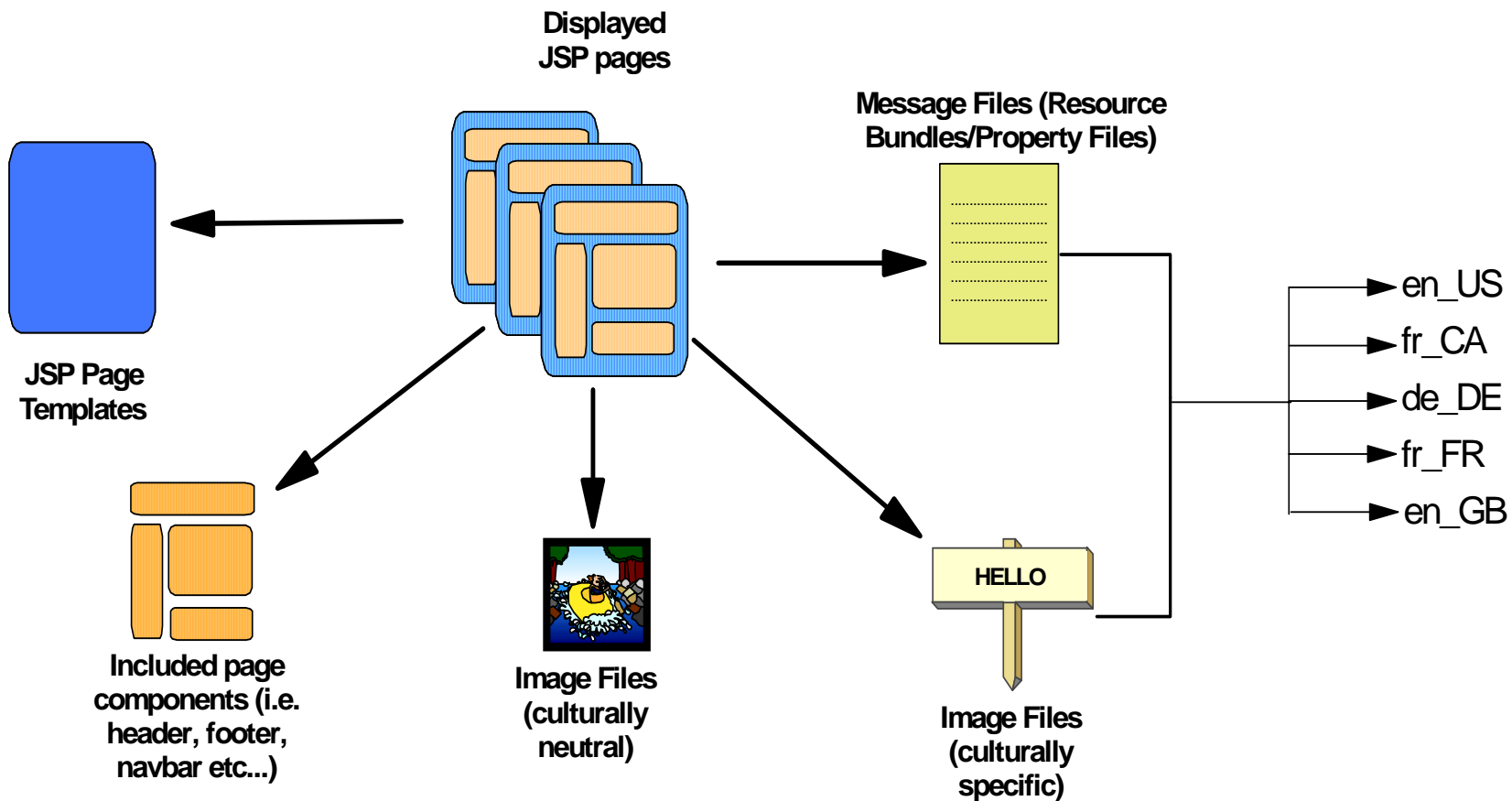
Legacy Data and Applications con't.

- Migration strategies for legacy data and applications:
 - ▶ migration from a non-Unicode database to a Unicode database can be done
 - ▶ customers can migrate their applications to Java applications or applets and use JDBC to connect to any database
 - ▶ migrate ODBC applications to Unicode and use an ODBC Unicode driver to connect to a DB2 database
 - ▶ Upgrade the COBOL environment (compiler, platform, APIs, etc.) so that Unicode can be processed as a native encoding system roughly equivalent to EBCDIC

WCS Globalization Methods

- Single page template for all languages
- Business processes
- Catalog data
- Data encoding

Globalized Page Framework



Globalized Page Framework

- Retrieving Static HTML Pages and Images
 - ▶ Static HTML text will be stored in directories that will be qualified by a locale.
 - ▶ At runtime, the page will be retrieved depending on which language the shopper is shopping in.

```
<Store Document Root>/en_US/page.HTML  
/fr_FR/page.HTML
```

etc.

- Multi-lingual Static Text

- ▶ All static text on a JSP page will be retrieved from Property files

```
ResourceBundle myResource=ResourceBundle.getBundle('messages','locale');  
where the property files would be:  
messages_en_US.properties etc.  
messages_fr_FR.properties
```

Globalized Page Framework

- Dynamic Data
 - ▶ Dynamic data would be retrieved from the data using various Server bean methods.
 - ▶ Retrieval of data is automatically performed in the context of the current language

```
<%String languageId = request.getParameter("lang");  
ProductDataBean product = newProductDataBean();  
DataBeanManager.activate(product, request);%>
```

```
<product>product description here: <%= product.getDescription().getshortDescription() %></product>
```

Globalized Business Processes

- Taxation and Shipping
 - ▶ Taxation and Shipping wizards allow the definition of rules based on jurisdiction
 - ▶ The data model in v5.1 has been improved to allow a better integration with third party taxation software
 - ▶ Concept of fulfillment centers has been added to allow for a clearer definition of ship-to and ship-from costs

Globalized Business Processes con't.

- Multiple currency support
 - ▶ Merchants can specify prices in each currency
 - ▶ If system can't locate price for a currency, currency conversion will be attempted
 - ▶ A shopper can switch the preferred currency in the middle of shopping. The system will update the price of items in each pending order to reflect the price in the new currency
 - ▶ Current preferred currency is stored in the session and will also be stored in a registered shopper account

Globalized Catalog Content

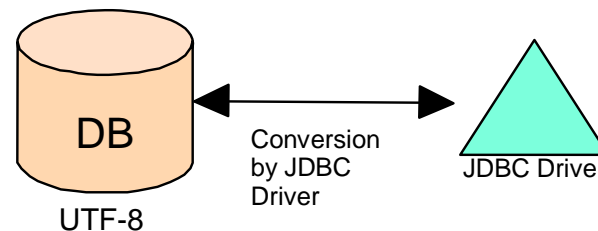
- Product and category descriptions, attributes and images may be varied across locations
- Multiple translations of catalog content, one for each of the supported locales
- WCS allows a merchant to manage catalog content for selectable display format

Data Encoding: Storage

WCS V5 is designed such that a single store can work with **any** language. A developer needs to follow a programming model to take advantage of this feature.

Data Storage:

- ▶ To support multiple languages in a product, it is necessary to be able to store data from each language in a single database
- ▶ Data is stored in the database using UTF-8 encoding. WCS configures the database at creation time to store data in UTF-8.
- ▶ A UTF-8 database has the correct character mappings to handle all single byte, double byte and bidi characters
- ▶ JDBC drivers load data from the database and store textual data in `java.lang.String`. To do this, the drivers convert text data from UTF-8 to Java's native 16 bit UNICODE encoding.



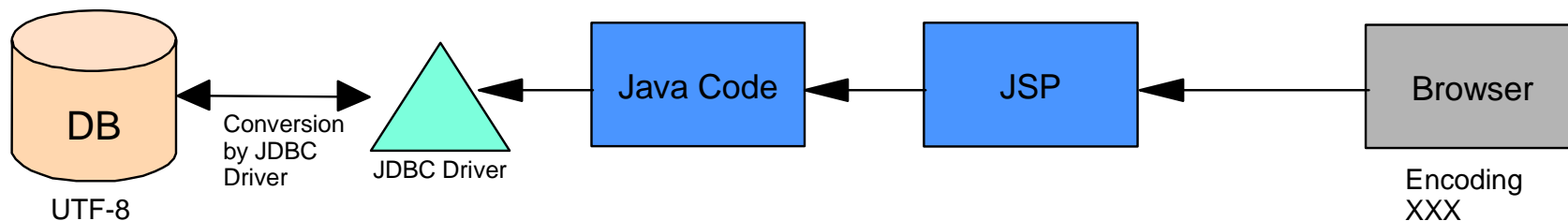
Data Encoding: Input

Data Input:

- ▶ Multi-lingual data can be entered through the browser using an IME (input method editor). This is available for most browsers and/or operating systems.
- ▶ Each language_id in the language table maps to an encoding value. This value is used to interpret the data going out to the browser.

LANGUAGE_ID	LOCALENAME	LANGUAGE	COUNTRY	VARIANT	ENCODING
-1	en_US	en	US	-	ISO8859-1

- ▶ Based on the current language_id, WCS will properly convert the data coming in from the browser into 16 bit UNICODE using the setCharacterEncoding() method.
- ▶ From there, the data is sent to the database, undergoes data transformation from 16bit UNICODE to UTF-8 (8bit UNICODE) and is stored in the database.



Data Encoding: Output

Data Output:

- ▶ Strings are handled inside Java code as UNICODE 16 bit encoding.
- ▶ JSP pages output strings using UNICODE 16 bit encoding (because JSP is 100% Java code).
- ▶ JSP authors can specify the encoding to be used to send the HTTP reply back to the browser. For example:

```
<%@ page contentType="text/html; charset=Shift-JIS"%>
```
- ▶ WAS will convert JSP output text from 16 bit UNICODE to the target encoding as specified in the JSP page and send the converted data back to the browser. For example:

```
Content-type: text/html; charset=Shift-JIS
```
- ▶ Browser will interpret the HTTP reply based on the encoding specified in the header and automatically switch to the encoding setting specified to display the data.

