

What is „Software Localization“

Software localization is the process of *translating a software system into different local languages*. Localization includes also adaptation of the software to cultural preference, such as the type of the decimal separator (dot for US, comma for Europe), the text direction (left-to-right or right-to-left), and the format of the ZIP code or the dialog layout.

Localization dramatically improves the acceptance of software in a given country. There are countries, where non-localized software simply won't sell (i.e. France or Italy).

Who is involved in software localization

- The **development company** creating the software system. The developing company typically ships the software in only one language - the source language.
- The **target market representative**: This is typically a sales office or a distributor in a local region. The region consists of one or more countries in which one or more local languages are spoken. Very often, the target market representatives are responsible for the organization of software localization.
- **The translator**: A person or company who actually executes the translation of words and phrases from the source language to the local target language(s).

Typical problems in software localization

Today localizing software is costly, time consuming and error-prone.

- The **“Skill set dilemma”**: Software localization today requires both, programming and translating skills: In order to achieve satisfying translation results, a person who is a native speaker and fluently speaks two languages is required. In order to get translated words and phrases “back into the original software”, this person must have at least basic programming knowledge. *Typically translators do not possess these programming skills.*
- The **“Core Know-How dilemma”**: In order to enable target market representatives to translate software, the development company sometimes ships source code of its application. This is certainly not desirable, because it may lead to introduction of errors which are specific to a localized version. *This results in unnecessary test cycles which mean higher development costs.* Additionally, the development company distributes its major asset, which is the source code, outside its controlled organization. *This may easily result in unauthorized copies or unauthorized use of intellectual property.*
- The **“Context dilemma”**: To work around the “Skill set dilemma”, software producers have invented some alternatives for software localization, which are all based on the same principle: Software texts are exported into an “easy to use” format, such as Excel, Word or a database. Translators use these files for translation. The major problem is that the context information gets lost. The translator does not see the environment, i.e. the dialog or the menu, in which a text occurs. Therefore, he may be unable to use the correct translation for a given word (**Example**: The English word “number” may be translated to the German word “Nummer” in one context, but to





visuallocalize



the German word “Anzahl” in another context). ***This leads to unnecessary quality assurance costs and warranty issues.***

- **The “Non-Standard-Programming dilemma”:** In order to display translated texts from Excel files or databases in a software application, developers invent special mechanisms for text loading. This is a non-standard behaviour which is not supported by any development tool. The developer cannot use the standard Visual Studio Dialog Editors to create his applications, but has to add specific string loading code to his application. Exporting and loading strings to file formats which are not intrinsically supported by the development tools are additional sources for malfunction. ***This leads to unnecessary development costs.***
- **The “Versioning and update dilemma”:** Development cycles become shorter and shorter. Whenever a software version is localized, a newer version may already be under construction and the translation process starts from scratch. Today, it is almost impossible to reuse translations from a previous software version. Correctly reusing a translated phrase means to apply the translation to the same context in which it has been originally used. ***This leads to unnecessary recurring translation costs.***
- **The “Quality dilemma”:** Software localization, especially when based on non-standard mechanisms, may break an applications’ functionality, ***leading to unsatisfied customers and loss of market share.***

What is Visual Localize NET



Visual Localize is an application that fully supports the software localization process of Microsoft Windows applications (including .NET applications), databases and websites based on XML and HTML files. ***It dramatically reduces cost, effort and complexity of software localization.***

How Visual Localize helps the Software Development Company

- **Faster time to market for the non-localized software:** Because *Visual Localize* allows programmers to concentrate on their development tasks, without having to care about localization, the development effort is reduced. The results are ***shorter development cycles and therefore lower development costs.***
- **Higher Quality at lower Costs:** *Visual Localize* does not impose any specific programming practices on the developer. Therefore the risk of introducing errors through code that handles the loading of language specific texts is completely eliminated.
- **Protection of Intellectual Property:** No source code is needed for the localization process when using *Visual Localize*. Therefore, the ***intellectual property of the software development company is protected*** against unauthorized use.



How Visual Localize helps the Developer

Visual Localize extracts all language dependant information directly from the compiled EXE / DLL files. An application developer can focus on the application development, without having to care about localization at all. The developer uses the Standard Visual Studio Development environment and develops his application as if it would be a single language application. **He is much more efficient and productive.**

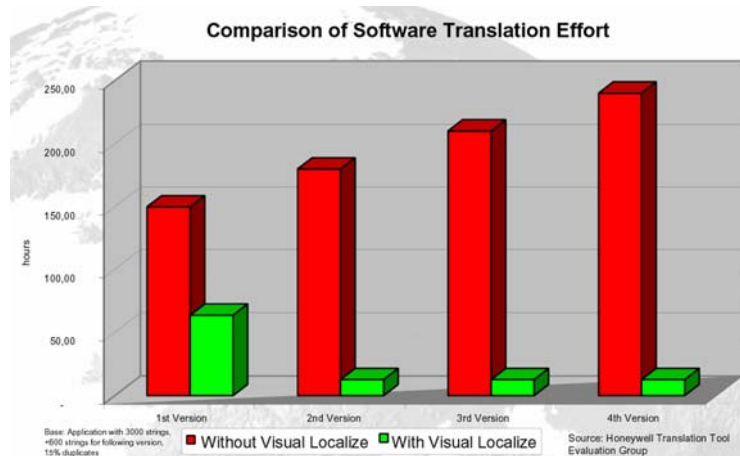
How Visual Localize helps the Translator

- **Minimized effort through automation:** *Visual Localize* remembers all previous translations and thus maximizes reuse. Additionally, it allows the import of translations from other applications or 3rd party dictionaries.
- **New Business Opportunities:** *Visual Localize* is a very easy to use tool. With *Visual Localize*, no programming skills are required in order to provide software localization services. Therefore, *Visual Localize* creates new business opportunities to translators who previously did not address software localization.
- **Visual Editing for better quality:** *Visual Localize* not only extracts the translatable texts from an EXE / DLL file, but also extracts the full context information. Texts are shown in a dialog/menu preview that allows a translator to exactly determine the meaning of a specific text phrase. Translation is done in a WYSIWYG fashion. A translation from English to French typically increases the string length by 20-40%. Therefore, the WYSIWYG dialog preview allows the language engineer to adapt size and location of control elements to the new length of the translated texts, producing superior quality of localized software. In addition, *Visual Localize* prevents the translator from introducing errors during the localization process.

How Visual Localize helps the Target Market Representative



- **Tremendous cost reduction:** There are several facets of cost reduction that *Visual Localize* offers:
 1. **Translation can be done by any translator.** With *Visual Localize*, no programming skills are required. As the required expertise is reduced, **the cost for localization is also reduced.**
 2. **Easier handling results in reduced administration costs:** With *Visual Localize*, all the translator needs to have is a single *Visual Localize* project file containing all information. Forget the error prone and costly handling and administration of hundreds of files that belong to a software system.
 3. **Automatic versioning and update:** Whenever a new software version is released, it is a matter of seconds to correctly apply all previous translations to the new version. The translator will only have to translate the new strings of the current version, instead of starting the localization from scratch. **Studies have shown a cost reduction of up to 90%.**



Visual Localize also takes care of consistency of all files that belong to a specific version of software.

- **Faster time-to-market for the localized version:** *Visual Localize* offers improved translation efficiency combined with reuse of previous translations and use of 3rd party dictionaries. ***This reduces the time to market for a localized software version from several weeks to only a few days.***
- **Improved Quality:** As *Visual Localize* works independent from the source code, it prevents the translator from introducing any bug into an application. Furthermore, *Visual Localize improves quality* by
 1. Use of context sensitive dictionaries.
 2. Automatic upgrades from previous versions.
 3. Visual context information in the form of previews.
 4. Automatic error checking.

Statement to Visual Studio .NET

Visual Studio .NET supports translatable resources. However, Visual Studio only puts translatable resources into satellite assemblies if they are different from the main language. This does not match the real-world translation process. In the real world, a translator will get a software version once it has been completed by the development company. The development company neither knows nor cares about the languages into which the software will be translated.

When a translator starts localization, he needs to have all existing resources and texts available. *Visual Localize* supports this real world scenario by creating full satellite assemblies automatically from the application.