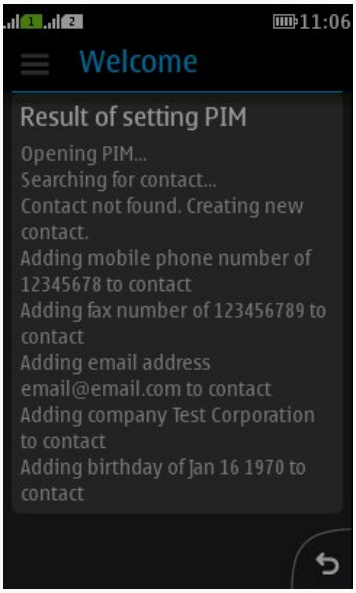
Adding and updating a PIM contact using Java ME

Overview

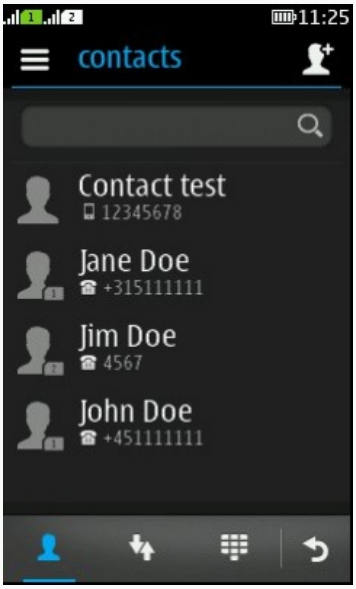
This MIDlet uses the FileConnection and PIM API (JSR-75) to add a new contact to the contacts database and to update that contact.

This code includes a workaround for S60 3rd Edition, Feature Pack 1 devices affected by the known issue Archived:Contact update deletes contact information in S60 3rd Edition FP1 (Known Issue).

The workaround part has been commented within the code as described in the known issue.



Adding the Contact upon MIDlet launch



Contact Added

Contact Details

Source file

```
import java.io.IOException;
import java.io.InputStream;
import java.util.Enumeration;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.pim.*;


public class AddContact extends MIDlet implements CommandListener {

   private StringBuffer pimResult;
   private Command exitCmd;
   private Form f;
   private StringItem si;
   private Display d;

   public AddContact() {}

      public void startApp() {
          d = Display.getDisplay(this);
          f = new Form("Welcome", new Item[] { getStringItem() });
          exitCmd = new Command("Exit", Command.EXIT, 0);
           f.addCommand(exitCmd);
           f.setCommandListener(this);
           d.setCurrent(f);
   }

   public void commandAction(Command cmd, Displayable displayable) {

           if (cmd == exitCmd) {
                  destroyApp(true);
                  notifyDestroyed();
           }
   }

   public StringItem getStringItem() {
       if (si == null) {

           //Displayable info will be gathered into a StringBuffer which
           //need to be turned into a StringItem before displaying on Form.

           pimResult = new StringBuffer();

           String phoneNo = "12345678";
```


(C) Copyright Nokia 2013. All rights reserved.

```
            ContactList clist = null;

            try {
                pimResult.append("Opening PIM...\n");
                clist = (ContactList) PIM.getInstance().openPIMList(PIM.CONTACT_LIST, PIM.READ_WRITE);
            } catch (PIMException e) {
                pimResult.append("Error opening PIM: " + e.getMessage() + "\n");
                return new StringItem("Result of setting PIM ", pimResult.toString());
            }

            Enumeration itemsEnum = null;
            InputStream is = null;

            Contact searchPattern = null;
            if (clist == null) {
                return new StringItem("Result of setting PIM", pimResult.toString() + "Contact list not opened\n");
            }

            Contact result = null;

            try {
                result = clist.createContact();
                searchPattern = clist.createContact();
                searchPattern.addString(Contact.TEL, Contact.ATTR_NONE, phoneNo);

                pimResult.append("Searching for contact...\n");
                itemsEnum = clist.items(searchPattern);

                boolean wasFound = false;

                // Contact found, append the existing contact data to the StringBuffer

                if (itemsEnum != null && itemsEnum.hasMoreElements()) {
                    pimResult.append("Contact found, \nupdating the contact.\n");
                    wasFound = true;
                    result = (Contact) itemsEnum.nextElement();

                    /*Workaround for S60 3rd Edition, Feature Pack 1 devices
                    which are affected by the Known Issue KI<nro>:

                    PIMItem.getFields() returns all the fields that have data
                    stored for them. This method enable access to all the
                    fields with data instead of iterating through all
                    supported fields and checking each of them individually.

                    */

                    '''/***** PIMItem.getFields() starts here *****/'''
                    /*
                    //Let's start by going through all the fields with data
                    //and store them as an integer array.

                    int[] fields = result.getFields();

                    //Let's use a for loop to go through all the stored fields
                    //and use their indexes as Strings

                    for(int i = 0; i < fields.length; i++) {

                    String str = Integer.toString(fields[i]);

                    //Using PIMItem.getFields().getFieldLabel() to retrieve
                    //the String labels for the fields returned by PIMItem.getFields().

                    String label = result.getPIMList().getFieldLabel(fields[i]);
                    //Let's append both the Strings (index and label)
                    //into the StringBuffer for presenting them on the Form.

                    pimResult.append(str +" "+ label);
                    pimResult.append("\n");
                    } '''*/  /***** PIMItem.getFields() ends here *****/'''
                }

                // If the contact does not exist, let's create a new one.
                // Let's name the new contact as "Contact test" and include
                // telephone, email, organisation and birthday data for
                // the contact.

                if (!wasFound) {
                    pimResult.append("Contact not found. Creating new contact.\n");
                    String[] name = new String[clist.stringArraySize(Contact.NAME)];
                    if (clist.isSupportedArrayElement(Contact.NAME, Contact.NAME_GIVEN)) {
                        name[Contact.NAME_GIVEN] = "Contact test";
                    }
                    result.addStringArray(Contact.NAME, PIMItem.ATTR_NONE, name);

                     if (clist.isSupportedField(Contact.TEL)) {
                        result.addString(Contact.TEL, Contact.ATTR_MOBILE, phoneNo);
                        pimResult.append("Adding mobile phone number of " + phoneNo + " to contact\n");
                        result.addString(Contact.TEL, Contact.ATTR_FAX, "123456789");
                        pimResult.append("Adding fax number of 123456789 to contact\n");
                    }

                    if (clist.isSupportedField(Contact.EMAIL)) {
                        result.addString(Contact.EMAIL, Contact.ATTR_NONE, "email@email.com");
                        pimResult.append("Adding email address email@email.com to contact\n");
                    }

                    if (clist.isSupportedField(Contact.ORG)) {
                        result.addString(Contact.ORG, Contact.ATTR_NONE, "Test Corporation");
                        pimResult.append("Adding company Test Corporation to contact\n");
                    }

                    if (clist.isSupportedField(Contact.BIRTHDAY)) {
                        result.addDate(Contact.BIRTHDAY, Contact.ATTR_NONE, 1304640000);
                        pimResult.append("Adding birthday of Jan 16 1970 to contact\n");
                    }
                }

                    try {

                        result.commit();

                    } catch (PIMException f) {
                       pimResult.append("PIMException thrown :" + f.getMessage() + "\n");
                    }

            } catch (Exception e) {
                pimResult.append("Exception thrown: " + e.getMessage());
            } finally {
```


(C) Copyright Nokia 2013. All rights reserved.

```
                searchPattern = null;
                result = null;
                itemsEnum = null;
                if (is != null) {
                    try {
                        is.close();
                    } catch (IOException ex) {
                        pimResult.append("IOException thrown closing inputstream " + ex.getMessage());
                    }
                    is = null;
                }
                if (clist != null) {
                    try {
                        clist.close();
                    } catch (PIMException e) {
                        pimResult.append("PIMException thrown closing contact list " + e.getMessage());
                    }
                    clist = null;
                }
            }
            //Converting the StringBuffer into a StringItem
            si = new StringItem("Result of setting PIM", pimResult.toString());
        }
        return si;
    }


    public void pauseApp() {

    }

    public void destroyApp(boolean unconditional) {
    }

}
```

Postcondition

A new contact has been added to the contacts database.

See also

Archived:Contact update deletes contact information in S60 3rd Edition FP1 (Known Issue)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.