NFC UIDReader using Java ME

This Java ME code example shows a basic MIDlet the reads the UID of an external Tag/Transponder and displays it on the screen. This was initially shown at NFC Forum Developer Training, WIMA 2008 and NFC-Congress Hagenberg 2008.

Warning: This example was created for the NFC implementation provided on Series 40 devices, and may include use of features not yet available for NFC on Symbian devices.

Download appropriate NetBeans Project: File:UIDReader.zip.

```
package at.nfcresearch.wima.examples;

// Packages for contactless Communcation
import javax.microedition.contactless.ContactlessException;
import javax.microedition.contactless.DiscoveryManager;
import javax.microedition.contactless.TargetProperties;
import javax.microedition.contactless.TargetListener;
import javax.microedition.contactless.TargetType;

// Packages for GUI Stuff
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Form;

import javax.microedition.midlet.*;

public class UIDReader extends MIDlet implements TargetListener, CommandListener {

    // Display Stuff
    private Command exitCommand;
    private Form form;

    public UIDReader() {

        // Create the GUI
        exitCommand = new Command("Exit", Command.EXIT, 1);
        form = new Form("NFC-Research.at: UID Reader");

        form.addCommand(exitCommand);
        form.append("Touch Tag to read ID.");
        form.setCommandListener(this);

        // Registration of the TargetListener for external contactless
        // Targets (in this RFID_TAG).
        try {
            DiscoveryManager dm = DiscoveryManager.getInstance();
            dm.addTargetListener(this, TargetType.NDEF_TAG);

        } catch (ContactlessException ce) {
            displayAlert("Unable to register TargetListener: " + ce.toString(), AlertType.ERROR);
        }

    }

    public void startApp() {
        Display.getDisplay(this).setCurrent(form);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    /**
     * Implementation of the Call-Back Function of the TargetListener
     * @param targetProperties: Array of Targets found by the Phone
     *
     */
    public void targetDetected(TargetProperties[] targetProperties) {

        // in case no targets found, exit the method
        if (targetProperties.length == 0) {
            return;
        }

        // show the UID of the first tag found
        TargetProperties tmp = targetProperties[0];
        displayAlert("UID read: " + tmp.getUid(), AlertType.INFO);
    }

    /**
     * Implementation of the Call-Back function of the CommandListener
     * @param command: command key pressed
     * @param displayable: associated displayable Object
     */
    public void commandAction(Command command, Displayable displayable) {
        if (command == exitCommand) {
            DiscoveryManager dm = DiscoveryManager.getInstance();
            dm.removeTargetListener(this, TargetType.NDEF_TAG);
            destroyApp(false);
            notifyDestroyed();
        }

    }

    private void displayAlert(String error, AlertType type) {
        Alert err = new Alert(form.getTitle(), error, null, type);
        Display.getDisplay(this).setCurrent(err, form);
    }
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.