www.tyfone.com

SIDE-X™ TEST TOOL

User Guide – Win/Mac/Linux

Revision History

Date	Remarks	Version
08/30/2017	Initial draft	1.0
02/28/2018	Release updates with screenshots	1.1
03/28/2018	Applet load, Delete Applet feature updates	1.2
04/03/2018	Reformatting	1.3

Trademarks and copyright

For questions visit: www.tyfone.com

Proprietary Notice

Disclaimers

Every effort has been made to ensure the accuracy of all information and statements contained in this document. However, Tyfone does not assume any liability for the use of this material. Tyfone reserves the right to make changes to this material at any time and without notice.

Customer support

Tyfone offers multiple technical support plans and service packages to help our customers get the most out of Tyfone products. For further information on Technical Support plans and pricing please send an email to support@tyfone.com.

To provide feedback on this document, send your comments to feedback@tyfone.com.

Contents

1 Overview

This document explains how to install and use the Side-X Test Tool on a Win/Mac/Linux platform. It is a piece of software which is used to load an applet into the SideCard and to communicate with the SideCard using Application Protocol Data Unit (APDU) commands. The application communicates with the SideCard via contact interface using ISO 7816 protocol. Requirements to run the Side-X Test Tool are defined below.

1.1 Hardware Requirements

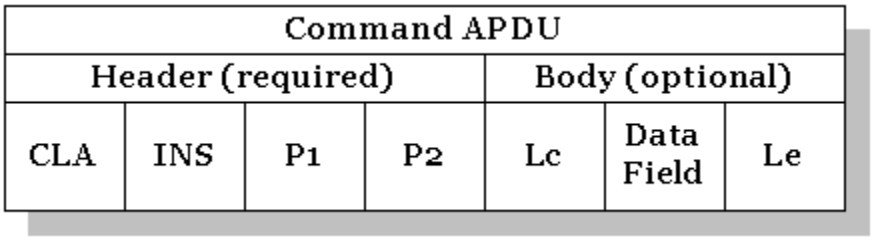
1. SideCard (which is pre-personalized)
2. Smart Card Reader
3. Desktop/Laptop System

1.2 Software Requirements

- Operating Systems
  1. Windows/Linux/MacOS
- Software
  1. Java, JRE 1.6 onwards
  2. Smart Card Reader Driver

2 APDU Communication

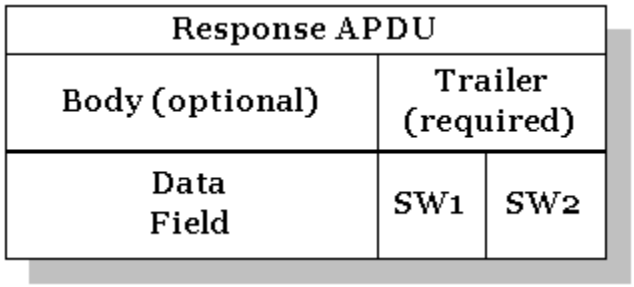
Application Protocol Data Unit command is a set of bytes. APDU communication involves sending a command APDU to the Firmware or Secure Element (SE) of a smart card, processing it, and receiving a response APDU. APDU commands are a queue of binary numbers of the following form:



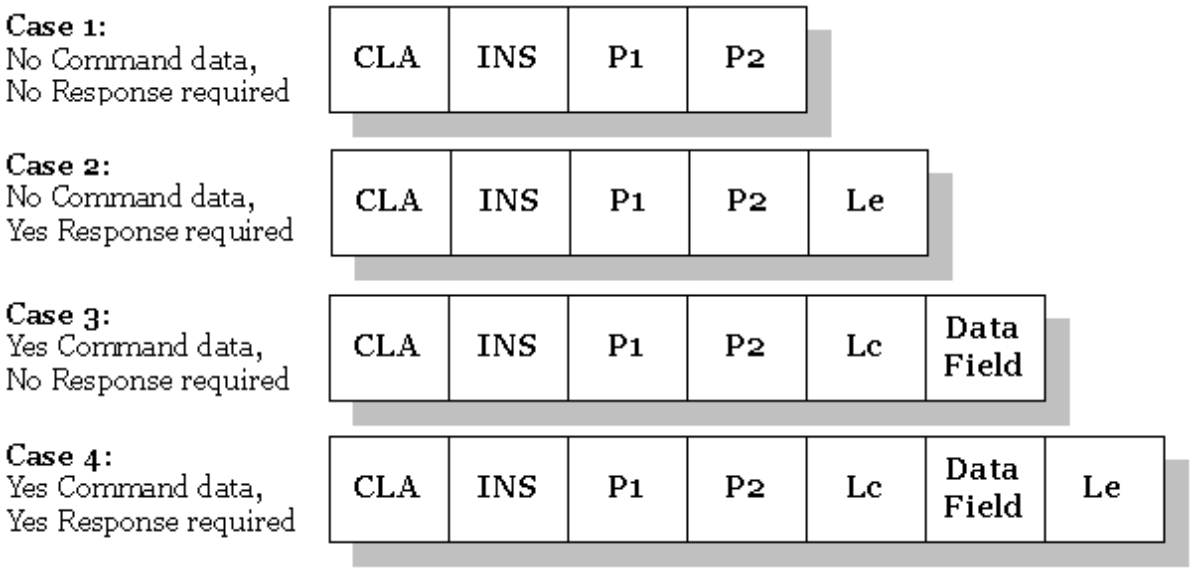
The first four sections, i.e. *CLA*, *INS*, *P1* and *P2* are mandatory in all APDU commands and **each one has one-byte length**. These one-byte length sections stand for Class, Instruction, Parameter1 and Parameter2 respectively.

The last three sections, i.e. *Lc*, *Data Field* (*CData),* and *Le* are optional. Lc represents the length of the CData field. Le specifies the maximum length of expected response data.

The **response APDU** is expected to contain response data with status words: SW1 SW2; which denote processing state in the smart card.



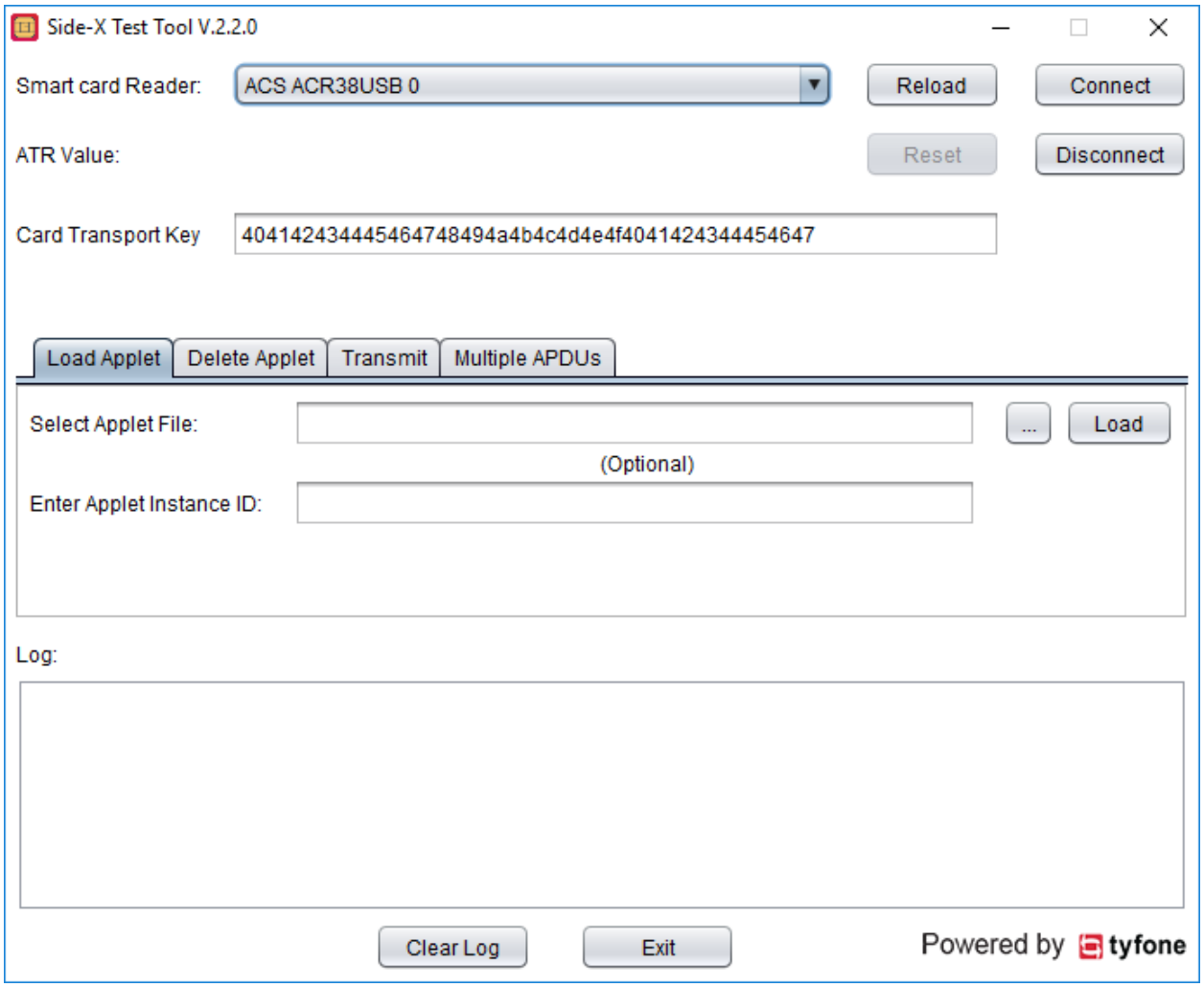
Different types of APDU commands:



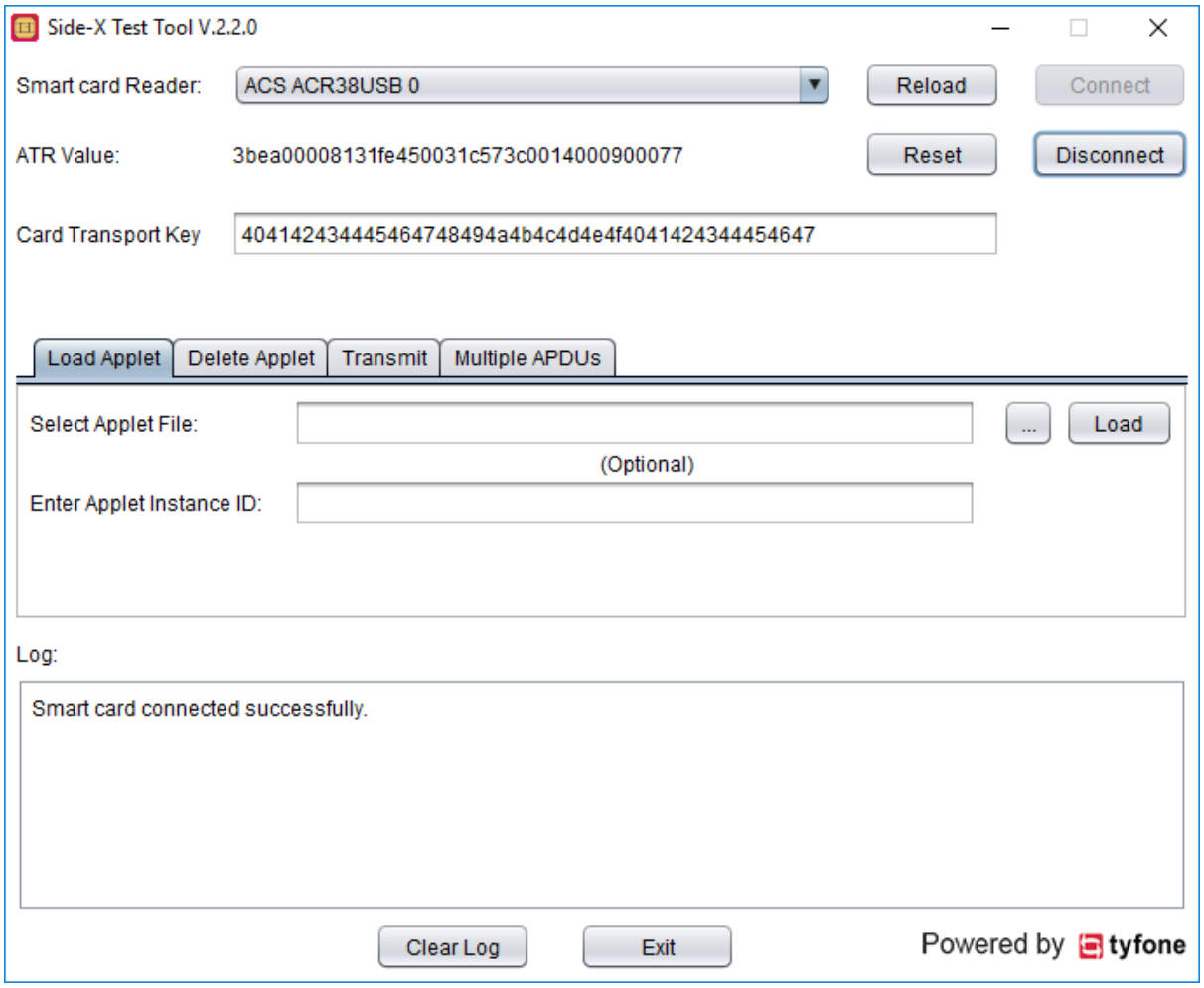
3 RUNNING SIDE-X TEST TOOL

1. Connect smart card reader to the desktop/laptop system and make sure that the connected smart card reader is detected by the system/laptop. If not, please install appropriate smart card reader driver.
2. Insert SideCard into the smart card reader.
3. Open the Side-X Test Tool by double clicking on Side-X_TestTool_<X.Y.Z>.jar

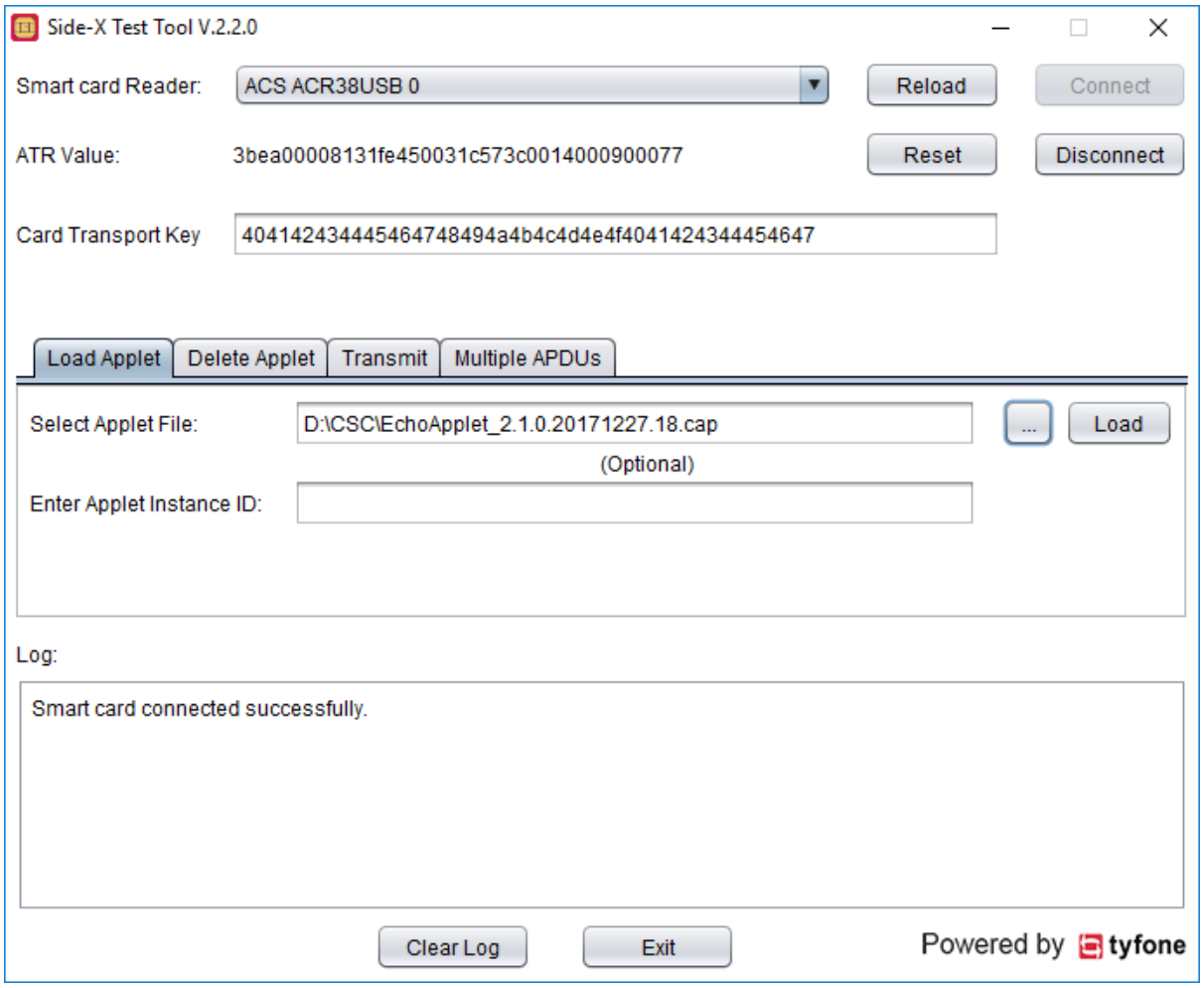
NOTE: The 'Card Transport Key' field displays the default key value which is required to authenticate with SideCard.



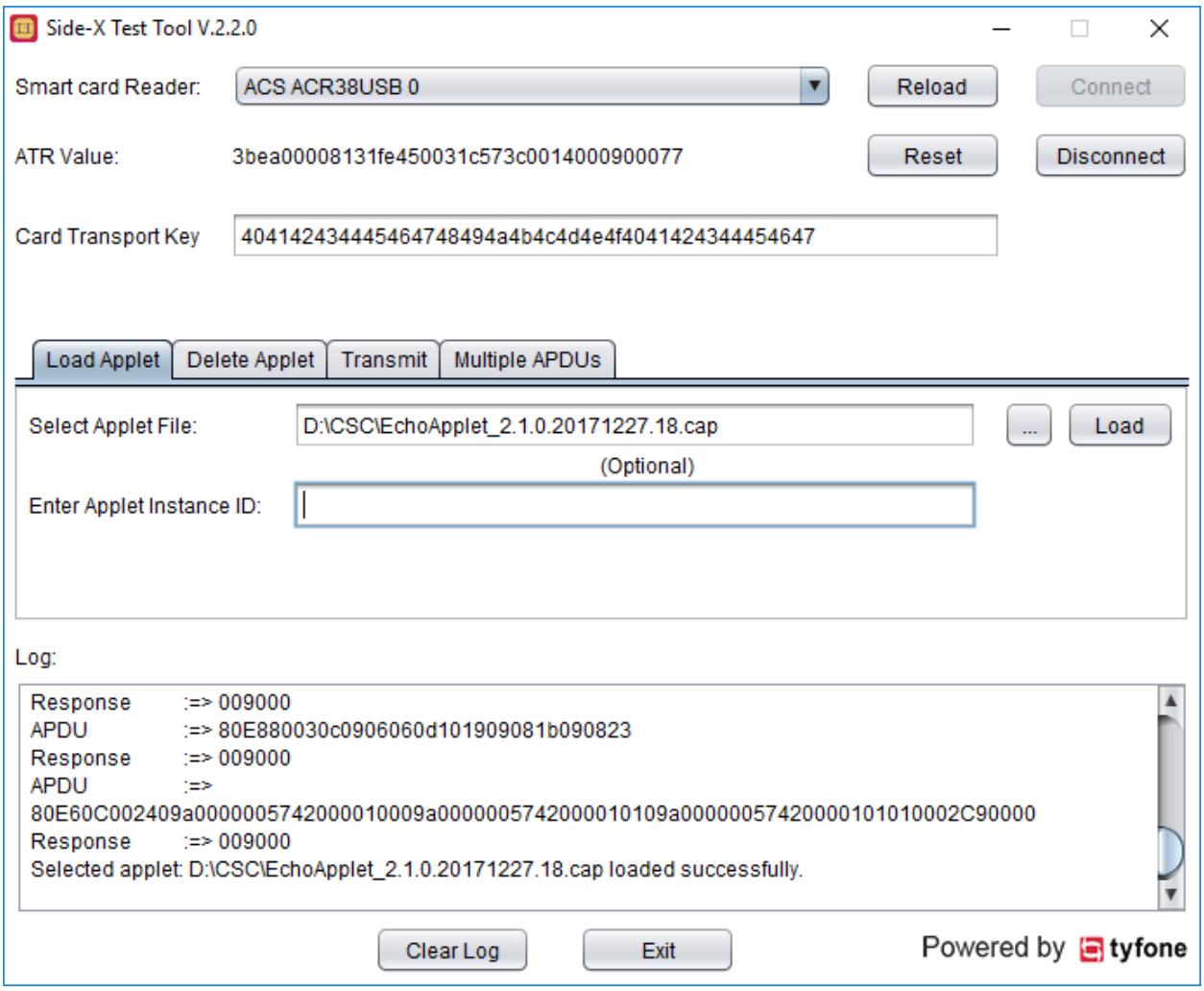
4. If the smart card reader name is not loaded, please re-connect Smart Card Reader to the system's USB port and select 'Reload' button.
5. Choose the smart card reader and select 'Connect' button.
6. On successful connection, the SideCard's ATR (Answer To Reset) value is displayed.



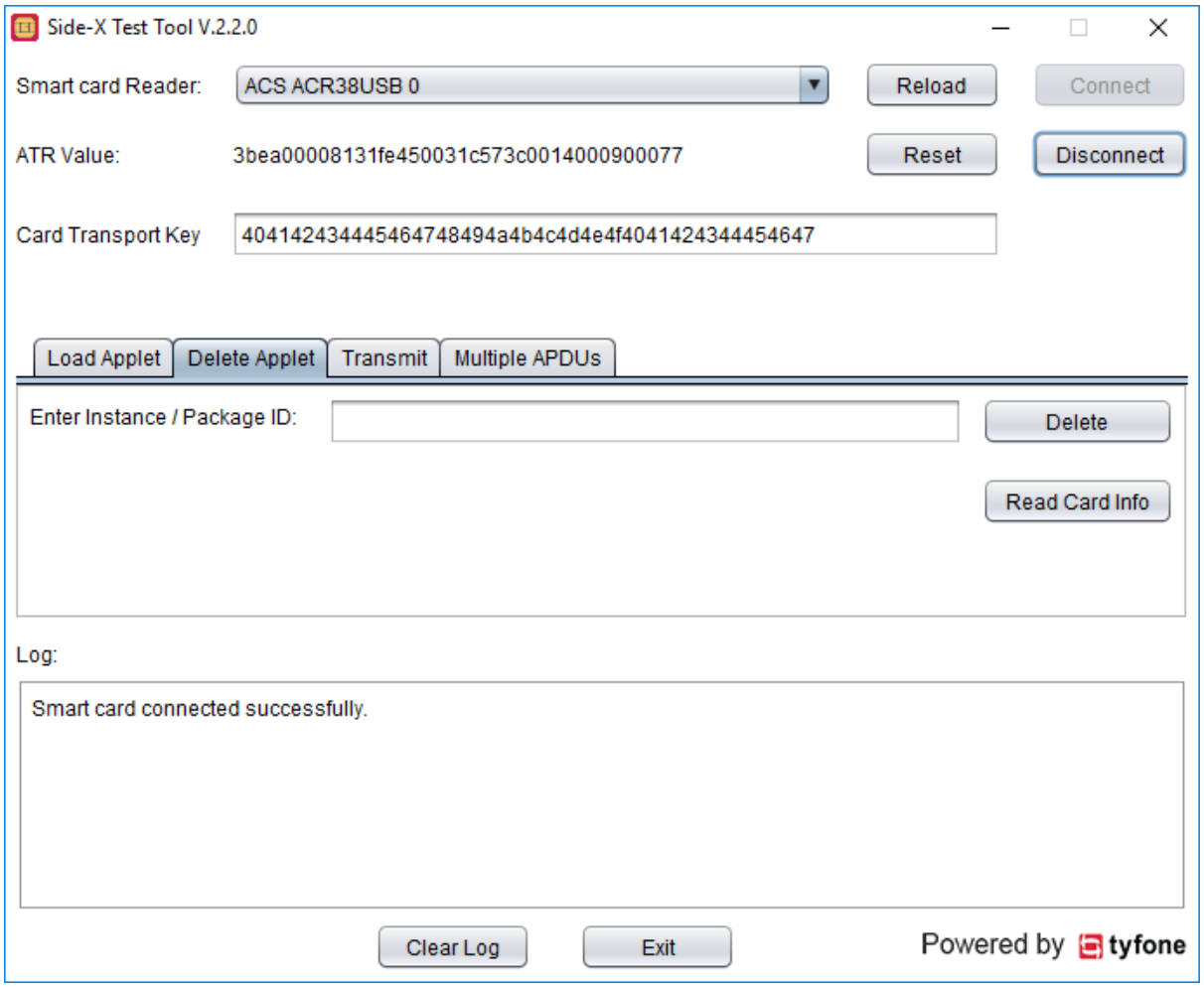
3.1 LOADING AN APPLET



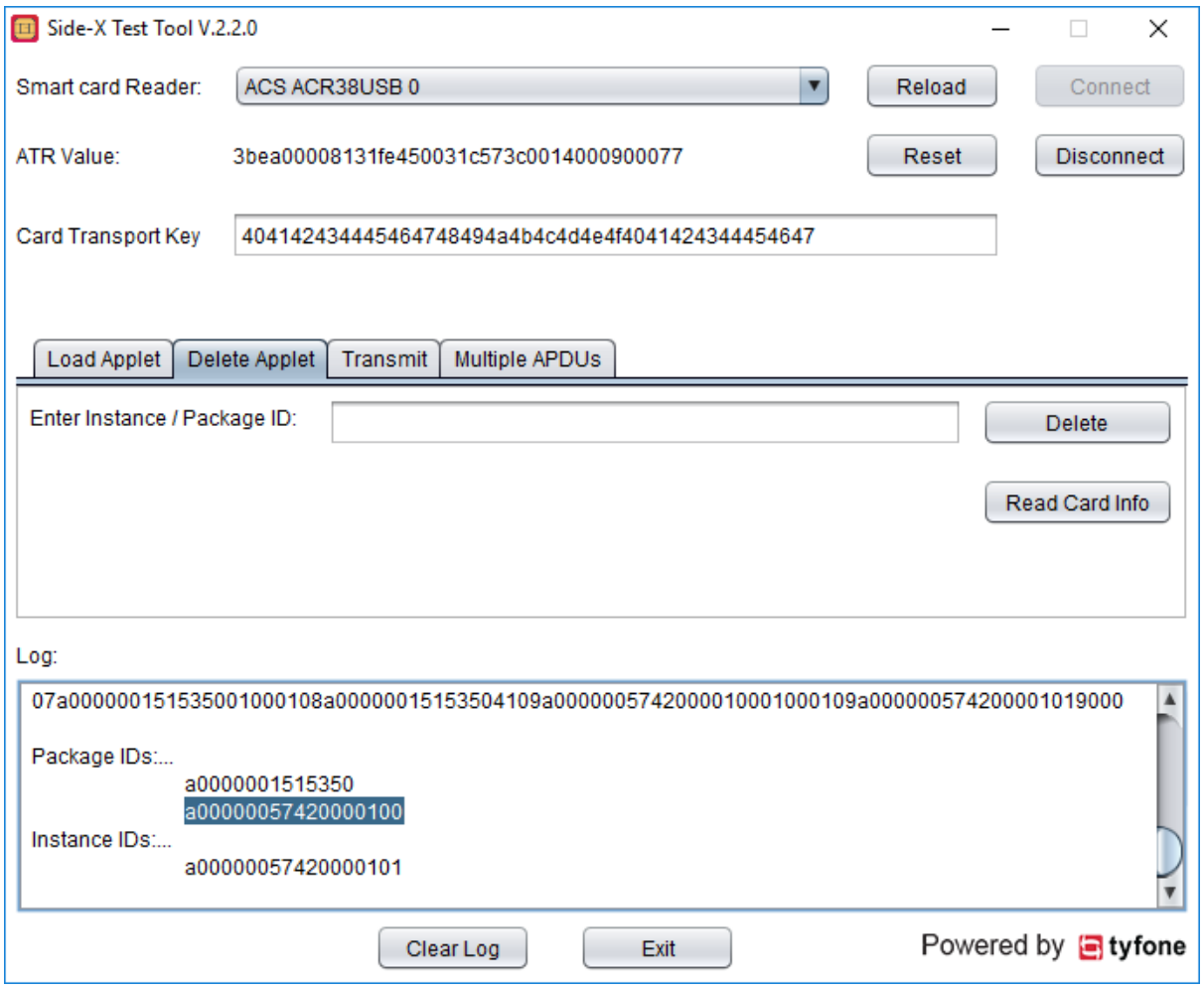
1. Select 'Load Applet' tab.
2. Choose the applet (*.cap) file by clicking on the '…' button. After selection of the applet file, optionally an instance ID for the applet can be provided.
3. Click on 'Load' button. If the operation is successful then 'Selected applet: xxx.cap loaded successfully.' message is shown in 'Log' section. Otherwise, an error message is displayed.

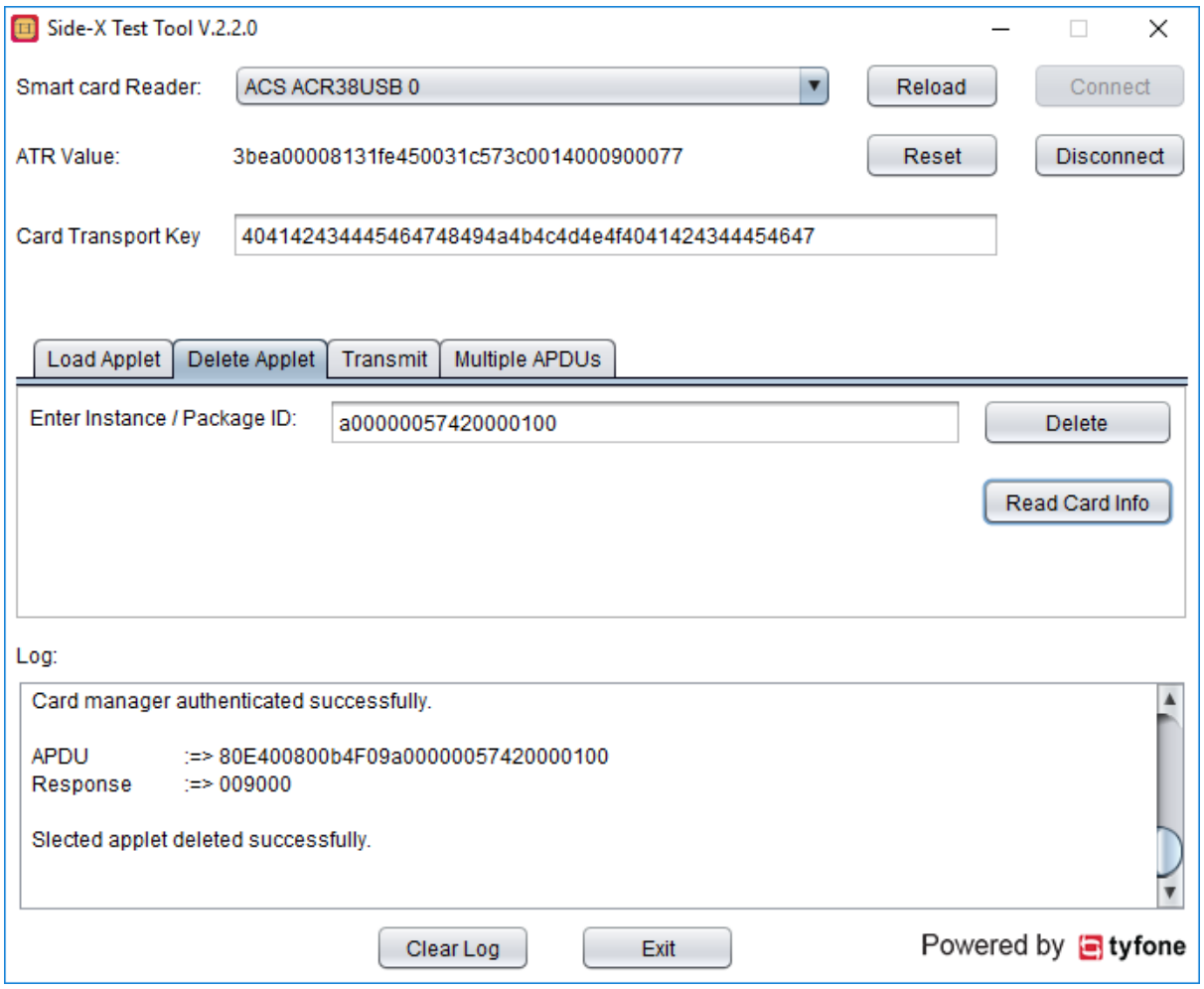


3.2 DELETING AN APPLET

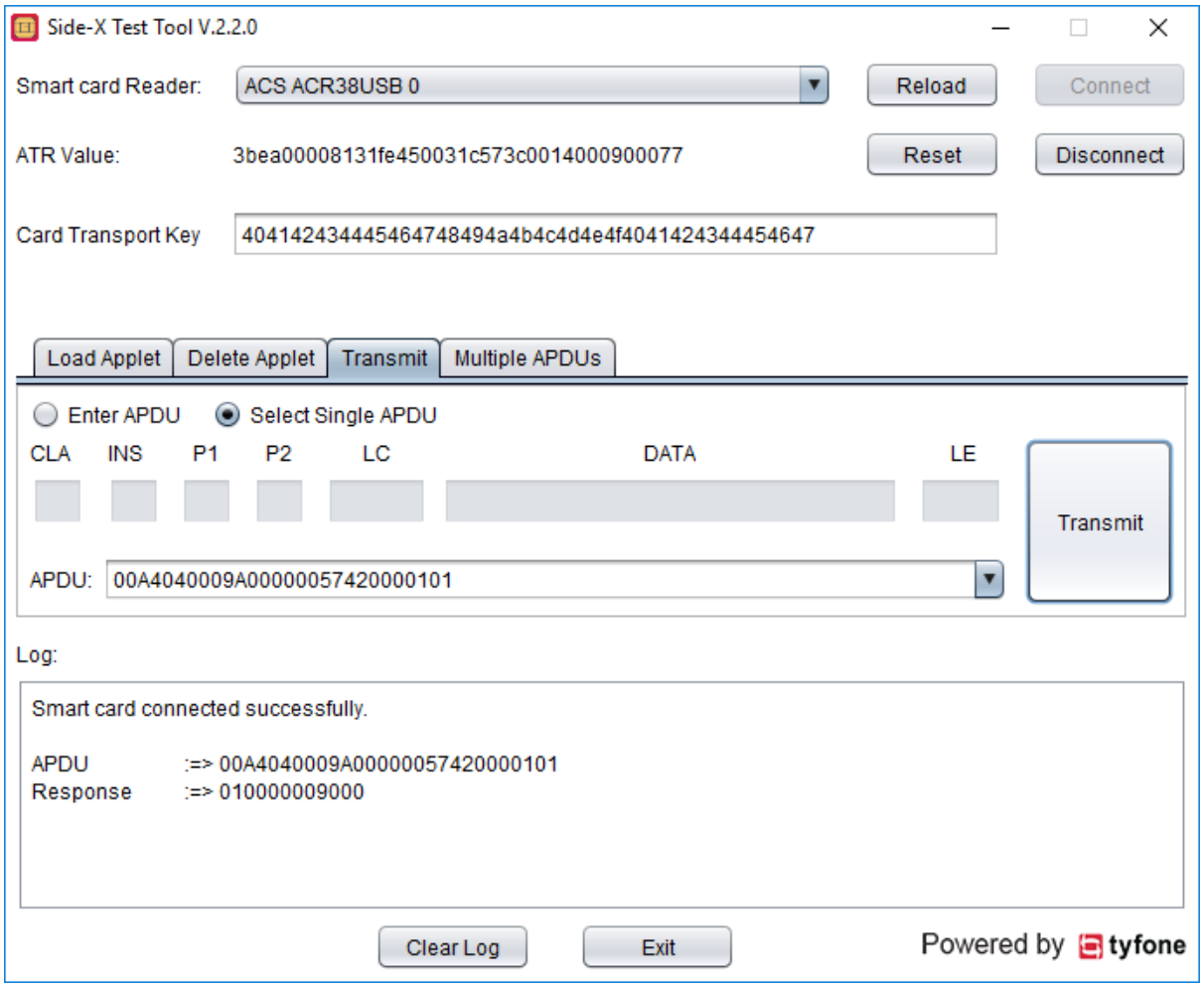


1. Select ‘Delete Applet’ tab.
2. Click on ‘Read Card Info’ button to display list of Package IDs and Applet AIDs available in the Smart Card.
3. Enter Instance ID or Package ID of the desired applet in the ‘Enter Instance/Package ID’ text field and click on ‘Delete’ button to delete the applet from the Smart Card.





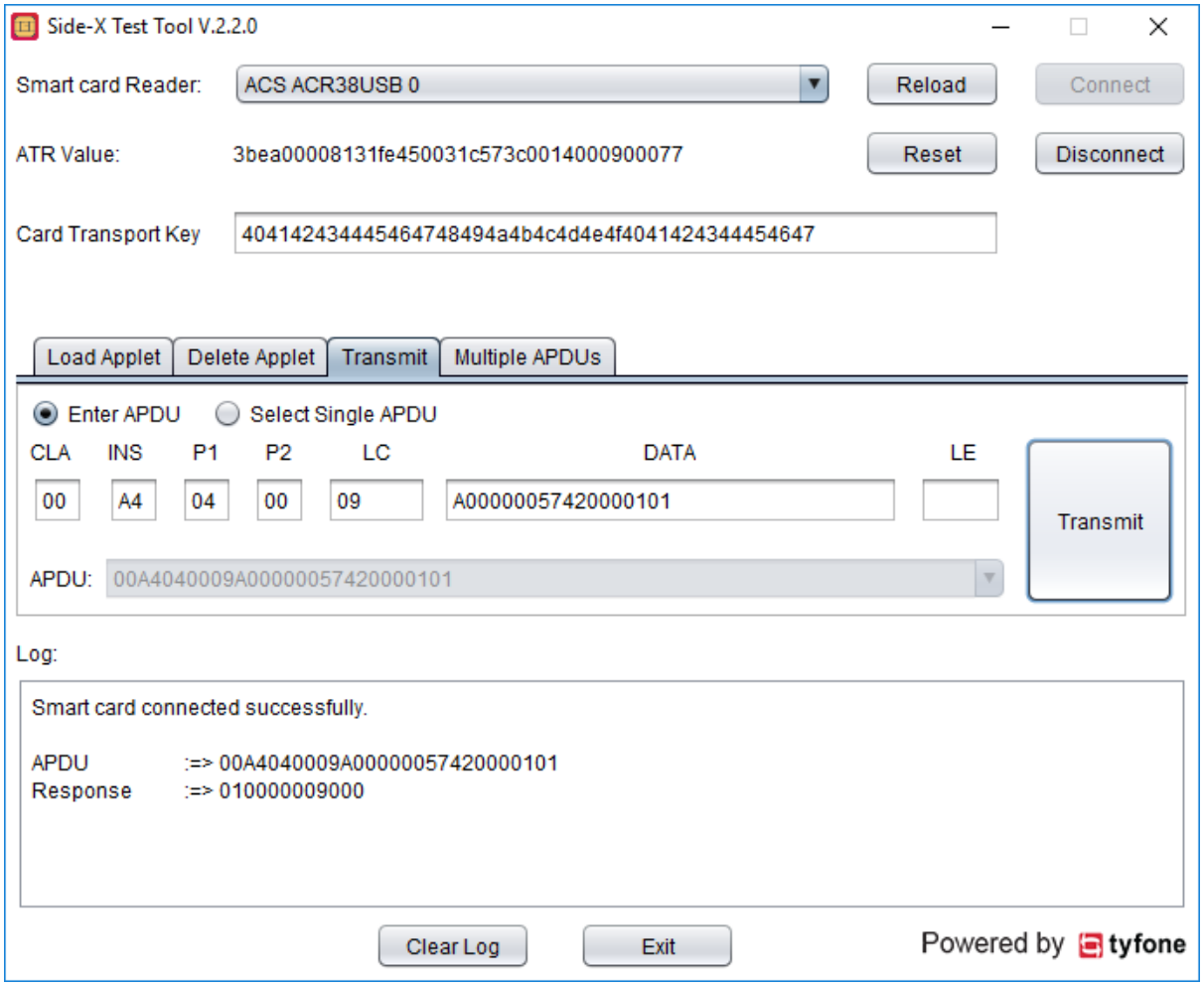
3.3 TRANSMIT SINGLE APDU



3.3.1 Using 'Select Single APDU' Option

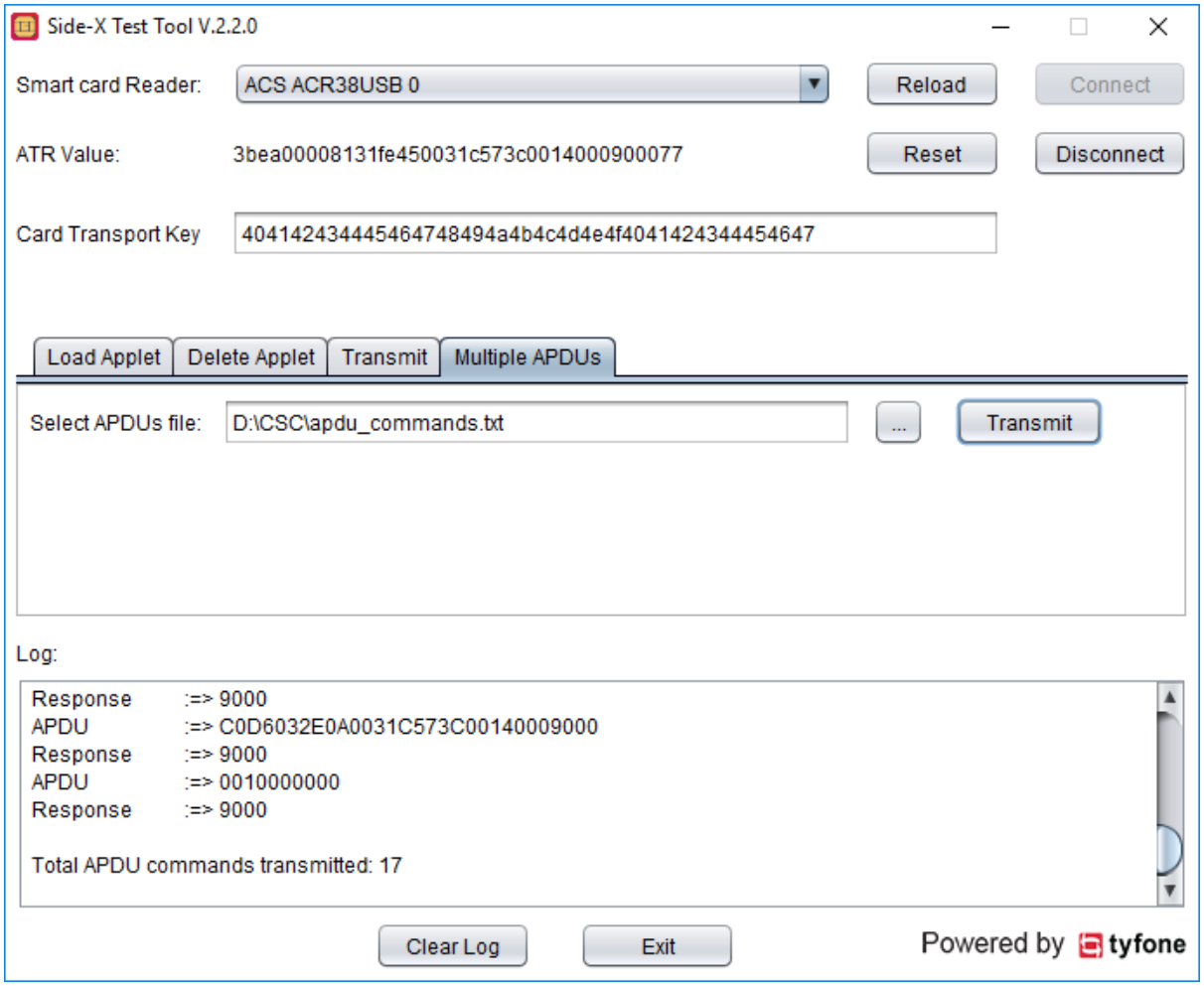
1. Select 'Transmit' tab.
2. Choose 'Select Single APDU' (default) radio button.
3. Select APDU from drop down list, or enter an APDU command in 'APDU' field.
4. Click on 'Transmit' button to send the APDU command to the smart card and read the response.
5. APDU command and its response data along with status bytes is displayed in the 'Log' section.

3.3.2 Using ‘Enter APDU’ Option

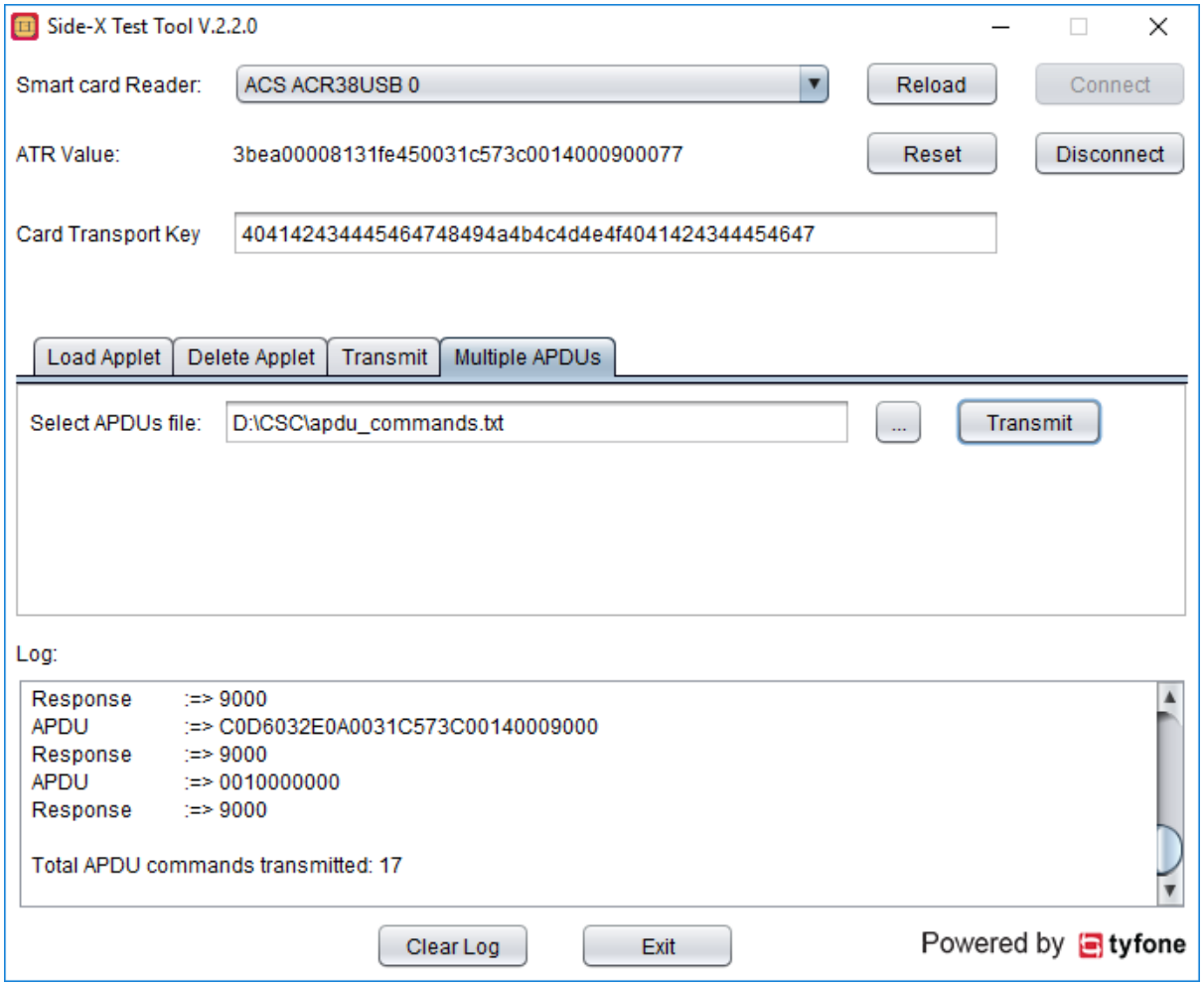


1. Select ‘Transmit’ tab.
2. Choose ‘Enter APDU’ radio button.
3. Enter APDU command in the APDU command parameter fields.
4. Click on ‘Transmit’ button to send the APDU command to smart card and read the response.
5. APDU command and its response data along with status bytes is displayed in the ‘Log’ section.

3.4 TRANSMIT MULTIPLE APDUS



1. Select 'Multiple APDUs' tab.
2. Select the multiple APDUs text file by clicking on '…' button.
   NOTE: In the text file, any lines that start with '//' or '#' will be ignored.
3. Click on 'Transmit' button. All the APDU commands listed in the text file are sent to the smart card and the corresponding responses are also read. The 'Log' section will display all the command/response messages along with a total count of APDU commands transmitted.



NOTE: Please refer to Applet Specification document to form the list of APDUs

4 SIDECARD COMMUNICATION APPLICATION DEVELOPMENT

We use PCSC Smart Card communication APIs to communicate with our SideCard.

4.1 PCSC Sample Code in JAVA (WINDOWS/LINUX/MAC OS)

```
import java.util.List;

import javax.smartcardio.*;


public class Blog {

 public static void main(String[] args) {

 try {

  // Display the list of terminals

  TerminalFactory factory = TerminalFactory.getDefault();

  List<CardTerminal> terminals = factory.terminals().list();

  System.out.println("Terminals: " + terminals);


  // Use the first terminal

  CardTerminal terminal = terminals.get(0);


  // Connect wit hthe card

  Card card = terminal.connect("*");

  System.out.println("card: " + card);

  CardChannel channel = card.getBasicChannel();


  // Send Select Applet command

  byte[] aid = {(byte)0xA0, 0x00, 0x00, 0x00, 0x62, 0x03, 0x01, 0x0C, 0x06, 0x01};

  ResponseAPDU answer = channel.transmit(new CommandAPDU(0x00, 0xA4, 0x04, 0x00, aid));

  System.out.println("answer: " + answer.toString());
```

```
  // Send test command
  answer = channel.transmit(new CommandAPDU(0x00, 0x00, 0x00, 0x00));
  System.out.println("answer: " + answer.toString());
  byte r[] = answer.getData();
  for (int i=0; i<r.length; i++)
   System.out.print((char)r[i]);
  System.out.println();

  // Disconnect the card
  card.disconnect(false);
 } catch(Exception e) {
  System.out.println("Ouch: " + e.toString());
 }
}
}
```

4.2 OUTPUT

```
Terminals: [PC/SC terminal Gemplus GemPC Twin 00 00]
card: PC/SC card in Gemplus GemPC Twin 00 00, protocol T=1, state OK
answer: ResponseAPDU: 2 bytes, SW=9000
answer: ResponseAPDU: 14 bytes, SW=9000
Hello world!
```