NFC Secure Element Example External Reader

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.

This example show the use of JDK 6.0 and the javax.smartcardio.* package in order to communicate with the Java Card Applet in the secure element of the handset. Download appropriate NetBeans Project: File:ReaderApplication.zip.
Make sure you have JDK 6.0 and a PC/SC Reader (NOT NXP's PEGODA!) installed.

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

import javax.smartcardio.*;
import java.util.List;
import java.util.ListIterator;

public class Main {

    private static byte[] SELECT = {(byte) 0x00, (byte) 0xA4, (byte) 0x04, (byte) 0x00,
(byte) 0x09, (byte) 0x74, (byte) 0x69, (byte) 0x63, (byte) 0x6B, (byte) 0x65, (byte)
0x74, (byte) 0x69, (byte) 0x6E, (byte) 0x67, (byte) 0x00};
    private static byte[] INS_INC = {(byte) 0x00, (byte) 0x01, (byte) 0x00, (byte)
0x00};
    private static byte[] INS_DEC = {(byte) 0x00, (byte) 0x02, (byte) 0x00, (byte)
0x00};
    private static byte[] INS_READ = {(byte) 0x00, (byte) 0x03, (byte) 0x00, (byte)
0x00, (byte) 0x01};
    private static CommandAPDU SELECT_APDU = new CommandAPDU(SELECT);
    private static CommandAPDU INC_APDU = new CommandAPDU(INS_INC);
    private static CommandAPDU DEC_APDU = new CommandAPDU(INS_DEC);
    private static CommandAPDU READ_APDU = new CommandAPDU(INS_READ);

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        TerminalFactory tf = TerminalFactory.getDefault();
        CardTerminals ct = tf.terminals();
        List<CardTerminal> l = null;
        Card card = null;

        try {
            l = ct.list();
        } catch (Exception e) {
            System.out.println("Error listing Terminals: " + e.toString());
        }

        System.out.println("List of PC/SC Readers connected:");
        ListIterator i = l.listIterator();
        while (i.hasNext()) {
            System.out.println("Reader: " + ((CardTerminal) i.next()).getName());
        }

        // IMPORTANT: change this to the PC/SC Name of your reader!!
        CardTerminal c = ct.getTerminal("OMNIKEY CardMan 5x21-CL 0");
```


(C) Copyright Nokia 2014. All rights reserved.

```
        System.out.println("Terminal fetched");

        try {
            while (c.isCardPresent()) {

                try {
                    Thread.sleep(100);
                } catch (Exception e) {
                    System.out.print(("Error Sleep"));
                }

                try {
                    card = c.connect("T=1");
                    System.out.println("Terminal connected");
                } catch (Exception e) {
                    System.out.println("Terminal NOT onnected: " + e.toString());
                }

                System.out.println("ATR: " + arrayToHex(((ATR)
card.getATR()).getBytes()));

                CardChannel ch = card.getBasicChannel();

                byte[] x = null;

                if (check9000(ch.transmit(SELECT_APDU))) {
                    System.out.println("SELECT OKAY");
                } else {
                    System.out.println("SELECT NOT OKAY");
                    return;
                }

                ResponseAPDU ra = ch.transmit(READ_APDU);
                if (check9000(ra)) {
                    System.out.println("Value: " + ra.getBytes()[0]);
                } else {
                    System.out.println("Error Reading Value");
                }

                return;




            }// while
        }// try
        catch (CardException e) {
            System.out.println("Error isCardPresent()" + e.toString());
        }

    }

    public static boolean check9000(ResponseAPDU ra) {
        byte[] response = ra.getBytes();
        return (response[response.length - 2] == (byte) 0x90 && response[response.length
- 1] == (byte) 0x00);
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
    public static String arrayToHex(byte[] data) {
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < data.length; i++) {
            String bs = Integer.toHexString(data[i] & 0xFF);
            if (bs.length() == 1) {
                sb.append(0);
            }
            sb.append(bs);
        }

        return sb.toString();
    }
}
```


(C) Copyright Nokia 2014. All rights reserved.