Detect external ISO14443 card

Simple MIDlet that detects external ISO14443 card and shows its UID and the URL that can be used to open a connection to it.

```
package test;

import javax.microedition.contactless.DiscoveryManager;
import javax.microedition.contactless.TargetListener;
import javax.microedition.contactless.TargetProperties;
import javax.microedition.contactless.TargetType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

/**
 * Simple midlet that detects external ISO14443 card and shows its uid and the
 * url that can be used to open a connection to it.
 */
public class ShowISO14443Connection extends MIDlet implements TargetListener,
        CommandListener {
    /**
     * Form to show information about detected card
     */
    Form form = new Form("ShowISO1443Connection");

    /**
     * Exit command
     */
    Command exitCmd = new Command("Exit", 1, Command.OK);

    /**
     * Constructor.
     */
    public ShowISO14443Connection() {
        super();
    }

    /*
     * (non-Javadoc)
     *
     * @see javax.microedition.midlet.MIDlet#startApp()
     *
     * Sets the form as current displayable and adds target listener for
     * ISO14443.
     */
    protected void startApp() throws MIDletStateChangeException {
        // Set form as current displayable
        form.addCommand(exitCmd);
        form.setCommandListener(this);
        Display.getDisplay(this).setCurrent(form);

        // Add target listener for ISO14443 cards
        try {
            DiscoveryManager.getInstance().addTargetListener(this,
                    TargetType.ISO14443_CARD);
        } catch (Exception ex) {
            // Show any exceptions if adding the target listener fails.
            form.append(ex.toString());
        }
    }

    protected void pauseApp() {
    }

    /*
     * (non-Javadoc)
     *
     * @see javax.microedition.midlet.MIDlet#destroyApp(boolean)
     *
     * Remove the target listener when done.
     */
    protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
        try {
            DiscoveryManager.getInstance().removeTargetListener(this,
                    TargetType.ISO14443_CARD);
        } catch (Exception ex) {
            // Ignore exceptions at this point
        }
    }

    /*
     * (non-Javadoc)
     *
     * @see javax.microedition.contactless.TargetListener#targetDetected(javax.microedition.contactless.TargetProperties[])
     *
     * Display uid and url of the detected (ISO14443) card.
     */
    public void targetDetected(TargetProperties[] targetProperties) {
        for (int i = 0; i < targetProperties.length; i++) {
            TargetProperties tp = targetProperties[i];

            // Show the uid
            form.append("uid: " + tp.getUid() + "\n");

            // Make sure that detected card is ISO14443.
            // This check is only needed if we are listening for other targets
            // as well.
            if (tp.hasTargetType(TargetType.ISO14443_CARD)) {
                form.append("Target is ISO14443_CARD\n");

                // Show the url to use to open the connection to the ISO14443
                // card
                try {
                    form.append("url: " + tp.getUrl(Class.forName(
                            "javax.microedition.contactless.sc.ISO14443Connection"))
```


(C) Copyright Nokia 2013. All rights reserved.

```
                + "\n");
            } catch (Exception ex) {
                form.append(ex.toString());
            }
        } else {
            // Never happens in this example
            form.append("Target is NOT ISO14443_CARD\n");
        }
    }
}

/*
 * (non-Javadoc)
 *
 * @see javax.microedition.lcdui.CommandListener#commandAction(javax.microedition.lcdui.Command,
 *      javax.microedition.lcdui.Displayable)
 *
 * Handle exit command.
 */
public void commandAction(Command c, Displayable d) {
    if (c == exitCmd) {
        exit();
    }
}

/**
 * Exit the midlet.
 */
private void exit() {
    try {
        destroyApp(false);
        notifyDestroyed();
    } catch (MIDletStateChangeException ex) {
    }
}
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.