

```
//*****//
//Prosty program odczytujący zawartość certyfikatu z karty Cryptoflex 8K //
//Schlumberger za pomocą rozkazów APDU karty. Program może odczytywać //
//zawartość dowolnego pliku. //
//*****//

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.lang.String;
import slb.iop.*;

public class SlbCardFileRead extends Applet
    implements SmartCardListener, IOPListener{

    public TextArea SendStringText;
    public TextField status;
    public TextField eventText;
    // public TextArea cardText;
    Font font = new Font("Verdana",Font.PLAIN, 10);

    SmartCard slbsm = new SmartCard();
    IOP iop = new IOP();

    WndFrame fr = new WndFrame("Zawartosc karty");

    String[] readers = iop.ListReaders();
    String[] cards = iop.ListKnownCards();

    public void DataSent(SmartCardEvent evt) {
        eventText.setText("DataSent event fired");
    }

    public void DataReceived(SmartCardEvent evt) {
        eventText.setText("odbieram dane ...");
    }

    public void CardInserted(IOPEvent evt) {
        eventText.setText("karta wsadzona... ");
        getSLBInfo ();
        fr.show();

        //eventText.setText("sprawdzam katalog ...");
        //short [] scdir = scDirectory ();
        //eventText.setText("sprawdzilem katalog");

        //eventText.setText("pokazuje wynik sprawdzania: " +
        //    new Integer(scdir.length).toString ());
    }

    public void CardRemoved (IOPEvent ctEvent) {
        eventText.setText("karta wyjeta... ");
        fr.hide ();
    }

    public void init() {

        setLayout(null);
        setBackground(new Color(Integer.parseInt("000032", 16)));
        setFont(font);
        status = new TextField(25);
```

```
status.setEditable(false);
status.setBounds(10,270,420,20);
eventText = new TextField(25);
eventText.setEditable(false);
eventText.setBounds(10,310,420,20);
eventText.setText("prosze wsadzc karte");

SendStringText = new TextArea();
SendStringText.setBounds(10,10,420,240);
SendStringText.setEnabled(true);

// cardText = new TextArea();
// cardText.setBounds(340,10,190,240);
// cardText.setEnabled(true);

add(SendStringText);
add(status);
add(eventText);
// add(cardText);
iop.addIOPListener(this);
slbsm.addSmartCardListener(this);
}

public void start() {
    showStatus ();
}

public void getSLBinfo () {

    eventText.setText("polaczenie z karta... ");
    if (iop.Connect(slbsm, readers[0])) {
        eventText.setText("polaczenie z karta...OK. ");
    }
    else {
        eventText.setText("polaczenie z karta nieudane.");
    }
    status.setText("Nazwa karty w czytniku: " + slbsm.GetCardName());

    eventText.setText("weryfikacja kodu transportowego...");

    int[] verifykey = {0x47,0x46,0x58,0x49,0x32,0x56,0x78,0x40};
    short[] readbinary = null;

    try {
        if (slbsm.VerifyKey(1, verifykey)){

            eventText.setText("weryfikacja kodu transportowego... OK");

//otwiera katalog na karcie 3F11 nastepnie 0030 gdzie znajdują sie certyfikaty//

            if (slbsm.Select("3F11")) {
                eventText.setText("otwieram 3F11, czytam ...");
            }
            else {
                eventText.setText(" ... nie został otwarty 3F11");
            }
            if (slbsm.Select("0030")) {
                eventText.setText("otwieram 0030, czytam ...");
            }
            else {
                eventText.setText(" ... nie został otwarty 0030");
            }
        }
    }
}
```

```
int count = 1910;
String str = "\n";
char[] readchr = null;

readbinary = slbsm.ReadBinary(0x00, count);
int ii = 0;
for (int i = 0; i < count; i++) {
    if ((readbinary[i] > 31) & (readbinary[i] < 127)){
        str = str + (char)readbinary[i];
    }
    else str = str + ".";
    if (ii == 80) {
        // cardText.append(str + "\n");
        fr.addToWnd (str + "\n");
        ii = 0;
        str = "";
    }
    ii++;
}

//addLine(str + "\n");
str = "\n";
if (slbsm.Select("0020")) {
    eventText.setText("Otwieram 0020, czytam ...");
    readbinary = slbsm.ReadBinary(0, 70);
    for (int i = 0; i < 70; i++) {
        if ((readbinary[i] > 31) & (readbinary[i] < 127))
            str = str + (char)readbinary[i];
    }
    addLine(str + "\n");
}
else {
    eventText.setText("... nie został otwarty 0020");
}
eventText.setText("");
}
else eventText.setText("kod transportowy nie został zweryfikowany");

}
catch (slbException s) {
    addLine(s.getMessage());
}

}

public void addLine(String str) {
    SendStringText.append(str);
}

public void showStatus () {
    addLine("\n");
    addLine("Karty zarejestrowane w systemie: " + "\n");
    for (int i = 0; i < cards.length; i++) {
        addLine(cards[i] + "\n");
    }

    addLine("\n");
    addLine("Czytniki zarejestrowane w systemie: " + "\n");
    for (int i = 0; i < readers.length; i++) {
        addLine(readers[i] + "\n");
    }
}
}
```

```
public short[] scDirectory (){
    short[] apdu = {0,0,0,0,0,0,0,0,0};

    int CLA    = 0xF0;
    int INS    = 0xA8;
    int p1     = 0x00;
    int p2     = 0x00;
    // int[] DataIn = NULL;
    int LengthOut = 10;

    try {
        apdu = slbasm.SendCardAPDU(CLA, INS, p1, p2, null, LengthOut);
    }
    catch (slbException e) {
        status.setText("Directory: " + e);
    }
    return apdu;
}

class WndFrame extends Frame {

    TextArea text = new TextArea ();
    Font font = new Font("Verdana",Font.PLAIN, 10);

    WndFrame (String str){
        setTitle(str);
        setResizable(false);
        resize(510,330);
        setFont(font);
        setBackground(Color.lightGray);
        setLayout(null);
        text.setBounds(10,30,490,290);
        this.add(text);
    }

    public void addToWnd(String str){
        text.append(str);
    }

    public boolean handleEvent(Event evn) {
        if (evn.id == Event.WINDOW_DESTROY) {
            hide ();
            //dispose ();
            return true;
        }
        else
            return super.handleEvent (evn);
    }
}
```