

How To Internationalize Your Web Site

Contents:

- » [INTRODUCTION](#)
- » [Localization, internationalization and globalization](#)
- » [THREE TIER WEB ARCHITECTURE](#)
- » [WEB SITE STRUCTURES](#)
- » [Static HTML authored without using HTML editors](#)
- » [Static HTML authored using HTML editors](#)
- » [Dynamically generated HTML](#)
- » [INTERNATIONALIZATION](#)
- » [When to internationalize](#)
- » [Internationalization issues](#)
- » [Identifying internationalization problems](#)
- » [LOCALIZATION](#)
- » [Content style](#)
- » [Content location](#)
- » [Web resources - the REZ file](#)
- » [Pre-translation testing](#)
- » [Localization kits](#)
- » [REFERENCES](#)

INTRODUCTION

Historically, English has been the lingua franca of the Internet because the United States (and subsequently the United Kingdom) were well ahead of the rest of the world when it came to being "online". The rest of the world is now catching up; Japan, Germany, China, all have a large number of users and the Spanish speaking population on the Web is increasing rapidly. As a consequence, the Web is becoming a multicultural and multilingual entity where Web sites are becoming available in the native languages of the audience.

On the face of it, to provide each Web site in a local language is as easy as "just translating it". In reality, local markets have specific requirements from both technical and marketing aspects.

This document is intended to highlight some of the difficulties of providing Web sites that are tailored for any target audience. It also provides a view on how the Web site is internationalized and localized.

Localization, internationalization and globalization

It is a good start to define the difference between some key terms that are used in this document. These three terms are sometimes abbreviated to the first and last letters and an indication of how many letters there are in between.

Localization (L10N) covers anything that involves altering market specific aspects of a product before it can be competitively introduced in to another country. In its simplest form, L10N refers to the translation of strings within the product or Web site so that the user sees the correct language.

Internationalization (I18N) involves altering all the market generic aspects of a product to promote easier, cheaper, faster and higher quality localization efforts. I18N generally involves a one-time, up-front investment. In the context of this document, I18N refers to code changes that are made to ensure that a product (or Web site) can be localized and that information is presented in a format to which the end user is accustomed.

Globalization (G11N) involves making all company-wide preparations in order to enter the international marketplace. Basically, G11N covers anything that needs to be done differently to optimize international success.

The diagram below may help to illustrate the various definitions.



THREE TIER WEB ARCHITECTURE

As a first step to understanding the Web site design, consider a typical web page and identify the different layers that are available. Web pages typically consist of three layers:

1. User Interface (UI) - this is the portion of the program with which the user interacts. The UI is what the user sees and makes the page look good. The UI is the look and feel.
2. Code - this is defined as scripting (either client-side or server-side). Code makes the page interactive and makes the site fun. The code is the engine.
3. Content - this is the dynamically changing part of the Web site (the words on the page). Content can be either static (updated rarely) or dynamic (updated frequently). The content is the information. Content gets updated frequently to keep the site interesting.

WEB SITE STRUCTURES

To get a Web site published, there are three main methods that are immediately apparent. These methods are outlined below to illustrate the different approaches of Web design.

Static HTML authored without using HTML editors

By far the most basic approach to Web site design would be to author the HTML in a non-WYSIWYG environment and construct the page through trial and error. Authoring HTML in this way is time-consuming and error-prone, as there is no automation to ensure that the HTML conforms to standards. When the content needs to be changed, the HTML needs to be edited by hand.

Static HTML authored using HTML editors

The next step up to Web site design would be to use a WYSIWYG editor, of which there are many. Authoring HTML in this way is a much better proposition as the HTML will be correct and the final look-and-feel can be envisaged easily. The downside of these editors is that they sometimes introduce their own "quirks" such as new tags, to help with the editing. Even using these tools, when the content needs to be changed, the HTML needs to be edited.

Dynamically generated HTML

The majority of corporate web sites will probably be using dynamically generated HTML, which is then passed through to the browser. The most common way of generating this HTML is server-side scripting languages, of which there are many to choose from. Some of the better known methods are Active Server Pages (ASP), Java Server Pages (JSP) and ColdFusion (CFM). Some of these file types may be familiar to regular Web surfers.

[Back to Menu](#)

INTERNATIONALIZATION

This section covers many problems and resolutions for the internationalization of a Web site. Internationalization of the site pertains to the Web site Code and UI tiers.

When to internationalize

Past experience has shown that the best time to embark on the internationalization effort is in parallel with core site development and before the localization is started. The following list outlines some benefits of this approach:

Errors are found early in the process

Any errors are fixed by people who are familiar with the code

Developers learn how to internationalize a product and thus learn how to avoid any internationalization errors in future projects

Internationalization issues

Even when a Web site is destined for release in several languages or markets, it is still easy to overlook problems that can potentially be introduced. Note that one language (e.g. German) does not necessarily correspond to one market (e.g. Germany and Austria). There are many different internationalization issues - the number that need to be considered depends on the format and complexity of the Web site. This document covers some of the common issues for internationalization.

Text embedded in graphics

Where possible, the use of text in graphics is discouraged. Most localizable graphics consist of text on top of some sort of structured background. To localize graphic text, it is necessary to get access to the textual part of the graphic. Localizable graphics should be handed off in a package that supports "layering" so that the text portion of the graphic is on a separate layer and easily accessible for localization.

If text-embedded files are necessary, the following guidelines should apply:

- Provide a well-documented, layered source file with details of the fonts and colors used
- Keep in mind that text within the graphic is probably longer for localized languages than for English so allow room for the text to expand.

As an alternative to using text-embedded graphics, it is possible to have only the graphic as a background and position the text on top of it. For example, using the tag and forcing the position of the text to be "relative" to the graphic.

Symbols and other design elements

As part of the Web site design, it is necessary to avoid culture-dependent symbols that are not clear to an international audience. A classic example would be an American mailbox with a little flag to indicate that there is new mail. This symbol is used on many sites to indicate e-mail but people outside of North America don't necessarily recognize the mailbox. For a web site, a better symbol would be an envelope, which is universally understood.

There are also many symbols that may have different meanings in different cultures. If there are any doubts regarding the hidden meaning of some symbols, it is better to use words instead. As a general rule, the following should be avoided in any graphics used:

- Hand gestures or body parts
- Graphics with multiple meanings (e.g. a "pillar" to indicate a "column")
- Religious symbols such as stars, crosses etc.
- Shapes that are tied to culture (e.g. stop signs, sports, mailboxes etc.)

Locale-specific content

The following list provides some of the items that would need to be changed during internationalization. These items are often hard-coded but should in fact use the system settings for the user's environment, where possible.

- Date formats (including calendar settings and day/month names)
- Time formats (12-hour vs. 24-hour clock etc.)
- Currency formats and other monetary-related information (taxes etc.)
- Number formats (decimal separator, thousand separator etc.)
- Fonts (names, sizes etc.)

For fonts, it is best practice to use Cascading Style Sheets (CSS) whenever possible. CSS allows fonts to be changed for all the pages in one place, and there will be fewer tags within the text for the localizers to sift through.

Other issues for consideration, which may not have formatting rules specified by the user's environment include:

- Address formats (postal codes, states etc.)
- Name formats
- Telephone number formats
- Units of measure
- Paper sizes
- Use of color for meaning (e.g. red = stop)

Concatenated strings

A concatenated string is an error message or other text that is dynamically generated and presented to the user in sentence form. Consider the following ASP code snippet:

```
<%
if nSource = 0
sSource = "server"
else
sSource = "connection"
end if
%>
...
<P>The <%=sSource%> is currently unavailable</P>
```

In languages where the "server" and "connection" nouns have different genders, it is not possible to translate the above string as the translation of "The" would depend on the noun.

The correct solution is to have either multiple error messages that are translated separately, or to find a different way of presenting the same information. For example, to report the number of errors encountered:

```
<P>Process complete, <%=NumErrors%> error(s) encountered</P>
```

Even in English, this is not a good approach. Re-writing the code can make it flexible enough for multiple languages.

```
<P>Process complete. Errors : <%=NumErrors%></P>
```

Sort order

Sort order is not the same for all languages, particularly for languages that do not use the Western alphabet. In Swedish, for example, some extended characters (e.g. å) get sorted after the letter Z. In many Asian cultures, characters are comprised by a prescribed tradition of brushstrokes and characters are sorted by the brush stroke order. Also, after localization, the first letter of the word might change, changing its position in the sort order list.

To build an internationalized Web site, it is necessary to either find a way to automatically sort the items (this can be a very difficult task) or ensure that the localizers can change the sort order of the list while they are localizing the code. The optimal method for the end user is to allow the localizer to personally sort the list.

Text truncation

Truncation occurs when an interface is designed with a particular word or words in mind, and the word length varies in other languages. Any words that are longer are then truncated in the interface. This is a common occurrence on Web sites where space is at a premium and designers want to maximize the available real estate.

Design and development of the web site should consider that all the words and phrases could expand. The general rule is to allow large sections of text to expand by 30%. Single words and terms can expand by up to 400%. Having a flexible design is critical to the success of localizing a Web site.

Truncation can occur in a variety of places, but most noticeably within buttons, graphics and tables.

Dynamic data

When localizing a site, one of the first decisions to make is what character encoding to use. On the Web, this is generally divided into UTF-8 (an eight-bit representation of Unicode which covers the characters of most of the world's major languages) or native encoding (encoding specific to a language or set of languages). There are costs and benefits to each system available. Regardless of what encoding is used, it is necessary to ensure that the data displays correctly to the end user and any data sent to and from a form or database maintains data integrity.

Web site components could include databases, form elements, COM objects, JavaScript™, DHTML, ActiveX® controls, etc. Each component of the Web site should be considered carefully and the following questions addressed:

What am I putting in? (for example, JavaScript)
 What am I getting out? (for example, from a database)
 Is what I put in what I get out? (for example, form elements)

The answers to these questions may not be the same for different elements on the site. If all strings for several language versions are in a single database, it may be necessary to use Unicode or UTF-8. If the HTML has been encoded natively, how will the data be transformed so that the user can read it on the Web site?

Identifying visitors

If the Web site is to receive visitors from many geographical areas around the world and it is required to provide a personalized service to everyone, it may be necessary to employ some server-side scripting techniques to achieve this. ASP, JSP and XSP can all provide this level of functionality.

Identifying internationalization problems

Having thought about the design of the Web site with a view to internationalization, there should be an approach that allows the identification of internationalization bugs.

Specifications

Sometimes just reading a functionality or technical specification from the point of view of an international user can reveal a number of issues. For example, a technical specification that states that a Web site will display the date and time it was last updated. This data will be collected from the server date and time. However, some countries are on the other side of the International Dateline so any date displayed may make the site appear at least a day older than it is. A better solution in this case would be to calculate this date based on the user's time, whereby the content would appear fresh.

Test Cases

One of the best ways to analyze whether a Web site is internationalized is by writing test cases. Obviously, this is best when there are dedicated international testers. Writing test cases allows focus on smaller and smaller units of the project. As test cases are written for each component, consideration can be given as to how others would use it. In addition to uncovering problems at the internationalization stage, it is possible to re-use these test cases at other stages of the project - particularly once the files have been localized.

LOCALIZATION

This section covers many problems and resolutions for the localization of a Web site. Localization of the site pertains to the Web site Content.

Content style

The target audience for a web site wants to get in and out quickly, with all the information they need. Localization services are potentially expensive and are paid by the word. The Web site needs to get to the point. By looking at all the content and deciding which information is most important, decisions as to what appears at the top-level of the page can be made.

If possible, it should not be necessary for the user to drill down deeper than two levels in to the site. The mood of the content should also be considered; not all humor translates from language to language, or culture to culture. Some language that's merely informal to one person could be considered rude in other parts of the world. To be concise, it's probably better to stick to the facts.

The following is a list of some issues which should be considered for the site content:

- Words with multiple meanings
- Abbreviations
- Mnemonics
- Acronyms
- Telegraphic style
- Slang or jargon
- Gender
- Creation of new words
- Shortened plurals or word combinations
- Anything that portrays a way of life or culture specific to one country

Content location

Any text displayed to the end-user should be centralized to make the content easier to localize. The simplest method of achieving this for the majority of the content is to use a back-end database to store the textual information for the site.

Static content

Static content can be held in the Web site pages (HTML, ASP etc.) as this will not change at regular intervals. Localization of the static content would typically be done using an HTML editor or translation tool that handles Web pages.

Dynamic content

Dynamic content is best held in a database for ease of maintenance. Localization of the dynamic content can take a variety of forms, the most natural of which would be a defined process for identifying updated content and automatically routing this through a pre-defined workflow.

Code content

If there is localizable text within the Code tier of the site, ensure that it is commented as much as possible. Localizers are generally hired for their translation skills, not programming skills. The easier it is for the localizer to identify the localizable text, the better. If the Web site contains scripting that needs to be localized, hire a localizer with sufficient experience not to inadvertently alter the script of the Web site.

Web resources - the REZ file

A more robust solution for localizable text in code would be to externalize the text from the code. Outlined below is a possible approach as to how this could be done.

Overview

Essentially, translatable text is placed in a text file that is included in the server-side scripting file with a statement like:

```
<!--#include file="text/abc.rez"-->
```

The basic format of these REZ files is:

```
[type] id = quoted text
```

type is optional

id is a programmer's identifier

quoted text is either single or double quoted text for translation

Each module of the site would have a text folder where these include files are kept, i.e. the REZ files are kept in a text folder off the main website folders.

Advantages

The advantages of using this approach are:

- It puts the emphasis on the developers to code internationally (i.e. easier to translate giving a cheaper and quicker turn-around)

- It removes the need for engineers to review code. This is important, as even with a smart script filter you cannot remove the need for an engineering review. Making your client code properly is the safest, cleanest and most cost effective way for SDL to avoid having to touch client code. It sometimes takes a long time to track down whether a quoted string needs to be translated or not - you sometimes have to backtrack through several files to see where and how a string is used. By externalising the script text into resource files - the vendor can avoid having to check script code. The resource files can also be commented to add context to the strings to aid the translators.

File format examples

Java Script (.JS) include file:

```
var NewMail = 'You have new mail in your Inbox';
var WarnDelete = 'Warning: This action will delete all files';
var PromptContinue = 'Do you want to continue?';
var ErrorUpload = 'You may only upload one file at a time.';
```

ASP Script (.ASP) include file #1:

```
ButtonNext = "Next"
ButtonPrevious = "Previous"
ButtonResubmit = "Resubmit"
ButtonConfirmChanges = "Confirm Changes"
```

ASP Script (.ASP) include file #2:

```
<%
Const NoValuesEntered = "No values entered."
Const AnlComplete = "Analysis complete. Click Next to continue."
Const WrongFormat = "Warning: Expecting REZ format, found "
Const Points = "(N)orth (S)outh (E)ast (W)est"
%>
```

Concatenation restrictions for Visual Basic

To make it easier to write automated tools for localization of these files, it would be prudent to write VB concatenation in the following style:

```
NL = Chr(13) & Chr(10)
MsgText = "Thank you for visiting SDL International." & NL
MsgText = MsgText & "We hope you found the site informative. "
MsgText = MsgText & "Please come back soon." & NL & NL
```

This is easier to handle than

```
NL = Chr(13) & Chr(10)
MsgText = "Thank you for visiting SDL International." & NL & _
"We hope you found the site informative. " & _
"Please come back soon." & NL & NL
```

Pre-translation testing

Pre-translation testing is the process of exercising the site's user interface, localizability, and site stability before localization. This is done by quickly editing all of the strings in the project to:

- Include some extended characters (e.g. é, ñ or ö) or Asian characters
- Increase the length of the terms and paragraphs

During the design phase, it may be desirable to run a prototype site through pre-translation testing to ensure that the design is flexible for all the terms to be translated. For more complex sites, pre-translation testing can be used to test dynamically generated data or to ensure that the controls can display extended characters correctly.

The benefit of pre-translation testing is that the process can be iterated or run at different stages of the project, to identify and resolve international issues without wasting the time of the localizer (for which the localizer will surely charge). Pre-translation testing can also save money, additional internal effort, and time to market by avoiding the need to fix bugs later in the project. This can often have an effect on the overall quality of the international product as well. Pre-translation testing would be used to test:

- String truncation
- Whether all the strings are accessible to the localizers
- Whether keyboard shortcuts can be localized
- Characters displaying correctly in HTML and on all controls/elements of the Web site
- Characters displaying correctly in and out of a database

It should be remembered that pre-translation testing can't test everything, this is where human testing and running a pilot project can be very useful.

Localization kits

Why write localization kits?

To ensure that the Web site is localized correctly, it is necessary to provide instructions for the localizers, testers, and engineers. Localizers need information on what to localize, for which audience they localize and, in most cases, what not to touch in the files.

Who to write the localization kits for

As mentioned above, the localization kit is aimed at localizers, testers and engineers. When the localization is outsourced, the project program managers may also need instructions. The style and content of the kit should reflect the target audience. For example, for project managers it is not necessary to go into the details of what needs to be changed. Project managers will be interested in an overview, the number of files and the quantity of words to localize. Engineers would prefer the instructions to be to the point and in their lingo.

How to write localization kits

Details above should provide some guidance as to how to go about writing a localization kit. The following general steps also apply:

- Prepare the project (internationalization)
- Research the hurdles (normally this would go in a specification)
- Identify the scope (file list, word count etc.)
- Identify the target audience (technical, non-technical etc.)
- Write instructions for each specific group of people working on the project
- Running a pilot project would help to test the localization kit.

REFERENCES

- [Developing International Software](#) by Nadine Kano
- Designing a Globalized and Localizable Web Site by Sjoert Ebben and Gwyneth Marshall
- The Localization Process: Globalizing Your Code and Localizing Your Site by Sjoert Ebben and Gwyneth Marshall
- Going Global: Not for the Halfhearted by Tapani Tuominen
- How Wide a World? Localizing Web Sites by Amy Burns

[This document has been published courtesy to [SDL International](#). All rights reserved to its owner]