



MADCAP FLARE HELP LOCALIZATION

ezGlobe

HANDLING PLACEABLES AND PROPRIETARY TAGS IN MADCAP FLARE PROJECT

INTRODUCTION:

Madcap Flare is one of the most popular authoring and publishing solutions for web, desktop, mobile and printed documentation. Madcap flare is bundled with Lingo, a Madcap proprietary translation tool; however, majority of companies elect to use other, more streamline computer-aided translation (CAT) tools such as SDL, MemoQ... The reason is simple: It offers more flexibility.

Thorough preparation of Madcap Flare source files for use with any CAT tool is key. EzGlobe's proprietary parsers complement the standard translation industry tools and yield translation-friendly files compatible with CAT technology.

KEYWORDS:

Madcap Flare, Madcap Lingo, HTML, XML, translation file format, parsing, parser, translation memory

MADCAP FLARE PROJECT

Madcap Flare uses two types of files; both contain proprietary tags and customizations:

- **HTML:** HTML-based files contain the body (topics)
- **XML:** XML-based files contain the project information (build information, table of contents, skin customizations, glossary...)

In order to achieve full localization where all text is translated and facilitate creation of the desired outputs, such as WebHelp or PDF, both file types need to pass through the translation workflow¹. This is why you need to submit the entire Madcap Flare project to localization. If you just provide the topic files, the WebHelp frameset will not be localized properly.

PREPARING FLARE FILES FOR TRANSLATION

If you want to translate Flare files using CAT technology (also known as “translation memory”) you cannot use the files “as is”. You need to create files in which we separate the content (text to translate) from the format (formatting information in form of tags). Further, the tags need to be protected to avoid inadvertent alteration.

In general, both HTML and XML are great formats for translation because they are very structured and can be easily parsed. However, Madcap Flare files contain proprietary tags and grammar, which are not always obvious and the standard CAT tools do not recognize them. The result may be incompletely translated files or corrupted Flare projects.

EzGlobe has a proven methodology for Madcap Flare localization. Our proprietary tools augment mainstream parsers and achieve the desired results: **No text is omitted from translation and all non-translatable text is protected.**

Example: HTML Files with Proprietary Tags

Flare Proprietary Tag →

```
<p class="p_new_verdana">The Download Dive Computer Data window allows you to browse for, download, and view data from your Seahorse Fun and Seahorse Advanced dive computer.</p>
  <p class="tip1">NOTE: Your Seahorse dive computer must be set to Read Mode.</p>
  <p class="p_new_verdana">
    <MadCap:keyword term="Download dive computer data" />
  </p>
  <h3 class="H3_New">Seahorse Advanced Dive Computer</h3>
  <p class="p_new_verdana">When attached to a USB port on your computer, a Seahorse Advanced Dive Computer behaves like a storage device (thumb/flash drive).</p>
```

← *Text to translate*

In the above example, a standard parser will consider the entire line as code: `<MadCap:keyword term="Download dive computer data" />`. It will not recognize

¹ Simplistically said, translation workflow is a 4-step process of pre-production (file preparation), step translation, post-production and quality assurance. However, when it comes to localization, the proverbial one-size-fits-all approach does not apply and each consists of variety of tasks that are specific to each individual project.

the string to translate: "Download dive computer data". This string is used to create an index and thus must be translated.

Example: XML Files with Specific Grammar

Proprietary
Embedded
Grammar

```
<?xml version="1.0" encoding="utf-8"?>
:CatapultToc
  Version="1">
  <TocEntry
    Title="[%=MyVariables.ProductName%] Online Help"
    Link="/Content/welcome.htm"></TocEntry>
  <TocEntry
    Title="Main Features of Seahorse Fun Dive Computer"
    Link="/Content/main_features_seahorse_fun.htm"
    StartSection="false"
    StartChapter="false"></TocEntry>
```

Same as in the previous example, the standard parser will consider the entire line as text or as code: `Title="[%=MyVariables.ProductName%] Online Help"`. It will not recognize the string to translate: "Online Help" or it will leave the tags vulnerable to inadvertent deletion or alteration. This string is used to create the table of contents and must be translated. In the absence of translation, you will receive a compilation error.

CONCLUSION

A one-size-fits-all approach that applies a standard parser and sends your files to translation will yield poor results. A fair amount of the content will not be translated and there is a high risk that some of the tags will be broken.

In the best case scenario, this will just mean that your documentation will be a mix of the source and target languages. In many cases, however, this will cause compilation errors that can be extremely difficult and thus costly to resolve.

Consider the cost of a single defect, and remember that **a defect introduced during translation due to an inadequate approach will be multiplied by the number of target languages!**

We welcome you to contact us with any questions. Request a free evaluation and cost estimate of your Madcap Flare localization project.

