

Localization Skills for Translators/Localizability Requirements for Clients

Author: By Carla DiFranco
E-mail: carladi@microsoft.com
Date: January, 2003
Source: The ATA Chronicle

Localization Skills for Translators/Localizability Requirements for Clients by Carla DiFranco

Material for this article was drawn from a talk the author presented to members of the New York Circle of Translators on October 22, 2002. A portion of this article also appeared in the December issue of the Gotham Translator, the NYCT newsletter.

Localization:

"Customizing software for a particular country. It includes the translation of menus and messages into the native spoken language as well as changes in the user interface to accommodate different alphabets and culture." (www.techweb.com/encyclopedia). The oft-used term "localization" may seem daunting to the modern translator interested in working in the localization field. There is certainly a special skill set involved in localization, but where should one start? The intention of this article is to outline some of the basic localization skills that translators will need to complete their tasks. In addition, we will examine the localizability requirements for clients (defined here as the technical planning for different locales that clients can do to ensure the accurate localization of a product or document). As you will see, localization is not necessarily just a job for the individual translator (or localizer, in this case). Rather, it is the responsibility of all parties involved in the development of the original product,

as well as the duty of project managers involved with the localization process on all sides. By meeting some basic localizability requirements as a product is being developed, there is a positive domino effect along the food chain. A well globalized product will be easier to localize, mainly because it is less prone to technical issues during the localization process. Also, clearly structured content is simply easier to translate.

Localization Skills for Translators

As a translator, you are the language- and subject-matter expert all rolled into one. Added to this, you may sometimes fill the shoes of the editor or proofreader. These skills will all come in handy when applying them toward localization. Because the field of localization is quite complex, it is only possible to name a few key pieces of technical knowledge here, but these should help increase your marketability and skill level. For the purposes of this article, the term "localization" will refer to software localization. This generally includes the user interface (UI) and documentation (user assistance, user education, help, etc.)

HTML

For many translators, HTML may already be common knowledge, but some of us don't have the opportunity to work with

these types of documents. Understanding how HTML works is a key localization skill. Being able to troubleshoot issues when they arise is also an invaluable skill clients increasingly rely on. If you are able to provide this service, it can only add marketability to your already valuable skills. This is not to say that all localizers should be web programmers! However, localizers should have a general knowledge of markup languages. Luckily, they are relatively easy to learn. As a localizer, you should:

- Know enough HTML so that you can recognize tagged content in an HTML document.
- Most HTML tags have a starting and ending tag, and are often nested in layers (think of how an onion looks when it is peeled). However, this may not always be the case. HTML is not as strict as we would like it to be, and it will still work with missing tags.
- Clarify with your client which tagged content is to be localized.
- Know how to recognize scripting in HTML files, such as Java script or an embedded cascading style sheet (a CSS is a data format used to separate style from structure on web-pages). Once identified, it will be easier to ask your client which parts are to be localized. There may be translatable content in the Java script, and the CSS may need to be edited for East Asian languages because of font face and size changes.
- Invest in a good HTML editor. I cannot stress this enough.
- WYSIWYG (What You See Is What You Get) editors create an editing nightmare!

There are a number of professional editors out there, such as Macromedia's HomeSite, which enable editing of HTML code without inserting other unwanted formatting.

- Translation tools are created to help you navigate translatable content in HTML. CAT (Computer-Assisted Translation) tools lock tags and other HTML code that you don't need to touch. Of course, this makes the HTML experience much easier.

It is relatively easy to ramp up to HTML. There are a number of books on the market that explain how it all works, and I have invested in the "For Dummies" series in the past. For those who have a basic understanding of HTML but need more information, the O'Reilly series provides a great reference. There are also a number of excellent (and free) tutorials online, as well as sites that provide lists of HTML tags (this is a finite list) and tips on how to create sites of your own. Many local community colleges also offer courses on basic HTML programming.

XML

XML is an excellent choice for localizable content because of its flexibility. XML will not replace HTML. Rather, it can be used as a vehicle for providing content for the web, print documentation, or other delivery formats, potentially at the same time. The difference between the two is in the markup language. While HTML applies structural markup (such as `` for bold, `<p>` for paragraph, `<h2>` for a second level header tag, etc.), XML applies semantic markup (such as `<emphasis>`, `<article>`, `<product-name>`, etc.). The elements in XML are

written out rather than abbreviated, so there is no confusion about what these should stand for. Like the HTML document, the XML document does not include any formatting information, since all that information is called from a separate file (an XSL or a CSS). XML is extensible, which means that elements can be created by the author as the situation deems necessary. Formatting can be changed for different locales. As an example, if an `<emphasis>` element were around some text, the XSL would stipulate exactly how to format the contents of that element. This means that bold formatting could be applied for an English locale, or that no formatting could be stipulated for a Japanese locale.

Here are some basic concepts about XML that are important for translators to know:

- Understand how XML is structured:
 - The structure of XML will be much more explanatory than HTML, so it will be easier to recognize translatable content.
 - Formatting information should not be in the XML file. If the structure is created properly, any formatting information that is attributed to a particular set of files should only need to be changed once.
 - It is imperative to always hand back valid XML files to your client. Validation tools are readily available on the market, as are XML editors.
- Translation tools are created to help you navigate translatable content in XML. Just like with HTML, CAT tools lock elements and other XML code that you don't need to touch, making the XML experience much easier.

Character Sets

At some point we have all experienced a webpage that displayed strange characters

or boxes where there should have been a diacritical. Perhaps some diacritical characters were replaced with a Japanese character. This happens because the computer is looking for one character set when it should be looking for a different one! Knowing a bit about how character sets work will help you troubleshoot and understand why things function the way they do. Originally, each character was mapped to a number on a code page. This would have been a good system if the entire world had used the same characters, but the problem was that there were hundreds, if not thousands, of code pages that all happened to use the same numbers for different characters. Since a computer is only trained to understand numbers, it needs to be told which code page to read when it is reading text.

In this example, the HTML page is looking for a specific code page: `<HTML>`

```
<HEAD>
```

```
<META content= "text/html;charset=1252">
```

If this number were changed to a code page for Hebrew (1255), then many of the characters in the document would show up as Hebrew characters, in accordance with the numbers assigned to each by the system. The computer is merely doing what it's been told. There is a solution to this mess, however.

Enter Unicode.

Unicode

Rather than using thousands of different code pages for all the languages in the world, the idea behind Unicode is to use only one code page for all languages. This should include all the characters used for written languages in the world, which can encompass more than a million characters! Unicode includes alphabets,

ideographic characters, and symbols. Ideally, these characters can be mixed in any way, since they will all live on the same code page (see www.unicode.org). Here are some basic things you should know:

- Understand how you can tell whether a document is Unicode, or whether it is calling an ANSI code page. This is easy to detect by looking at the very top of the document source. If a document is Unicode, either UTF-8, UTF-16, or UTF-32 will be listed there.
- Unicode calls the same code page for all languages.
- Unicode is a standard agreed-upon system. Check the Unicode.org site for more information on the consortium. There are quite a few publications that deal with the subject of Unicode, and the Unicode website is also a wealth of information. In short, it is an easy subject to bone up on. If you are more knowledgeable about character sets and how they work, you may be more capable of troubleshooting and assisting your clients when these issues arise.

Translation Tools

Also known as CAT tools, translation tools help streamline workflow and are also designed to increase productivity. Like any tool, there is generally a learning curve, which normally reaches its peak about 3:00 a.m. when the deadline is five hours later. Each brand of translation tool has a slightly different interface and functionality, but they all do more or less the same thing. Here is a short list of basic skills you should have regarding translation tools.

- Understand how to manage a translation memory (TM). A TM that records your source and target as you translate, then collects source-target pairs for each

segmented phrase, sentence, or paragraph, and stores this for future use.

- As the localizer, you are the language expert. Your clients may depend on you to update strings or terminology in the TM and to ensure that the information is current.
- Understand how different merge functions work with TMs. You may need to merge more than one TM into another.
- Know how to create a well-formed terminology database. The terminology database may be for personal use or intended as a deliverable. In either case, there are standard fields and methods for creating this database that may need to be followed.
- “Translation tools” refers to a wide array of tools that are used to translate and edit text. Know how the tools work together and how to best use them in your workflow. Each localizer has a slightly different work method, and the process can be manipulated for various text types and projects.
- Machine translation (MT) refers to a fully automated translation process that is performed by the computer. Often these programs are only worth what has been put into them, and good programs are engineered to work together with human translations.
- Understand the word-counting and analysis process. Your clients will be running word-count analyses before files are handed off. In order to avoid surprises, you should be able to perform this function as well.
- Know how to localize tagged format files using translation tools. It is important to know how to lock certain

tags and elements from localization. If you are well versed in this process, your clients will rely on your expertise.

The ability to work effectively with translation tools takes time. It is much easier to see the savings after you have used a particular tool for a while and worked through some trouble spots. The key to these tools is to understand how they work. As with any new piece of software, time is generally your best ally. If you are interested in training, there are many options available, including vendor training, tools workshops, translation programs, publications, and online groups and forums. After you have mastered the process, you will be able to spot potential problems and issues as you work. Knowledge of the major and minor points about file formats, encoding, and tools is a great first step toward learning the localization process. The skills outlined in this article are merely the tip of the proverbial iceberg. There are a number of highly complex and important issues that surround the field of localization, and understanding the basics will help you build upon these skills for the long-term.

Localization Requirements for Clients

Generally, the focus is on what translators and localizers must know to survive and succeed in the localization industry, but this is not the complete picture. A final localized product is not purely the responsibility of the localization project manager to organize and the localizer to translate and fix. The responsibility for a well-localized product rests on the shoulders of every person who has developed, managed, and, ultimately, localized that product. Rather than the full responsibility of localization occurring at the end of the food chain (with the freelance translator or localizer), certain things should

be put into place long before localization begins, so that the localizer's job is focused on his or her expertise. In this way, localizers can be more productive, as well as less randomized by all the other issues that pull them away from their work. In the long run, this means that the total localization effort will be less costly and more streamlined.

Currently, the localization process seems to take place on two tracks (see Graphic 1). On one side, there is the product developed for the English-speaking markets, and on the other, there is the localized product. The process for both these products is very different. So that the localized product will work properly, many steps are added before and during localization. For example, consider what would happen if the product was not properly tested, if a different character set did not work in the product, if the content had to be adjusted for the target locale, if the translation tools didn't work on the source files, etc.—it is easy to see how this can become a veritable nightmare. Every project manager has had a project like this one, and it is amazing how well we learn the lessons that our clients should have figured out a long time ago. This all goes to prove that a little globalization goes a long way. It is easy to see, oftentimes after the fact, that some key planning could have been worked into the beginning of the product cycle.

Localizability Requirements

The globalization process that takes place upstream on the client's end can also be referred to as the "localizability" requirements. The following list of requirements is by no means a short one, but I would like to mention a few here that will give an idea of the types of things to watch out for.

Translatable Content/Strings—Keep it Simple

The computer and high-tech industries are filled with jargon, catch phrases, and slang that we tend to use every day. While this is fine for hallway conversations or e-mail, this turns out to be quite a costly endeavor for localization. Standard English should be enforced from the very start as a product is being developed or documented. An interesting example of a term that is currently being used in documentation is presented in Graphic 2.

This text was taken from the Release Notes of a product, the full text of which can be found at www.hyperware.com/e/downloads/documents/releasenotes.pdf. This is an excellent example of modem jargon in documentation. While the meaning of this may be apparent to some, it will not be apparent to all. I quote the entry for this term at www.wordspy.com, which states:

Breadcrumbs

Noun. A navigation feature that displays a list of places a person has visited or the route a person has taken. Backgrounder: Today's word is based on the fairy tale of Hansel and Gretel, who threw down bits of bread to help find their way out of the forest. This feature is common on websites where the content is organized as a hierarchy or as a sequence of pages. Yahoo! (www.yahoo.com) may be the most famous example.

While this passage would be understandable to a Western European localizer (the target audience would most likely have a cultural reference to the fairy tale here), such jargon would be totally unacceptable to an East Asian localizer. The time and effort it would take to translate this term into the language of the target locale would be costly (research time, queries to project managers and decision-making over the term, etc.)

Even if all terminology used in a product is well formed and clear in meaning, other issues that often come up are a lack of consistent terminology between the UI and the help documentation. This is another area where potential queries and misunderstanding will cost time and money. Additionally, any differences in either the English or the localized product will have very negative effects on the user experience, potentially harming the product or the company name in any country.

How are the files created or edited?

Aside from the translatable content, there are many issues that also may crop up as a result of the file types used for localization. It may seem like a shortcut to save a document as HTML in Word. However, when it comes time to edit or translate this document, the residue of this change is definitely not pretty. A great deal of extra formatting is inserted in the HTML document, and detecting the translatable text suddenly turns into a needle-in-a-haystack chore. If an HTML file is going to be localized, it should be created using an HTML editor that does not leave behind unnecessary **tagging**. Many WYSIWYG editors will enter extra text into an HTML document, but there are simple editors on the market that do not. Investing in a tool such as HomeSite (www.riiaeromedia.com/software/homesite/) is much less costly than hours of reformatting. Also, the cleaner the documents are in the source language, the fewer errors will be encountered during and after localization.

The content of HTML files is generally straightforward, however, when scripting is introduced to the file, this may become

problematic if it is not implemented properly. If Java script is used in an HTML file (which is often the case), there may be localizable text embedded in the Java script. Although the Java script section of the document is clearly marked, it may not always be as easy for a localizer to pick out localizable text from lines of Java script. A better option in this case is to provide a *.js file that the individual HTML files will reference. Within this file, localizable text should be separated out from the actual scripting. In this way, localization is quicker and less error prone.

By the same token, if a CSS is embedded in a set of HTML documents rather than existing as a separate file, font information in the CSS may need to be changed. For East Asian languages, this must be changed in every file rather than in only one place. The time used in changing fonts can be better spent working on the localization of the content.

XML Content

This type of content offers a number of excellent options for localization. Because of the extensible nature of XML, elements that represent the content can be created and used to allow different attributes depending on the language of the file. For example, any XML schema that is created for any source language should also include elements that may be needed for the language of the target locale. Often a product that is sold worldwide will have different features for different locales. XML provides the ability to identify extra content that may be needed or removed for these locales—while still maintaining a single source document. For example, if an East Asian locale needs extra documentation on a particular feature in the product, an extra element for this content could be created. This content could be locked for the languages that do not need

it, but opened for the languages that require this extra text.

Feedback

The issues raised here concerning the quality and consistency of the source documentation may seem like common sense. From the perspective of the localizer or project manager, it may seem self-evident. However, it is amazing to see the number of poor choices or mistakes that continue to be repeated. Hindsight may be 20/20, but affecting a change in how products are created is not out of the realm of possibility.

In order to do this, a feedback loop should be put in place between all parties involved with a localization project so that costly mistakes can be avoided in the future. Not only are issues such as those described here costly to fix over and over again, they also take valuable time away from the localizer's real task. If the localizer is able to better concentrate on his or her expertise, the result will often be a higher quality localized product.

Partner Together

In order for these issues to be solved at the start of product development, it is vital that feedback be implemented. Above all, this feed-back should make it up the food chain. Localizers should be communicating with their project managers, and project managers should be communicating with their clients. After projects have been completed, it is a good idea to discuss the good, bad, and the ugly with all parties concerned. Only then will the process begin to change.

In an ideal world, localization companies will partner with their clients. Such a partnership benefits the clients, enabling them to produce world-ready software

and documentation. Using the expertise that their localization partners provide, projects can become less costly and easier to localize in the long run. By the same token, localization project managers cannot provide this type of information without the expertise and feedback of their free-lance localizers. In this scenario, rather than localization occurring as an afterthought or a nuisance, it will be simply the last step in the process toward creating a global product. Just the few issues raised here, when applied to real-life projects, turn the localization loop into a chaotic and costly enterprise. A project that started out so innocently and well intentioned can end up costing way too much time and energy. With well informed localizers, project managers who can apply past solutions to issues in new projects, and clients who understand the value of good localizability planning, there will always be plenty of benefits to go around.

Notes

0. For example, the Kent State University Institute for Applied Linguistics and the New York University Department of Foreign Languages and Translation offer courses in technical translation, project management, and translation tools.
1. A Practical Guide to Localization by Bert Esselink; The Handbook of Terminology Management by Sue Ellen Wright and Gerhard Budin; XML Internationalization and Localization by Yves Savourel; and Beyond Borders: Web Globalization Strategies by John Yunker. All these references have chapters or articles on translation tools.

Some Resources

Dr. International. 2002. *Developing International Software, Second Edition*. Redmond: Microsoft Press.

Esselink, Bert. 2000. *A Practical Guide to Localization: For Translators, Engineers, and Project Managers*. Amsterdam: John Benjamin.

Graham, Tony. 2000. *Unicode: A Primer*. Hoboken: John Wiley & Sons.

Macromedia HomeSite
(www.macromedia.com/software/homesite/).

Savourel, Yves. 2001. *XML Internationalization and Localization*. Indianapolis: Sams.

TechWeb Encyclopedia
(www.techweb.com/encyclopedia/).

The Unicode Home Page
(www.unicode.org).

Wordspy
(www.wordspy.com).

Wright, Sue Ellen and Gerhard Budin. 2001. *The Handbook of Terminology Management: Application-Oriented Terminology Management*. Amsterdam: John Benjamins.

Wright, Sue Ellen and Gerhard Budin. 1997. *The Handbook of Terminology Management: Basic Aspects of Terminology Management*. Amsterdam: John Benjamin.

Yunker, John. 2002. *Beyond Borders: Web Globalization Strategies*. Indianapolis: New Riders Publishing.