**Initiate Application Process**

Initiate Application Process is the next step after the Application Selection. It initiates the transaction process and provides to the card a Processing Options Data Object List (PDOL) which contains necessary data. The card returns the Application Interchange Profile (AIP) and the Application File Locator (AFL). The AIP shows which Application are supported by the card. The AFL list all files that are required for the Transaction.

**Script**

EMV.prototype.initApplProc = function() {

// Create PDOL

var data = this.getProcessingOptions(null);

print(data);

var tl = new TLVList(data, TLV.EMV);

assert(tl.length == 1);

var t = tl.index(0);

if (t.getTag() == EMV.RMTF1) { // Format 1 0x80

this.cardDE[EMV.AIP] = t.getValue().left(2);

this.cardDE[EMV.AFL] = t.getValue().bytes(2);

} else {

assert(t.getTag() == EMV.RMTF2); // Format 2 0x77

tl = new TLVList(t.getValue(), TLV.EMV);

assert(tl.length >= 2);

this.addCardDEFromList(tl);

}

}

**Get Processing Options**

/\*\*

\* Send GET PROCESSING OPTION APDU

\*

\*

\*/

EMV.prototype.getProcessingOptions = function(pdol) {

if (pdol == null) {

//var pdol = new ByteString("8300", HEX); // OTHER

var pdol = new ByteString("830B0000000000000000000000", HEX); // VISA

}

var data = this.card.sendApdu(0x80, 0xA8, 0x00, 0x00, pdol, 0);

return(data);

}

The Get Processin Options command initiates the transaction within the card. The card returns the AIP and AFL.  
The command has the following structure:  
  
Source: [EMV Book 3](http://www.emvco.com/specifications.aspx?id=155)

|  |  |
| --- | --- |
| **Code** | **Value** |
| CLA | '80' |
| INS | 'A8' |
| P1 | '00'; all other values are RFU |
| P2 | '00'; all other values are RFU |
| Lc | var. |
| Data | PDOL related data |
| Le | '00' |

**PDOL**

The data the terminal should send to the card is given in the PDOL. The PDOL is stored in the FCI of the ADF and has the tag '9F38'. The PDOL only contains the expected tagname and length.  
Here an example for VISA cards:

FCI:

6F [ APPLICATION 15 ] IMPLICIT SEQUENCE SIZE( 40 )

84 [ CONTEXT 4 ] SIZE( 7 )

0000 A0 00 00 00 03 10 10 .......

A5 [ CONTEXT 5 ] IMPLICIT SEQUENCE SIZE( 29 )

50 [ APPLICATION 16 ] SIZE( 4 )

0000 56 49 53 41 VISA

87 [ CONTEXT 7 ] SIZE( 1 )

0000 01 .

9F38 [ CONTEXT 56 ] SIZE( 12 )

0000 9F 33 03 9F 1A 02 9F 35 01 9F 40 05 .3.....5..@. // PDOL

5F2D [ APPLICATION 45 ] SIZE( 2 )

0000 64 65

This VISA-Card needs for the transaction the following data:

|  |  |  |
| --- | --- | --- |
| **Tag** | **Name** | **Length** |
| '9F33' | Terminal Capabilities | '03' |
| '9F1A' | Terminal Country Code | '02' |
| '9F35' | Terminal Type | '01' |
| '9F40' | Additional Terminal Capabilities | '05' |

The PDOL related bytestring has the tag '83', the length byte and the concatenated data from the list above without tag and length bytes as value. In this case we set the value to zero.

96 C: 80 A8 00 00 - UNKNOWN\_INS Lc=13

0005 83 0B 00 00 00 00 00 00 00 00 00 00 00 .............

Le=0

R: SW1/SW2=9000 (Normal processing: No error) Lr=36

0000 77 22 82 02 78 00 94 1C 10 01 04 00 10 05 05 00 w"..x...........

0010 08 06 06 01 08 07 07 01 08 08 09 01 08 0A 0A 00 ................

0020 08 01 04 00

The initApplProc function extract now the AIP and AFL information of the response and store them in the cardDE array. The response can have two different formats with ever tag '80' or tag '77'. In this case the data in the response have the tag '77' and will be processed by the function addCardDEFromList which stores the AIP and AFL in the cardDE array.

EMV.prototype.addCardDEFromList = function(tlvlist) {

for (var i = 0; i < tlvlist.length; i++) {

var t = tlvlist.index(i);

print(t.getTag().toString(16) + " - " + t.getValue());

this.cardDE[t.getTag()] = t.getValue();

}

}