

Effective Update Management in the Localization Process

By Tom Shapiro, Member, Boston Chapter

Overview

Whether one is localizing documentation or translating Web sites into multiple foreign languages, managing updates is a major component of the localization process. Content development often involves constant updates. Therefore a localization methodology must have the infrastructure to manage change seamlessly, efficiently, and accurately. It must also offer complete flexibility to accommodate each project's unique schedule, requirements and development cycle.

A file and change management system must ensure version control for highly complex, constantly updated file sets. In a localization context, such a system must allow for the concurrent development of source files and localized materials as well as enable corporate-wide implementation that expedites file sharing among large, globally dispersed teams.

Planning

The first step in managing updates is planning. The more that a company plans collaboratively with its localization services provider, the easier it becomes to develop an update strategy that is optimized for meeting the company's specific objectives. Prior to beginning a project, it is necessary to confirm the following:

- Development cycles
- Update schedules
- Projection of nature and volume of updates
- Customer priorities
- Ship date deadlines and nature of deadlines, and so on

“Leveraging”

In a localization context, “leveraging” is the process of re-using translations, localized graphics and sizing information. Leveraging prior work when processing updates not only ensures a high degree of consistency, but also shortens time-to-market and reduces localization and technical support costs. As a precursor to any localization update management plan, content authors need to have strict guidelines regarding the timing and the types of updates that are permissible. What might be perceived to be “minor” updates to the source could result in needless extra costs and shipping date delays. This is exacerbated when organizations translate materials into a large number of languages.

When applicable, Translation Memory (TM) solutions can expedite the update process. This use of TM includes Internal Repetition (the re-use of repeated text within a single file set), as well as leveraging of translations from prior versions of a product. In other cases where TM may not be a cost-effective solution, an alternative leveraging mechanism should be available wherever possible.

For updates, your update management system needs to automatically identify and calculate the delta (modifications) between new and pre-existing files, regardless of the number of changed

files or the amount of change between successive deliveries. The localizer should have the capability to identify and track all change in the files, thus saving you time and freeing up your resources.

When leveraging prior translations or re-using Internal Repetition, it is critical to establish a quality assurance strategy to ensure that leveraged translations are still acceptable in the new context, that linguistic consistency is maintained, and that code integrity and other technical quality issues are being verified. Quality assurance (QA) is often forgotten in the equation, yet without managing the QA process in line with updates, a huge amount of costly rework can result.

Version Control

Localization is similar to product development in that version control is absolutely critical. However, it is different from product development in certain respects. Whereas core development deals with only one language set, localization can deal with 2, 3 or 30 file sets concurrently. In the case of simultaneous shipment of the source and target language versions, there is the additional requirement of keeping all of the target language versions in sync with constant changes.

Your file and change management system needs to allow a large, globally distributed team to share and update files while maintaining version control and concurrently clarifying file ownership. It needs to manage multilingual file sets, where there may be thousands of files and dozens of file types per language, with constant updates that must be implemented across the different language versions throughout the process. Complicating matters, the system needs to support asynchronous movement among the different language versions. For example, when localizing into 15 or more languages, it is not uncommon to “tier” the languages and implement different schedules corresponding to the different tiers, thus potentially resulting in a very different update management implementation depending on the particular language.

Your system also needs to support teams with a wide variety of participants, such as translators, editors, product managers, technical publications specialists, reviewers, engineers, testers, and others. located in multiple countries. The different team members should be able to find files easily and quickly, regardless of how much change occurs. In addition, the system needs to enable you to revert back to any prior version easily and quickly in any of the languages if necessary.

Conclusion

Effective management of updates is critical to the success of individual localization projects as well as long-term localization strategies. Although the most reliable means to ensure that your scheduling and pricing objectives are met is to plan as much as possible from the start, it is often the case that late, unexpected updates are unavoidable. A solid infrastructure and methodology, coupled with extreme flexibility, are imperative for incorporating updates into localized versions seamlessly, efficiently, and accurately.

Tom Shapiro (tom.shapiro@rubric.com) is VP, Sales & Marketing at Rubric, a localization and translation services provider to technology and life sciences companies. Shapiro's articles about localization, translation and global business have been published in *MultiLingual Computing & Technology*, *Software Business*, *Software Export Guide*, *Mass High Tech*, *Boston.com*, and *The ATA Chronicle*.