

Localization. General Overview .....	2
What is localization? .....	2
What type of localization is most suited to your needs? .....	2
Business .....	3
Delegation of localization.....	3
The coupled approach.....	3
The integrated approach .....	3
Technology .....	4
Localizing Windows Executables .....	4
Multilingual Architecture .....	4

## Localization. General Overview

### What is localization?

Translation renders the words of one language into the words of another. Localization ensures that the translation will be understood by the target audience. For example, talking to a Russian about “carrying coals to Newcastle” would generate a blank response. However, when translated *and* localized, the Russian audience would hear “to go to Tula with one’s own samovar”. Translation cannot work alone without localization, as a linguistically correct translation may often lead to confusion and misunderstanding.

For software this means that a user in Japan, for example, should not only see the program interface in his own language, but he is also to be able to address issues for which the software is designed in no less effective a way than the user in the original, North American locale.

It is also important to understand what *internationalization* means and what the functions of an international product are. An international product is one that is generic and free from any cultural bias. A localized product is one that incorporates the unique elements required for a particular region.

#### **More people are reached using an international approach**

Specific national and cultural references are best avoided as their use may even have an adverse effect. Idioms, metaphors and stereotypes may be fine in one country, but they may cause offence in another. The same applies to the use of visual effects and gestures.

By using inherently native turns of phrase, rhyming, humor and so on, you are simply complicating the work of the translator and this can lead to avoidable misunderstandings.

In written text, too, presentation of numbers and currency vary from country to country, and telephone numbers should always be written out in full, including local and international calling codes.

Localization covers the following issues:

- Translation of user interface, online help and documentation.
- Adjusting the program to become locale-sensitive and to behave according to local expectations (e.g., sorting; formatting of numbers, dates, time and currencies; bidirectional awareness, character sets and so on).
- Address formats (ZIP code formats, telephone number digits, etc).
- Hot key assignments (Ctrl-O opens a file in an English application, Ctrl-A in a Spanish one).
- Adding or removing certain functionality in accordance with local market needs (privacy regulations in Europe, VAT calculations, accounting rules, etc.).
- Ensuring that the language, layout, iconography and color schemes used in the application are appropriate to the local culture.
- Building, testing and debugging the localized version.

Internationalization is the process of making an application localization-ready.

### **What type of localization is most suited to your needs?**

The approach to localization involves two aspects: business and technology.

## **Business**

There is a number of possible approaches available, and the best suited solution will depend on numerous factors, such as the degree of importance of the localization, the affordability and the time-to-market.

### **Delegation of localization**

Delegating or outsourcing localization, known by some developers as an “over-the-wall” approach, means that a company delegates the responsibility for the localization, including its costs, to resellers. Of course, this obviates the need for the company to become involved in the localization work, thus saving on costs and it is likely that the product will have a truly native “feel” on the local market.

However, there are considerable disadvantages. As the VARs are focused on selling, they will spend longer on product development and the time to market will suffer. There will also be a lack of direct control or coordination of upgrades and new releases on the part of the manufacturer. This control may also stretch to the proprietary aspect of the end product: if a VAR produces the localized version, it is expected that they will also own the rights to it.

There are many smaller companies that nevertheless opt for this solution, as it offers a relatively inexpensive means of entering international markets. This approach is chosen when there is a high uncertainty regarding sales of the localized version or projected ROI from your own localization efforts is not high enough to justify it.

### **The coupled approach**

As companies gain a better understanding of future sales and localization ROI for key geographies, and risks go down, manufacturers prefer to have a greater degree of control over their localized products and, thus, they take on specialist staff or contract work to a localization company. This results in a certain coordination of the efforts of the product development and the product localization operations.

There are numerous advantages of this strategy, as opposed to simple delegation. The manufacturer gains control over the localized product, meaning it can apply its corporate image with confidence and uphold corporate quality standards without detriment to lead times to bring products and updates to market.

As proprietor, the manufacturer decides who its reselling and distributing partners are on the international markets.

Of course, bringing localization in-house has a cost implication and all localized products will require beta and pre-release testing and this is only possible once the English version is finalized.

### **The integrated approach**

With a truly global market, it is now a requirement for certain products to be released in more than just the English language. There is a distinct marketing benefit from the simultaneous release of software products in localized formats and the major manufacturers are now developing software together with multilingual graphics interfaces, necessitating the application of an integrated approach to localization.

In this way, beta or pre-release software can be released simultaneously in multiple languages. This means that beta testing can be performed worldwide, enabling any necessary adjustments to be made across the world in a short space of time.

A totally integrated approach means that uninterrupted access is required to GUI files and, therefore, localization specialists working in an integrated environment have to operate in one location, with frequent access to RCS or the software developer's file vault.

Despite the advantages, the integrated approach is not cheap and, for all but the largest companies it could be an expense that can simply not be justified.

## Technology

At present, the majority of software applications are created in a single language interface and this means that, in the face of growing globalization, they will be at a distinct disadvantage in years to come. Two principal solutions are to localize Windows executables and introduce multilingual architecture:

### Localizing Windows Executables

This is the only thing you can do fast if you haven't developed multilingual architecture (see a detailed description below). There are special localization tools that are able to extract resources from Windows binary files into a translator-friendly format and to compile executable target binaries with the translated text. Such tools have a number of benefits:

- They are easy to use.
- They maintain the context of the translatable items, allowing the translator to choose the precise expression and tone for every translatable item.
- They can be integrated with translation memories, to maintain translation consistency.
- They deliver running target executables

And such a localization process as a whole is a lot faster and cheaper than the first internationalization application (that is the development or refactoring of multilingual architecture) even though no tools are perfect and there is always a large volume of qualified (equals expensive) engineering work.

However, it is important to understand that it solves only the task of interface translation; all other tasks mentioned above remain untouched.

Disadvantages of localizing executables:

- This process is expensive, primarily owing to the volume of specialist manual work involved.
- Each software release entails a volume of manual work for the extraction and merging processes and this takes time. This presents a problem for making frequent development updates.

### Multilingual Architecture

Multilingual architecture offers an alternative approach, where the software application architecture is modified to separate program logic from language-specific aspects. The language-specific aspects are kept separately, for example, on a separate database. The advantages of this approach involve the following:

- Multilingual architecture enables users to work on server applications with different languages and even change languages during operating time.
- Costs and time expenditure can be lower as the translation is only required for resource files. The software is usually designed so that it can be converted into Microsoft Word, the favored processing application for translators, as it supports translation memories and other computer-assisted translation tools.

This approach is not without disadvantages, particularly as regards development work, where a considerable amount of time is needed to introduce multilingual architecture and also the developers need to know how a translator operates, to avoid excessive manual intervention.

A quick summary of advantages and disadvantages of both approaches:

**Localization of Executables    Multilingual  
Architecture**

Time to Market	fast	slow
Initial Investment	low	high
Repetitive Costs	high	low
Development Cycle Time	slow	fast