Importing a Phonebook contact in vCard format in Java ME

This code snippet demonstrates how to import a contact from a file to the Phonebook application.

Overview

To import a Phonebook contact from a file:

1. Open the file that contains data for the contact and open the input stream from this file.
2. Set the input stream as the parameter in `PIM.fromSerialFormat`. This method returns an array of imported PIM items. The returned array may contain also other information, so additional checking is needed.
3. Make a copy of each imported contact by calling the `ContactList.importContact` method of the already opened list.
4. Save each copy in the contacts list by executing the `Contact.commit` method on each copied contact object.
5. Close the file.

This MIDlet imports contacts from files in UTF-8 format without any transfer character encoding. Contact data in the file should be in vCard format.

Source file: ImportContact.java

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Alert;

import javax.microedition.pim.PIM;
import javax.microedition.pim.PIMItem;
import javax.microedition.pim.ContactList;
import javax.microedition.pim.Contact;
import javax.microedition.pim.PIMException;

import javax.microedition.io.Connector;
import javax.microedition.io.file.FileConnection;

import java.util.Enumeration;
import java.io.InputStream;

public class ImportContact extends MIDlet implements CommandListener  {

    private Display display;

    // List where contacts will be placed.
    private List contactListCtrl;

    // Command for exporting selected contact. Placed on contactListCtrl.
    private Command cmdImportContact;
    // Command for exiting from application. Placed on contactListCtrl.
    private Command cmdExit;

    // Name of file with contact data to be imported.
    private final String IMPORT_FILENAME = "importcontact.vcf";

    /**
```


(C) Copyright Nokia 2014. All rights reserved.

```
     * Constructor
     */
    public ImportContact() {
        if(checkPIMSupport() == false) {
            exitMIDlet();
        }

        initializeComponents();
    }

    /**
     * Initializes components of midlet.
     */
    private void initializeComponents() {
        // Get display
        display = Display.getDisplay(this);

        // Create list control
        contactListCtrl = new List("contacts", List.IMPLICIT);

        cmdImportContact = new Command("Import contact", Command.SCREEN, 0);
        contactListCtrl.addCommand(cmdImportContact);

        cmdExit = new Command("Exit", Command.EXIT, 0);
        contactListCtrl.addCommand(cmdExit);

        contactListCtrl.setCommandListener(this);

        // Add contacts to list control.
        addContactsToListCtrl();
    }

    /**
     * Checks PIM support.
     * @return - true if PIM is supported, false otherwise.
     */
    private boolean checkPIMSupport() {
        String propValue = System.getProperty("microedition.pim.version");
        if(propValue != null) {
            return true;
        } else {
            return false;
        }
    }

    /**
     * Adds contacts to list.
     */
    private void addContactsToListCtrl() {
        try {
            // Get list of contacts.
            ContactList contactList = (ContactList)PIM.getInstance().openPIMList(
                    PIM.CONTACT_LIST, PIM.READ_WRITE);

            // Check if "name" field of contact item is supported.
            // If not supported, exit from application.
            if(contactList.isSupportedField(Contact.NAME) == false) {
```


(C) Copyright Nokia 2014. All rights reserved.

```
                throw new Exception();
            }

            // Clear list control.
            contactListCtrl.deleteAll();

            // Add contacts to list control.
            Enumeration contacts = contactList.items();
            while(contacts.hasMoreElements() == true) {
                Contact contact = (Contact)contacts.nextElement();
                String[] name = contact.getStringArray(Contact.NAME,
                        PIMItem.ATTR_NONE);
                String family = name[Contact.NAME_FAMILY];
                // Add contact's summary to list control
                contactListCtrl.append(family, null);
            }

            // Close list of contacts.
            contactList.close();

        } catch(PIMException pimExc) {
            // TODO: Handle error on working with PIM.
        } catch(SecurityException secExc) {
            // TODO: Handle error on access to PIM.
        } catch(Exception exc) {
            // If unknown error was catched, exit from application.
            exitMIDlet();
        }
    }

    /**
     * Import contact from file placed private folder.
     * @return true if contact was imported successfully, false - otherwise.
     */
    private boolean importContactFromFile() {
        try {
        // Open file for importing contact.
        String path = System.getProperty("fileconn.dir.memorycard");
        String fileName = path + IMPORT_FILENAME;

        FileConnection file = (FileConnection)Connector.open(fileName,
                Connector.READ_WRITE);

        // If there is no file then create it
        if(file.exists() == false) {
            throw new Exception("File for importing was not found.");
        }

        // Export contact.
        InputStream inStream = file.openInputStream();

        PIMItem[] items = PIM.getInstance().fromSerialFormat(inStream, "UTF-8");
        // Assume what there is only one contact item in file.
        if(items.length > 0) {
            importContactToList((Contact)items[0]);
        }
```


(C) Copyright Nokia 2014. All rights reserved.

```
        // Close file.
        inStream.close();
        file.close();

        showAlert("Info", "Contact was imported from file.");

        } catch(Exception exc) {
            showAlert("Error", exc.getMessage());
            return false;
        }

        return true;
    }

    /**
     * Imports contact to phonebook list of contacts.
     * @param importedContact - contact to be imported.
     * @throws javax.microedition.pim.PIMException - on PIM error.
     */
    private void importContactToList(Contact importedContact)
            throws PIMException {
        // Open list of contacts
        ContactList contactList = (ContactList)PIM.getInstance().openPIMList(
                PIM.CONTACT_LIST, PIM.READ_WRITE);

        // contact list creates new contact based on imported contact
        Contact newcontact = contactList.importContact(importedContact);

        // Save new contact to list
        newcontact.commit();
        // Close list of contacts
        contactList.close();
    }

    /**
     * Shows alert with specified title and text.
     * @param title - Title of alert.
     * @param message - text of alert.
     */
    private void showAlert(String title, String message) {
        Alert alert = new Alert(title);
        alert.setString(message);
        alert.setTimeout(Alert.FOREVER);
        display.setCurrent(alert);
    }

    /**
     * From MIDlet.
     * Signals the MIDlet that it has entered the Active state.
     */
    public void startApp() {
        display.setCurrent(contactListCtrl);
    }

    /**
     * From MIDlet.
```


(C) Copyright Nokia 2014. All rights reserved.

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

    /**
     * From MIDlet.
     * Signals the MIDlet to terminate and enter the Destroyed state.
     */
    public void destroyApp(boolean unconditional) {
        // No implementation required.
    }

    /**
     * Performs exit from midlet.
     */
    private void exitMIDlet() {
        notifyDestroyed();
    }

    /**
     * From CommandListener.
     * Indicates that a command contact has occurred on Displayable displayable.
     * @param command - a Command object identifying the command.
     * @param displayable - the Displayable on which this contact has occurred.
     */
    public void commandAction(Command command, Displayable displayable) {
        // Handles "show details" command on selected contact.
        if(command == cmdImportContact) {
            if(importContactFromFile() == true) {
                addContactsToListCtrl();
            }
        }
        // Handles "exit" command.
        if(command == cmdExit) {
            exitMIDlet();
        }
    }
}
```

Postconditions

A list of phonebook contacts is shown on the display.

By pressing the "Import contact" menu command the user can import a contact from the file. The file with contact to be imported should be in the root directory of the memory card and named IMPORT_FILENAME constant. After importing, the contact list control displays the changes in the Phonebook contact list.

Supplementary material

Executables and source files are available at Media:ImportingContact.zip.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.