milengo

In-Context Software Localization Solution Brief

Context can be crucial when localizing software products. Without information for linguists on what exactly they are translating and where it appears within an application, linguistic errors that can affect the usability of a product or its reputation in the target market are extremely likely. A common illustration of this problem is verb-noun confusion. The word “view” in English will remain the same in both forms; in German, however, it could be Ansicht, Sichtweise or Ausblick (nouns) or anzeigen, ansehen or anschauen (verbs). Furthermore, GUI issues, such as truncated strings or duplicate hotkeys, can also arise when linguists are working out of context.

Ultimately, there is no single solution compatible with all software types and platforms that fulfills this need to convey context information to linguists. This does not correspond to a lack of options, however. With the right approach, sufficient context can be obtained. At Milengo, our focus is to engage with our clients’ product and development teams in order to determine the best solution depending on the application at hand.

Milengo case study

Milengo works with one of the world's leading contact center software solutions provider. For the translation of a number of the company's products, Milengo employs a comprehensive strategy to ensure as much context as possible is available to linguists.

Where DLL is provided for translation, SDL Passolo is used so that linguists can benefit from the tool's visual localization functionality. With this, translators can call up an in-context view in order to see how their translations will appear in the final product. To complement this, an online spreadsheet shared between Milengo staff, linguists and the client's team ensures that all translator queries remain in one central location.

Milengo's production team also have access to the client's software on a virtual machine. This offers the possibility for project managers to address queries related to context alongside the client. For security reasons, access to this virtual machine is granted only to approved staff on Milengo's side.

After translation is completed, the localized application is built and tested by the client's development team. Once a build is available, Milengo goes into the application and takes screenshots according to specific test case scenarios. These screenshots are then sent to our linguists for a final in-context review before the release is completed.

Selecting the right tool for the job

Our first step is to look at the client's preferred source format for translation. Based on this, we can determine whether there is any particular computer-aided translation (CAT) tool that will be most suitable for use from a context perspective.

For example, when localizing applications built on the .NET framework, two particular CAT tools stand out: SDL Passolo and Alchemy Catalyst. Both tools offer a visual localization functionality, so that linguists can see exactly how their translations will appear when used in the localized application. When working with .exe or .dll as a source format, this approach can deliver significant benefits.

Web-based applications can also benefit from the visual localization functionality on offer with cloud-based translation tools, such as Smartling. Again, linguists have the ability to view their translations string by string exactly how they would appear in the live application. Milengo is experienced in integrating these platforms into our processes and can discuss their feasibility depending on the specific requirements of the client.

Applications built for other operating systems do not - primarily due to the prevalence of Windows - currently have a similar visual localization compatibility with commercially available translation tools. While any standard tool will be able to process the platform-specific resource files, other options for context will need to be considered.

For mobile applications, SDL Passolo, Alchemy Catalyst and a selection of cloud-based tools offer in-context localization, particularly for the Android platform; however, in many cases, context needs to be obtained through alternative methods.

How else can context be obtained?

Often, the translation tool being employed will not provide linguists with direct context; for example, if .exe/.dll for Windows or .apk for Android simply cannot be shared for localization. In these cases, Milengo will initiate discussion on the most suitable options and bring in our localization solutions team if required.

There are a number of tried-and-tested approaches to gathering context for linguists that can be employed on all projects. These include:

Query management process

We encourage linguistic queries during any localization project. As you can expect a large increase in the amount of queries should there be little or no context available, it's important to define and implement a strict query management process from the outset.

If no existing preference for handling queries is in place on the client's side, Milengo will initiate this. Setting this process up involves identifying the most suitable client contacts for answering queries as well as agreeing on matters such as the frequency and categorization of queries.

Access to software for linguists

This can be in the form of receiving a demo license for the product, access to the application via a virtual machine or the ability to log in to a staging environment. Such access can help a Milengo project manager or linguist obtain context for core functions or features of your software.

Security is a major concern for our clients in this regard, and we are happy to have any required non-disclosure agreements in place before setting up this access.

Automated screenshooting

Appending strings with screenshots can be a time-consuming task; however, it may be feasible to automate this process in order to reduce the effort required. Links to these screenshots can be incorporated into the translation tool, so that linguists can access them while translating. Milengo is experienced in developing scripts to handle this process for our clients.

String commenting

Milengo's localization solutions architects are on hand to discuss how best to implement string comments. For example, in XLIFF (XML Localization Interchange File Format), developer comments for translators can be placed within the `<note>` tag. This comment can be imported into the CAT tool and shown to linguists when they are translating the corresponding source text contained in the `<trans-unit>` element. For example:

```
<trans-unit id=”1”>
    <source>View</source>
    <target>Anzeigen</target>
    <note>View is a verb here</note>
</trans-unit>
```

Linguistic testing workflows

Ideally, as much context as possible should be available during the translation stage of the localization process. However, in-context testing steps can also be employed to verify the quality and contextual accuracy of UI translations.

Milengo can integrate into an existing linguistic testing framework or offer engineering support to advise on and implement an effective process. In the latter scenario, once our localization solutions teams have clarified requirements, such as test coverage, number of test cases, etc., we can assist in the development of automation scripts for testing a localized, pre-release build of an application.

Such linguistic testing steps can be carried out in a number of different ways; for example, with local end users or Milengo's own native-speaking linguists working with screenshots, a pre-release version or within a dedicated testing environment. Therefore, we encourage discussion between our localization solutions architects and the client's engineering/product teams to identify the best approach.

To learn more about Milengo's solution for in-context software localization, please contact sales@milengo.com