Semi-Automatic Localization of Screenshots for User Documentation

Dr Adrian Morse,
Documentation Manager, Picis

Topics

Overview

- The problem, solution and limitations

Infrastructure

- Variable files, Excel workbook, MadCap Capture profile

Screenshot Localization Process (per Project)

- Folder structure; populating variable files; editing images; switching languages

Demonstration

2

Traditional methods for localizing screenshots

- Recapturing screenshots in localized software
- Faking screenshots

Potential problems

- Time consuming for many languages
- Can be difficult or impossible to recreate the screen due to:
    - Special configuration
    - Custom coding (e.g. SQL, HL7) to show user-entered text
    - Post-edits by author
    - Localized software may not be available yet

Solution requirements

- All screen text ("strings") can be exported to Excel

- Each string has an ID, language identifier and text

- The following software is needed:

  - MadCap Capture (v5 or later)
  - Microsoft Excel (any version)
  - PaintShop Pro (any version) or similar
  - A text editor with Find/Replace across multiple files

Solution concepts

- Capture lets you add variables to images
  (in text rectangles).

- Variables link to definitions in a Flare variables file.

- You can create a variables file for each language.

- Before adding variables in Capture you need to prepare
  the image with an image editor:

    - "Wipe" existing text
    - Adjust background to allow longer strings

  Linguistic knowledge is only required for user-entered text.

Limitations

- Preparing an image takes time. (A bit longer than faking a single image.)

- The <u>look</u> of final text is not identical to original text – it actually looks sharper. Readers are unlikely to notice.

- The <u>position</u> of final text is not identical to original text because of tweaking to allow for longer strings. Readers are unlikely to notice.

> If your images need to be 100% identical representations of what is on screen then this method is not for you.

DICIS

Variable files

- Variable files are XML files in Flare projects

- Found in **Project\VariableSets** folder

- In Capture an image can be linked to a Flare Project File.
  (Flare is not needed.)

- Create a Project\VariableSets folder and "dummy" Flare project file
  for each language

- Folder needs two variable files -- *Standard.flvar* and *Custom.flvar*

- XML files must contain lines in orange, plus a variable definition for
  each string:

```xml
<?xml version="1.0" encoding="utf-8"?>
<CatapultVariableSet>
<Variable Name="7" Comment="ENGLISH">OK</Variable>
</CatapultVariableSet>
```

PICIS

Excel workbook

- Excel workbook to store strings and create variable definitions. Let's call this *ScreenshotText.xlm*.

- *ScreenshotText.xlm* needs two worksheets:
  Standard —for strings that are built into the software
  Custom —for "data" strings shown in example screenshots.

- Each worksheet needs these columns:
  ID
  String
  Language
  Variable Definitions

- The Language column should have an autofilter applied to it so that entries can be easily filtered by language.

- All columns should be empty prior to every project.

°•●● PICIS
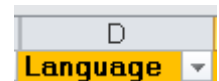
MadCap Capture profile

- A profile saves time. (Instead of opening/saving each file you save all at once.)

- You need to create a profile that re-saves the file in the expected format e.g. PNG
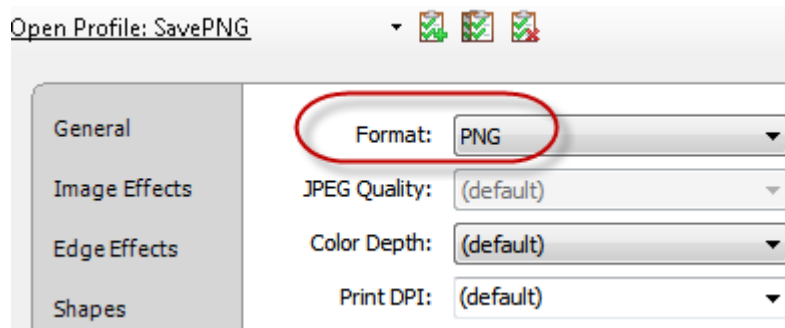
- Only **Format** should be specified. Other values left at default.

Folders for screenshots

Each documentation project needs:

- A folder for "master" images and .props files.

  - Images in this folder are edited in Capture. They are not directly used in final localized document/help file.
  - For each project, take screenshots with English software as usual and put them in this folder.
  - Use any screen capture program.

- An empty folder for each language, where you will place the final localized images. For example:

  📁 _Master Files
  📁 Final English graphics
  📁 Final French graphics
  📁 Final Spanish graphics
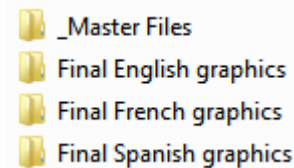
For each new project…

i. Obtain strings from developers (all languages) and add them to the "Standard" worksheet in the *ScreenshotText.xlm* file.

ii. Obtain any "data" strings needed for images and add them to the "Custom" worksheet of the *ScreenshotText.xlm* file.
(Translations for these will need to be done by a translator as normal).

iii. Run a macro in each worksheet of the *ScreenshotText.xlm* file to populate the "Variable Definitions" column.

For each language in the project…

1. Populate the relevant *Standard.flvar* variables file:

 a. Open the "Standard" worksheet of the *ScreenshotText.xlm* file.

 b. Filter the "Language" column by the language you want.

 c. Open the relevant *Standard.flvar* variables file in a text editor.

 d. Copy visible rows from the "Variable Definitions" column of the "Standard" Excel worksheet and paste between the `<CatapultVariableSet>` lines of the *Standard.flvar* file.

2. Repeat steps for the *Custom.flvar* variables file (using the "Custom" worksheet of the *ScreenshotText.xlm* file).

First…

i. Place a copy of the original images in the project folder you made for master images.

ii. Open each image in Capture and link it to the (dummy) English Flare project file.

Then, for each string in an image…

i. Use an image editor to remove the string. Leave an uninterrupted background where the text used to be.

ii. Search for the string in the *ScreenshotText.xlsm* Excel file. Note its ID and the length of the longest localized version.

iii. Use an image editor to adjust the background image to accommodate the longest localized version.

iv. In Capture, add a text rectangle for the replacement text. Inside it add a link to the related variable via its ID number. Use this syntax:

```
[%=Standard.5686%]
```

(In this example the string with ID = 5686 is pulled from the file *Standard.flvar*.)

Switching languages

- Switch language by linking images to another Flare project file.

- Ideally for all images with a Capture profile as a batch job.

- Unfortunately, Capture profiles don't allow linking to Flare projects, but there is a workaround:

    i. Change links in all "props" files using a text editor.

    ii. Open all image files in Capture and save.

- Ideally, **File > Save All**. Unfortunately, as changes were made outside Capture it thinks there are no changes to save.

- Workaround is to use a profile that does something innocuous as part of a batch job.
  > In our case, we apply a profile that saves files in PNG format.

DEMO

UA Europe Conference, June 2013