Software Localization

Lindsay Smith, Country Manager, The York Group (UK) (Lindsay@TheYorkGroup.com)

A common misconception amongst software vendors is that, in order to sell their product in foreign language markets, all that needs to be done is to take the text strings in the product and re-write them. Reality, however, is somewhat different. To avoid a costly failure, consideration must be given to all of the work required to properly prepare a product for operation in another language. This process is usually referred to as 'localization'.

Localization is more than just changing the language of the words in the product's vocabulary - otherwise there would be a temptation to call it 'translation'. True localization takes into account a number of technical and cultural factors as well as the target language itself. The resulting localized product looks and feels right to native inhabitants of your target market country.

Localization is the sum total of all the changes of language, usage, program code, and technology that need to be made to your product in order to offer a product of equivalent industry standard in a foreign language market. Indeed, many localization issues can also occur between two markets with a common spoken language. For example, there are many differences of spelling and usage between the US and the UK.

Localization is actually a very specialist subject and you will be faced with the task of either building this specialty within your organisation, or using external sources of help.
Below we have picked just four facets of localization to illustrate the level of fine detail that needs to be considered when producing a high quality product.

Translation

This is clearly the fundamental constituent of a localization project and there are four main factors that need addressing during a translation. First is the translation of the text of the many components of the product from its original language to the target language.

Second is the character set. Look at the keyboard of your own PC. Here, in the UK, we use a 'qwerty' keyboard specifically designed for entering English text. A German keyboard is different. A French one is different again. They have been optimized for slight differences in the frequency of characters used, as well as for the use of accented characters.

Third are character codes. Behind the keyboard layout is the question of how the computer recognizes the characters that are being entered. There is no (not in common use anyway) single set of computer codes that can handle all known languages. Luckily in Western Europe there is such a common code set, but the keyboards differ even so.

Fourth is the length of the text strings. Text length provides a real challenge to the user interface designer. Screen layouts shrink and grow dependent on language. For example, German needs nearly 50% more space than English to convey the same meaning. A title or message that only just fitted in the available space in English could precipitate a screen re-design in German, if you believe, as we do, that interface design is very important, then simply stretching or squashing to accommodate foreign words is going to look (and be) second rate.

Related to text length is word length. Particularly notorious in this respect are the Germanic languages with their frequent use of compound words. Simple algorithms for finding words in blocks of text, that we take for granted in English, do not work in German. Screen layouts that rely on lists of ('normal' length) words are frustrated by Germanic spelling.

Counting and Time Keeping

There is a wide variation, even over a limited geographic region such as Western Europe, in the way that numeric information is displayed. There is a similar diversity in culture that seeps through into computer software 'look and feel'.

Arithmetic is handled in the same way in all Western languages. What differs is the way the numbers are represented. The commonest example is the US practice of writing mm/dd/yy for a numeric date contrasted with the UK usage of dd/mm/yy, which can result in ambiguity with dates such as 5/6/96. Equally, the French practice of transposing decimal points and commas in numbers (e.g. '1.000,1' for 1,000.1) needs to be catered for.

Currency notation is also quite different from region to region. This can range from prefixed special characters (e.g. £1 or ¥200) to prefixed alphabetic strings (e.g. FF1,0 and DM 3.0) to suffixed alphabetic strings.

Holidays also have significant regional variations (e.g. don't assume December 25th is Christmas Day). If your application has a diary facility then these factors need to be taken into account.

The best quality products show the consumer that the product has been designed with this in mind.

Application Specific Variations

This is a 'catch all' category for sweeping up those things in your own product that may not function correctly in another region or culture. Any part of a product which is driven by, or assists in meeting, laws or regulations in the home market will very likely require modification to fit different rules in the target market (e.g. finance).

Any application where alphabetic ordering was important may produce different results in different languages - and would be virtually impossible to operate in some Asian languages.
An application that operates across national boundaries can also cause problems. Simple electronic mail is an example of this in that many systems do not cater for different character sets and languages, never mind different document formats.

The Whole Product

There are constituent elements of the product that, whilst not strictly software, may need to be localized. A simple example is the Software License, normally displayed on the user's screen at the time the product is started up. You need specialist legal help to get this translated, since the minutiae of the rules vary from country to country. Copyright notices are similar in this respect.

We would include in this section any tools you use to manufacture and support the product. If a regional support office is to be set up, then will the help desk tool you currently use also work in the local language?

Other examples are the product packaging, marketing collateral, training materials and any internal technical support documentation. Also of course, are the language variants of your World Wide Web site.

Performing a localization

The first step, assuming this product has never been translated before, is to survey the product for the kind of localization issues we described earlier. It is essential that this survey is accurately done. Only then can cost and the project time scales be calculated. Indeed it may reveal hitherto unknown issues that completely change the commercial viability of the venture. This is a skilled task and it is best performed with the help of someone who has had previous experience of localization.

Once the survey is complete then the project can be planned. From here onwards there are several methods of translation:

1. Do the complete project in house using your own permanent staff
2. Outsource the complete task to a translation specialist
3. Use your local re-seller to do the translation
4. Manage the project yourself and hire specialist sub-contractors

The first of these is inappropriate for a small company because the cost of full time employment of the specialist (linguist) staff would be prohibitive.

The second method can also be very expensive unless the project is a very simple and straightforward translation. If a high level of change is required to the product's code, then outsourcing this change would be both expensive and high risk.

When a product has already been localized once or twice, it may be possible to ship a translation 'toolkit' with the product which would enable distributors and re-sellers to do a translation. This may not only save translation costs, but it can also form a component part of the overall commercial arrangement you have with those particular re-sellers.

For most software vendors embarking for the first time on product localization we would recommend the fourth method - namely that you keep the project under your direct control, and use specialist sub-contract effort to supplement your skill set. For most Western European languages there is a pool of expertise here in the UK that can be hired on sub-contract terms for a specific project. The actual boundary between what your company does and what the sub-contractors do is highly dependent on the degree of code change to your product that is necessary and the level of resources that you are able to deploy onto the project.

The typical project commences with a translation of the vocabulary of the product, usually called the 'glossary'. This is a list of all the specialist words that are visible in the software and documentation. For Microsoft Windows based products much of this material has already been made available by Microsoft and can be accessed freely. Translation of the glossary is the key quality step in the project. It is absolutely essential that contemporary translations having exactly the correct shade of meaning in the target language are found for the English terms in your product. This can only be achieved by someone who is not only a native target language speaker but is also computer literate and familiar with the translation of technical software. This step is so important that the glossary translation should be independently checked.

The software changes needed for the localization should now be done, and the product tested for correct operation. Now the software translation proper can begin. For most average sized products one or two translators should finish the job in a few weeks. On completion of the software translation the product should again be tested for function, since text changes often reveal bugs that have been dormant in the product. Following that the language elements of the translation should be independently checked.

Once the software translation is complete and checked, the help text and then the documentation can be translated. It is best to keep to this sequence because, until the translated operation of the product has been established, then not only is it difficult to translate the operational parts of the documentation but the diagrams and screen shots in the documentation cannot be included.

We would recommend that translated software is always beta tested. By its very nature, it is impossible for your (source territory) staff to have a view of a translation's target language quality. You are always reliant on the judgement of third parties. Therefore this extra step of checking, by a customer, gives you the confidence that the product is ready for market. It is not always necessary to wait until the documentation is available to start this test. It will depend very much on which customers, or potential customers, are available for beta testing at the time, and what their documentation needs are.

In conclusion we reiterate the critical success factors in any translation project,

- We cannot over emphasize the value of good preparation. The survey stage is essential to scope and plan the rest of the project. Time spent on this activity at the start of a project can save considerable cost in wasted translator effort and expensive re-work later on.

- Translation itself is a skilled task. Speaking the target language is not enough. The translator must have high levels of computer literacy and be familiar not only with the application being translated, but also with the vocabulary being used.

- Localization will inevitably bring to the surface bugs that have been dormant in the product. Be prepared to test the product well, and to budget for sufficient technical support and bug fixing activity.

- Always check translations at least once using an independent translator. Ideally, get a customer or a local office involved for a final approval of the linguistic style and content.