Modifying a contact in Java ME

Overview

This code snippet demonstrates how to modify contacts in the phone contact list.

When the application starts, it gets the first contact from the phone contact list by using the method `PIM.getInstance().openPIMList(PIM.CONTACT_LIST, PIM.READ_WRITE)` to retrieve the actual contact list and `ContactList.items().nextElement()` to retrieve the first contact.

After this, the user can change any information he or she wants and press "save". Before saving a field to the contact list, the application checks if it already exists and uses methods with the set prefix if the field exists or methods the with add prefix is new.

Source file: ModifyContact.java

```
import java.util.Date;
import java.util.Enumeration;
import java.util.TimeZone;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.TextField;
import javax.microedition.lcdui.DateField;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.midlet.MIDlet;
import javax.microedition.pim.Contact;
import javax.microedition.pim.ContactList;
import javax.microedition.pim.PIM;
import javax.microedition.pim.PIMException;
import javax.microedition.pim.PIMItem;

public class ModifyContact extends MIDlet implements CommandListener {

    //Main form.
    private Form form;
    private Display display;
    private Command exitCommand;
    private Command saveCommand;
    // Text fields for contact data.
    private TextField familyName;
    private TextField givenName;
    private TextField number;
    private DateField birthday;
    private Contact contact;
    private ContactList contactList;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public ModifyContact() {
        form = new Form("Modify PIM Contact");

        // Add command buttons
        saveCommand = new Command("Save", Command.OK, 0);
        exitCommand = new Command("Exit", Command.EXIT, 1);
        form.addCommand(exitCommand);
        form.setCommandListener(this);

        display = Display.getDisplay(this);
        display.setCurrent(form);

        // Add and initialise input fields
        fillContactFields();



    }

    private void fillContactFields() {
        //Retrieve contact list.
        try {
            contactList = (ContactList) PIM.getInstance().openPIMList(
                    PIM.CONTACT_LIST, PIM.READ_WRITE);

            Enumeration enm = contactList.items();
            if (enm.hasMoreElements() == true) {
                contact = (Contact) enm.nextElement();
                form.addCommand(saveCommand);
            } else {
                form.append("There is no contacts in phonebook.");

                return;
            }

            // Create input fields using values from first PIM contact.
            String[] nameArr =
                    new String[contactList.stringArraySize(Contact.NAME)];
            if (contact.countValues(Contact.NAME) > 0) {
                nameArr = contact.getStringArray(Contact.NAME, Contact.ATTR_NONE);
            } else {
                nameArr[Contact.NAME_GIVEN] = "";
                nameArr[Contact.NAME_FAMILY] = "";
```


(C) Copyright Nokia 2013. All rights reserved.

```
            }
            givenName = new TextField("Given name",
                    nameArr[Contact.NAME_GIVEN], 20, TextField.ANY);
            familyName = new TextField("Family name",
                    nameArr[Contact.NAME_FAMILY], 20, TextField.ANY);

            String output;
            if (contact.countValues(Contact.TEL) > 0) {
                output = contact.getString(Contact.TEL, Contact.ATTR_NONE);
            } else {
                output = "";
            }

            number = new TextField("Telephone", output, 20, TextField.ANY);
            birthday = new DateField("Birthdate", DateField.DATE,
                    TimeZone.getTimeZone("GMT"));

            Date date = new Date(0);
            if (contact.countValues(Contact.BIRTHDAY) > 0) {
                date.setTime(contact.getDate(Contact.BIRTHDAY, 0));
            }
            birthday.setDate(date);
        } catch (PIMException e) {
            showError(e.getMessage(), form);
        }

        form.append(givenName);
        form.append(familyName);
        form.append(number);
        form.append(birthday);
    }

    /**
     * From MIDlet.
     * Called when the MIDlet is started.
     */
    public void startApp() {
        // Check for availability of PIM interface.
        if (System.getProperty("microedition.pim.version") == null) {
            showError("PIM API not supported.", form);
        }
    }

    /**
     * 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) {
        try {
            contactList.close();
        } catch (PIMException e) {
        }
    }

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

    /**
     * Save changes to contact list.
     */
    private void saveContact() {
        try {

            // Add values to new contact from input fields
            if (contactList.isSupportedField(Contact.NAME)) {
                String[] name =
                        new String[contactList.stringArraySize(Contact.NAME)];

                if (contactList.isSupportedArrayElement(
                        Contact.NAME, Contact.NAME_FAMILY)) {
                    name[Contact.NAME_FAMILY] = familyName.getString();
                }
                if (contactList.isSupportedArrayElement(
                        Contact.NAME, Contact.NAME_GIVEN)) {
                    name[Contact.NAME_GIVEN] = givenName.getString();
                }

                if (contact.countValues(Contact.NAME) > 0) {
                    contact.setStringArray(
                            Contact.NAME, PIMItem.ATTR_NONE, 0, name);
                } else {
                    contact.addStringArray(
                            Contact.NAME, PIMItem.ATTR_NONE, name);
                }
            }

            if (contactList.isSupportedField(Contact.TEL)) {
                if (contact.countValues(Contact.TEL) > 0) {
                    contact.setString(Contact.TEL, Contact.ATTR_NONE, 0,
                            number.getString());
                } else {
                    contact.addString(Contact.TEL, Contact.ATTR_NONE,
                            number.getString());
                }
            }

            if (contactList.isSupportedField(Contact.BIRTHDAY)) {
                if (contact.countValues(Contact.BIRTHDAY) > 0) {
```


(C) Copyright Nokia 2013. All rights reserved.

```
                contact.setDate(Contact.BIRTHDAY, PIMItem.ATTR_NONE, 0,
                        birthday.getDate().getTime());
            } else {
                contact.addDate(Contact.BIRTHDAY, PIMItem.ATTR_NONE,
                        birthday.getDate().getTime());
            }
        }

        contact.commit();
    } catch (PIMException e) {
        showError(e.getMessage(), form);
    }
}

/**
 * Shows error message
 * @param message the message that will be displayed
 * @param displayable the displayable where the message will be shown
 */
private void showError(String message, Displayable displayable) {
    Alert alert = new Alert("");
    alert.setTitle("Error");
    alert.setString(message);
    alert.setType(AlertType.ERROR);
    alert.setTimeout(5000);
    display.setCurrent(alert, displayable);
}
}
```

Postconditions

The MIDlet shows the first contact from the Phonebook. The user can change any information he or she sees and save it by pressing the "Save" button.

Supplementary material

The source file and the executable application are available for download at Media:Modifying a contact in J2ME.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.