# Listing contacts in Java ME

This code example demonstrates how to list contacts in Java ME using the PIM API.

To read the contacts, use the method `ContactList.items`. There are several methods with this name, but in this snippet the one that has no parameters is used. For more information, see the Javadoc for JSR-75 PIM API.

## Preconditions

In order to test this snippet there should be at least one contact in the database. The application also needs permission to read and write PIM database data.

## Source

```
private List listContacts;

/**
 * Holds reference to the currently opened contact list.
 */
private ContactList contactList;

private static final Command BACK_COMMAND =
        new Command("Back", Command.BACK, 0);
```

```
/**
 * Instantiates a listContacts.
 */
private void setupContactList() {
    listContacts = new List("Contact list", Choice.IMPLICIT);
    listContacts.addCommand(BACK_COMMAND);
    listContacts.setCommandListener(this);
}
```

```
/**
 * Executes the snippet.
 */
private void executeSnippet() {
    openContactList();
}

/**
 * Fills listContacts fith data read from PIM database and brings
 * the list to the foreground.
 */
private void openContactList() {
    fillContactList();
    display.setCurrent(listContacts);
    printString("Done");
}

/**
 * Reads contacts list from PIM database 'Contacts' and fills listContacts
 * with contact 'name' field value.
 */
private void fillContactList() {
    try {
        printString("Opening contact list ...");
        contactList = (ContactList)PIM.getInstance().openPIMList(
                PIM.CONTACT_LIST, PIM.READ_WRITE);

    } catch (PIMException ex) {
        printString(ex.toString());
        return;
    }
    Enumeration contacts = null;
    Contact contact = null;
    try {
        printString("Getting contact list items");
        contacts = contactList.items();
    } catch (PIMException ex) {
        releaseContactList();
        printString(ex.toString());
        return;
    }
    // Checking contact count
    if (!contacts.hasMoreElements()) {
        printString("Contact list IS empty!");
    }
    if (listContacts.size() > 0) {
        listContacts.deleteAll();
    }
    while (contacts.hasMoreElements()) {
        contact = (Contact)contacts.nextElement();
        String contactInfo = contact.getStringArray(Contact.NAME,
                Contact.ATTR_NONE)[Contact.NAME_GIVEN];
        if (contactInfo != null) {
            listContacts.append(contactInfo, null);
        }
    }
}

/**
 * Closes the contact list if one is opened.
 */
private void releaseContactList() {
    if (contactList != null) {
```


(C) Copyright Nokia 2013. All rights reserved.

```
        try {
            printString("Closing contact list");
            contactList.close();
        } catch (PIMException ex) {
            printString(ex.toString());
        }
    }
    contactList = null;
}
```

```
/**
 * 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 == BACK_COMMAND) {
        display.setCurrent(mainForm);
        // Close ContactList object instance
        releaseContactList();
    }
}
```

## Postconditions

The contacts in the phonebook are listed on the screen.

## 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 to 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:ListContacts.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:ListContacts.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.