Using localised images in Java ME

Overview

This code snippet demonstrates how to implement the localisation of images in a Java ME application.

One way to implement this is to have variations of an image for each locale in the application, and have some resource to determine which image to use with the specified locale. These settings may be in the manifest and/or the JAD file of the application.

To do this, add the following lines in the manifest of the application:

1. img-en: Hello-en.png
2. img-fr: Hello-fr.png
3. title-en: English
4. title-fr: French

The first two lines contain the file names of localised images for English (en) and French (fr). During runtime, the application gets these lines with the `MIDlet.getAppProperty()` method, and loads the appropriate image. By getting the current locale using the `System.getProperty()` method, and then loading the required image, localisation of images can be achieved.

Source file: manifest.mf

```
MIDlet-1: UsingLocalizedImagesMidlet, , UsingLocalizedImagesMidlet
MIDlet-Vendor: Vendor
title-en: English
img-en: Hello-en.png
MIDlet-Name: UsingLocalizedImages
title-fr: French
img-fr: Hello-fr.png
MIDlet-Version: 1.0
MicroEdition-Configuration: CLDC-1.1
MicroEdition-Profile: MIDP-2.0
```

Source file: UsingLocalizedImagesMidlet.java

```
    /**
     * Outputs localized images to the specified form
     * @param form - form where fields with localized timestamps will be placed
     * @throws java.lang.Exception if error occurs while localizing images.
     */
    private void showLocalizedImages(Form form) throws Exception {
        // Get current locale
        String currentLocale = System.getProperty("microedition.locale");
        TextField locField = new TextField("CurrentLocale", currentLocale,
                128, TextField.UNEDITABLE);
        form.append(locField);

        // Outputs localized images to the form
        for(int n = 0; n < locales.length; n++) {
            addLocalizedImage(form, locales[n]);
        }
    }

    /**
     * Adds localized image to form.
     * @param form - the form for placing image item with localized image
     * @param locale - locale for localizing image
     */
    private void addLocalizedImage(Form form, String locale)
            throws Exception {
        String title = getLocaleTitle(locale);
        String imgName = getLocaleImageName(locale);
        if(title == null || imgName == null) {
            throw new Exception("Unable to get applicaton settings.");
        }

        StringBuffer buffer = new StringBuffer();
        buffer.append(title);
        buffer.append(" (");
        buffer.append(imgName);
        buffer.append(")");

        title = buffer.toString();

        InputStream ios = this.getClass().getResourceAsStream("/" + imgName);

        Image image = Image.createImage(ios);

        ImageItem imgItem = new ImageItem(title, image, 0, null);
        form.append(imgItem);
    }

    /**
     * Gets locale title from application settings
     * @param locale - the id determining returning title for locale
     * @return title for specified locale
```


(C) Copyright Nokia 2013. All rights reserved.

```
     */
    private String getLocaleTitle(String locale) {
        return getAppProperty("title-" + locale);
    }

    /**
     * Gets locale date format from application settings
     * @param locale - the id determining returning name of localized image
     * @return localized image for specified locale
     */
    private String getLocaleImageName(String locale) {
        return getAppProperty("img-" + locale);
    }
```

Postconditions

When running the snippet, image fields with variations of an image containing the word 'Hello' in French and English are displayed.

Supplementary material

This code snippet is part of the stub concept, which means that it has been patched on top of a template application in order to be more useful for developers. The version of the Java ME stub application used as a template in this snippet is v1.1.

- The patched, executable application that can be used to test the features described in this snippet is available for download at Media:UsingLocalizedImages.zip.
- You can view all the changes that are required to implement the above-mentioned features. The changes are provided in unified diff and colour-coded diff (HTML) formats in Media:UsingLocalizedImages.diff.zip.
- For general information on applying the patch, see Using Diffs.
- For unpatched stub applications, see Example app stubs with logging framework.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.