Localising app attributes in Java ME

This code snippet is one of the series of snippets that demonstrate how to implement multi-language applications in Java ME.

Overview

There are several techniques for localizing a MIDlet:

- Using application attributes
- Using text files
- Using resource bundles

This snippet describes the first technique; using application attributes.

Generally speaking, application attributes (or MIDlet attributes, or system properties) are key-value pairs that represent information about the MIDP application. Attributes can be specified in the application descriptor file (`.jad`), in the manifest file (`manifest.mf`), or both. There are specific rules about how the attributes are located and how possible overlappings are resolved, but they are outside of the scope of this code snippet. See the "See also" section for more information.

In the technique presented in this snippet, the attributes related to localization are specified in the JAD file. This allows adding languages easily since no coding is required. On top of that, it is possible to create separate JAD files for different locales, in which case the application provider can allow the user to download only the file the user is interested in. On the other hand, this approach may not be useful for a large amount of resources because reading large JAD files may cause performance issues. Also note that JAD files address only string resources.

The JAD file

```
exit-en: Exit
exit-fr: Quitter
localeLbl-en: Locale
localeLbl-fr: Locale
text-en: Here's some text.
text-fr: Voici du texte.
textLbl-en: Text
textLbl-fr: Texte
title-en: Localization example
```

Note: In the JAD file above, the French title (`title-fr`) is omitted to demonstrate the usage of default values when retrieving the attributes. See the `localize()` method for implementation details.

Source

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.midlet.MIDlet;
```

```
public class L10nMIDlet extends MIDlet implements CommandListener {
    private Form mainForm;
    private Command exitCommand;
    private String locale;

    /**
```


(C) Copyright Nokia 2014. All rights reserved.

```
     * Constructor. Constructs the object and initializes displayables.
     */
    public L10nMIDlet() {
        // Obtain the system locale
        locale = System.getProperty("microedition.locale");

        initUI();
    }

    public void initUI() {
        // Initialize the form
        mainForm = new Form(localize("title", "Localization example"));

        // Display the locale
        StringItem localeItem = new StringItem(localize("localeLbl", "Locale"),
            locale);
        mainForm.append(localeItem);

        // Display a localized text string
        StringItem textItem = new StringItem(localize("textLbl", "Text"),
            localize("text", "Here's some text."));
        mainForm.append(textItem);

        // Create the exit command
        exitCommand = new Command(localize("exit", "Exit"), Command.EXIT, 1);
        mainForm.addCommand(exitCommand);
        mainForm.setCommandListener(this);
    }

    /**
     * Localizes a string.
     * @param key the key used for identifying a specific value in the
     *      application descriptor file
     * @param def the default value in case "key" is not found from the
     *      application descriptor file
     * @return the message localized; "def" if the key is not found
     */
    private String localize(String key, String def) {
        // Append the current locale to the specified key and use the produced
        // string to obtain the property from the application descriptor file
        String value = getAppProperty(key + "-" + locale);
        if (value == null) {
            value = def;
        }
        return value;
    }

    /**
     * From MIDlet.
     * Called when the MIDlet is started.
     */
    public void startApp() {
        // The initial display is the first form
        Display.getDisplay(this).setCurrent(mainForm);
    }

    /**
```


(C) Copyright Nokia 2014. All rights reserved.

```
     * From MIDlet.
     * Called to signal the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
        // No implementation required
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to terminate.
     * @param unconditional whether the MIDlet has to be unconditionally
     * terminated
     */
    public void destroyApp(boolean unconditional) {
        // No implementation required
    }

    /**
     * From CommandListener.
     * Called by the system to indicate that a command has been invoked on a
     * particular displayable.
     * @param command the command that was invoked
     * @param displayable the displayable where the command was invoked
     */
    public void commandAction(Command command, Displayable displayable) {
        if (command == exitCommand) {
            // Exit the MIDlet
            notifyDestroyed();
        }
    }
}
```

Postconditions

A MIDlet is localized using application attributes.

See also

- Retrieving MIDlet attributes
- Localising text files in Java ME
- Localising resource bundles in Java ME

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.