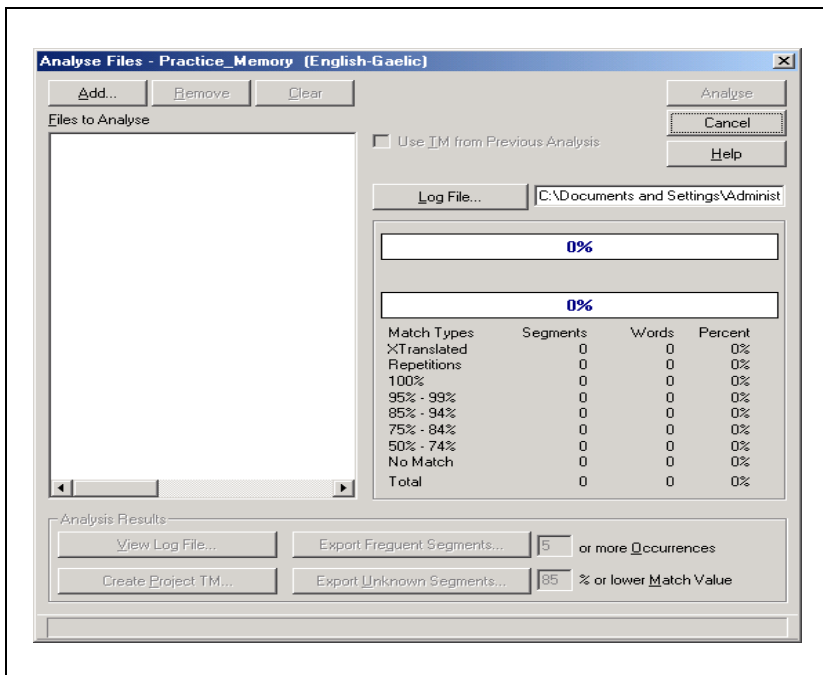


4

Documentation Localisation

Notes:



Overview

The focus of what I have been asked to discuss in this presentation is the localisation of documentation. This covers a multitude and would not be possible to cover in the brief time allowed. However, in the 90 minutes available to us I will, at very least, touch on most of the concepts involved.

Documentation has traditionally been divided into two categories:

- Online Help
- Printed Documentation

However, in recent times the this definition has widened to include all material that is delivered over the Internet. The reasoning for this is quite simple. The individuals involved in the preparation and localisation of Printed manuals and Online Help were already familiar with Translation Memory, Multimedia, Graphic Design, and Web Technologies. This meant that they were ideally placed to manage the localisation of eContent.

Over the course of the next 90 minutes, I will discuss the basics of documentation localisation and will touch on the following areas:

- Translation Memory
- Help Testing and Engineering
- DTP and Localisation
- Creating Localisable Documentation
- The Case for Single-Sourcing

Hopefully, I can give you a feel for the multiplicity of roles that a UA Engineer has to fill. If you already occupy this role within your company, I am sorry but it is likely that what I have to say is not going to be new to you. I may however, be able to offer you some alternative ways of doing things. If you are a technical writer, this session will hopefully raise some questions that you will take back with you to your company. With a bit of luck it may add some additional criteria to your list the next time you decide to change an authoring tool.

Authoring Tools and Technologies

I cannot start this discussion without first discussing the tools of the trade for the people who create documentation. Let's first address the technologies.

Notes:

Documentation

The tools used to author Printed Documentation have been around for quite some time and haven't changed all that much in that time. Adobe FrameMaker is still the dominant player in the Printed Manual market. FrameMaker is an excellent tool and well worth the effort of learning it. However, it is a steep learning curve. It is a complex tool with an unfriendly interface.

MS Word is getting more and more complex with each release. It has captured a large portion of the Technical Publications market and its market share is growing all the time. However, Word has issues and its not just that I am rarely "Writing a letter". There are significant issues with its capacity to handle ordered and unordered lists. There are also compatibility issues between Word 97 and Word 2000 that can have a significant on the general well-being of DTP professionals.

Other than MS Word and FrameMaker, there aren't any major players. However, InDesign is becoming more common. QuarkExpress and PageMaker are used frequently for smaller, often marketing-type, documents.

Help

There are a wide range of Help technologies varying depending on the platform used to deploy the Help and also the preference of the Help author. Examples of Help technologies include:

- MS Online HTML Help
- MS WinHelp
- Java Help
- Oracle Help
- Cross-platform HTML-based Help, e.g., RoboHelp WebHelp

All of these Help formats are generated using one of many Help authoring tools. The most common of these being RoboHelp.

Versatility

In addition to Help and Doc tools there are a wide range of multimedia tools that localisation documentation specialists must be familiar with.

Notes:

Translation Memory

A translation memory is in its simplest form a database where a translator may record old translations for future reuse and easy searches. Although these programs are best classified under computer-aided/assisted translation, one must not confuse them with machine translation programs - translation memory software does not translate anything by itself, whereas a machine translation system actively produces language and translations based on linguistic data, such as grammatical rules and glossaries. A translation memory system leaves all the actual translating to the human translator.

Basically, translation memory records sentence pairs: a source language segment (usually a sentence) is combined together with a target language segment. If an identical (or similar) source language segment comes up in another translation later, the translation memory program will find the previously translated segment and suggest it as the basis for the new translation. The suggestion may be either approved as it is, edited to match the current situation or rejected entirely. Most programs use a "fuzzy matching" algorithm, which makes the programs ever more useful; searches will find not only hundred per cent matches but also segments that only resemble the search phrase to a certain extent.

There are a great deal of similarities between the various software packages. It is possible to search not only for full segments, but also single words or phrases, which helps the translator to find terminology and keep the text consistent. Also, there is usually an interface to a terminology management program - usually the manufacturer's own, sometimes third-party programs too. There are translation memory programs that work together with machine translation software as well. Some work directly within an existing word processor, others (most) have their own editor. All of these packages feature filters for importing and exporting various file formats. Most, if not all, also provide an alignment tool for adding older translations (not done with the software) into the memory for reuse.

Why use TM

Most documentation projects are updates. The material is not written from scratch. Similarity between releases may be in the region of 70 to 80%. Translation Memory technology allows us to leverage all of the previously translated material, and to maintain consistency of style and terminology across releases.

Notes:

A note on TMX

TMX stands for Translation Memory eXchange. It is a vendor-neutral, open standard for storing and exchanging translation memories created by Computer Aided Translation (CAT) and localization tools. The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process.

Notes:

Preparing Documentation for Localisation

Preparing documentation for localisation can be divided into two broad categories:

- Documentation Preparation
- Help Preparation

Preparing Documentation

The most commonly-used DTP applications are FrameMaker, Word, Interleaf, QuarkXpress, and PageMaker. Of these, FrameMaker and Word are the most prevalent. With the exception of Word, these formats cannot be directly processed by Translation Memory tools. Filters must be used to convert native DTP files into a Translation Memory-compatible format - usually RTF.

The functions of a filter are:

- To convert the native DTP file to a format which can be translated using a TM tool. This format is usually RTF.
- To preserve style and structure information
- To enable the conversion of the translated file back to the native file format with the minimum amount of post-translation DTP work.

Prior to applying a filter a number of DTP preparation steps must be performed - these steps vary depending on the DTP package, and depending on the particular Translation Memory tool which is being used. For example, if Trados is being used prep for FrameMaker files includes:

- Verifying that the TOC and Index generate correctly
- Verifying that all necessary fonts are available
- Removal of change bars
- Verifying that all graphics have been imported by reference
- Removal of discretionary hard-returns
- Verifying that all files (including graphics) are present
- Saving of all FrameMaker files to MIF format

These steps are performed in order to minimize the amount of post-translation DTP work which will be required, and also to optimize segmentation during translation.

Once these steps have been performed the Trados FrameMaker filter - S-Tagger for FrameMaker - can be applied. The filter converts the MIF files to an RTF format known as STF. The STF format contains a representation of the style, content, and structure of the original FrameMaker file.

Notes:

The next step following the application of a filter is Leverage Analysis. The files are loaded into the Translation Memory tool and analyzed against the appropriate TM to establish the scope of translation required. The report generated by the analysis shows the leverage available from the Translation Memory and the wordcounts for each of the files analysed. The report sub-divides the leverage available into repetitions, 100% matches, high fuzzy matches, low fuzzy matches, and no matches.

A project TM can also be extracted which will contain all of the relevant segments for the project. This subset of the main Translation Memory is then included with the translation kit which will be sent to the translators.

Preparing Help

Because there are a wide range of Help technologies around, the preparation of Help for Translation is a reasonably involved one. However, there are a number of basic steps that are common to all Help projects:

- 1 Open and generate the Help.
- 2 Examine source files to ensure that they build correctly and that there are no missing topics, graphics, or other files.
- 3 Analyze the Help files to identify which files require translation.
- 4 Isolate translatable files
- 5 Prepare translatable files. If necessary make changes to Tag Editor ini file for JavaScript or other scripting. If necessary prepare translation instructions.
- 6 Analyze files against memory.
- 7 Translate using HTML translation tool.

Graphics

On most occasions the graphics for documentation projects will be created using one of the popular graphic design programs such as Adobe Photoshop or Corel Draw. These programs create layered graphics which allow the designer to isolate the text from the background of the graphic. These layered graphics are then flattened and exported an appropriate format.

Notes:

However, when localisable text is contained within graphics a problem arises. Because these formats are not layered it is difficult to isolate and translate the text. The solution is to locate the layered graphic, translate the text, and then export the graphic again.

For this reason, it is important that you get the source, layered graphics as part of the client hand-off. If the layered files are not available, translation is still possible, but it will be considerably more difficult and as a result expensive.

Notes:

Help Testing and Engineering

Once you receive translated files back from the translator there is still a considerable amount of work that needs to be done before the files can be delivered to the client.

When files are sent back from translation, the following process occurs:

- 1 The files are cleaned to remove the Source language text from them. This will also update the Translation Memory.
- 2 The files are then copied into a folder containing all of the untranslatable files from the Help project.
- 3 The Help system is built. The build process may be as simple as regenerating the Help system in RoboHelp.
- 4 Finally, the new translated Help build is then tested against a full, running, copy of the English Help system.

Help Testing

The golden rule of Help testing is test what the user will see. Never test the pre-build Help system.

There are an ever-increasing number of Help testing utilities on the market. However, the most effective that I have found is a tool called HTMLQA. This is a tool published by SDL.

HTMLQA allows you to visually compare the Source and Target Help systems. The basic idea is to make the target language Help system look exactly the same as the Source files. In addition, HTMLQA also has a number of verification features that allow you to create reports of differences between the files. The type of issues that frequently occur are:

- The number of target files is greater/less than the number of source files
- Broken/missing links
- Missing graphics
- Broken tags

With advances in translation technology, the number of these errors has been diminishing. However, they are still not gone forever.

As a good starting point, you should always run a report first and fix any issues that might appear in the report. Once this is done, the visual inspection should take a lot less time.

Notes:

Help Engineering

Once you have identified the files that need fixing. It is then a simple matter of going into the HTML and fixing them. Obviously to do this you first have to be able to write in HTML. However, by carefully comparing the source file to the target file, it is often possible to identify where the file has been broken and fix it. However, the more comfortable you are with the coding of the files the easier it is to fix them.

Notes:

DTP and Localisation

DTP isn't really my area but I thought I should at least make a few comments about an area that is fundamental to documentation localisation. I had a quick word with a DTP specialist before writing. She mentioned to me that one of the main areas where issues occur for DTP occurs before they have even begun to work on the files. The area where the greatest problems occur is in the hand-off from the client. These are a few of her recommendations for hand-offs:

- Make sure you receive the source file for the project. This may seem simple but surprisingly a large number of hand-offs either do not contain the source file or else they contain multiple versions of the same file, leaving DTP to work out which one is the most up-to-date version.
- Provide a PDF copy of the file. This is included so that any font issues and missing graphics can be easily identified.
- Fonts should be embedded in the PDF. This is so that any missing fonts will be visible allowing the DTP specialist to identify the font for the source document.
- Include information on the application to be used. This includes information on the application version and also the platform.

In addition to these points, she also had some additional pointers for those creating the files:

- Use standard widely used applications to create documentation
- Use standard fonts. If you have a corporate font that must be used, make sure you inform your vendor
- Save call-outs and Table text as text rather than as graphical elements.
- Use styles rather than local formatting
- Leave sufficient white space to accommodate text expansion
- Ideally DTP software should isolate all style information in the style sheet and should also automate cross-references, Indexing, and headers and footers.
- When localising into double-byte character sets, allow for font issues and vertical text expansion
- Avoid including text in icons or graphics
- Assume that the text in callouts and graphics will expand and allow room for expansion
- Link graphics rather than embed them
- Note the screen configuration and resolution for screenshooting. Ideally create a style sheet for screenshooting
- Avoid culturally specific icons and graphics
- Avoid using representations of people and animals in icons and graphics

Notes:

Single-Sourcing and Tool Selection

Some will argue that I am in a very grey area when I introduce single-sourcing as a localisation issue. It has been argued to me on more than one occasion that this is an issue for Technical Writers not those involved in localisation. However, I feel that any documentation models that lend themselves to relatively straightforward localisation are worthy of consideration as localisation models.

Properly and appropriately implemented a single-source strategy can:

- Improve consistency
- Reduce time to market
- Reduce overall localisation costs

However, in order for single-sourcing to aid in the reduction of localisation costs, localisation must be part of the tool selection process. Decisions made about which tools to use, and crucially how to use them, have implications not only for the source language documentation cycle but also for the localisation cycle as well.

What is Single-Sourcing

The first question is what is single-sourcing? Quite simply it is the process of producing multiple output formats from a single source. Typically this involves the creation of Help or Documentation in the standard way and then converting them using a conversion utility into other formats including:

- Printed Documentation
- Online Help (in various formats)
- XML
- HTML
- eBooks

There are a large number of utilities on the market to perform this task. The tool you choose largely depends on what initial format you wish to create. For example, for those who wish to author Online Help and use a utility to create Printed Documentation from the Help content, RoboHelp is probably the best tool on the market. If you prefer to create printed documentation first and then convert that to a web-ready format and Online Help, then Webworks Publisher is (in my opinion) the best tool on the market. However, even though both of these tools will allow you to effectively single-source in one language, the story becomes more complicated when you introduce a multilingual element to the equation.

In this next section, I will discuss the single-sourcing models used by both RoboHelp and Webworks and by comparing both I hope to illustrate both the advantages of single-sourcing and the potential pitfalls when localisation is introduced into the mix.

Notes:

Advantages of Single-Sourcing

The main advantages single-sourced documentation in a localisation context is the reduction in word count. Because the majority of text is reused, the total new words translated is dramatically reduced.

It also allows you to significantly reduce the time spent on engineering and DTP tasks.

Single-Sourcing Models

Webworks Publisher and RoboHelp are two of the most commonly used single-sourcing tools. RoboHelp is primarily as Help authoring tool. However, it also provides utilities that allow the creation of multiple output formats.

In contrast, Webworks is a tool that was created specifically to convert documentation into various output formats. Both of these tools have their merits. However, for the duration of this course we will examine how the tools can be integrated into a workflow that also involves localisation.

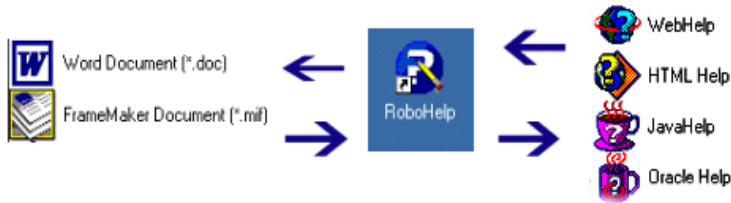
RoboHelp

As mentioned earlier, RoboHelp is primarily a Help-authoring tool. However, over the last couple of releases it has dramatically improved its import and export functions. It now allows both Word and FrameMaker to be imported and converted to Online Help. In addition, it also allows Online Help in all formats to be converted to documentation in the form of MS Word documents. The RoboHelp single-sourcing model is shown in Figure 1 on page 14. This diagram shows RoboHelp's capacity to import both printed documentation and Online Help formats and to output both Help and Printed Documentation.

With RoboHelp you are given the option of writing the Help first and then outputting printed documentation, or as an alternative, you can create your printed documentation in either FrameMaker or MS Word and then import these documents into RoboHelp.

A point worthy of note however, is that the import and export functions in RoboHelp are one-way only. This means that if you, for example, create your initial documentation in FrameMaker and then import it into RoboHelp, you are restricted to using that version of the Help for all future updates. If you update the printed documentation, you will have to update the Help separately (once the initial import into RoboHelp has taken place).

Notes:

Figure 1: RoboHelp Single-sourcing model**Notes:**

Webworks Publisher

Webworks publisher is a single-sourcing utility. It is not an authoring environment. Webworks creates a template containing all of the style elements of the Printed documentation and your chosen Online Help or other output format. It chunks information into Help topics on the basis of the style information contained in the Printed documentation. For example, you may choose to create a new Help topic each time the Heading1 style appears in your document. Once the template has been created any documents of similar format can be used to output multiple formats. The key to the successful use of Webworks is consistent use of formatting in the source documents. If you use a consistent template for your documentation across multiple projects, then you can generate any output you wish using one standard Webworks Template.

Figure 2: Webworks Publisher Model

Localisation Workflow

Hopefully now you have a basic idea of how both tools work. So what are the implications of the model they use to the localisation workflow. To discuss this question, the most important area that must be discussed is the ability of both tools to reuse previous work.

Here is a typical example of a Technical Writing job. Create an Online Help system and Printed Documentation using MS Word and any Help creation tool of your choice. This Help system will be localised into four languages.

Scenario 1

Creating the English files

You author the printed documentation in MS Word and then import that word document into RoboHelp and make any required changes in formatting and content necessary to create an Online Help system.

Localisation

You have a choice for the localisation, you can either:

- Ask your vendor to localise the Word document and then import the document into RoboHelp for each language.
- Send them both the Word Document and the English Online Help and ask them to localise both.

The second scenario is the most probably and most reasonable. However, by doing so you are losing some of the advantage of single-sourcing. You will still get excellent leverage and maintain consistency between the documents. However, the difference in formatting will mean that you will not get the same leverage as you would have had all of your content been in the same format.

Scenario 2

Creating the English files

You use MS Word to create the printed documentation. You create a Webworks template to map the Word styles to Online Help styles and convert the Word Document to Online Help.

Localisation

You send your vendor the MS Word document and the Webworks template. They localise the Word document and use the Webworks template to create the Online Help.

The Point

In addition to the logic of the scenarios I have outlined above, there are also other criteria that a Technical Writer will use to decide on a single-sourcing strategy. There are also localisation considerations that need to be considered such as support for languages and scripts. However, the point I am trying to make here is that if localisation is properly integrated into the documentation cycle it should influence all elements of the Technical Publications decisions, including decisions on the appropriateness of tools.

Notes:

I have worked in documentation for a number of years. I have also worked in localisation. I have yet to find a company with a truly integrated approach to the creation and localisation of documentation.

On the face of it single-sourcing presents ways of dramatically reducing the work involved in creating AND localising documentation. However, alot of thought needs to be given to how single-sourcing is implemented.

Notes: