Application Protocol Data Unit (APDU)

As previously indicated, the next phase of a card session is the Execution of the Transaction(s). The specific operations performed during a transaction depend on the type of card and account (credit, debit, etc.) and the user's request. Regardless of the specific operations, the transactions are accomplished by issuing commands from the terminal to the smart card. The smart card performs the requested operation(s) and potentially communicates a result. The card's operation can be as simple as reading a location in memory or as complex as performing a cryptographic operation. Regardless of the operation, the communication between the terminal and the card is conducted using Application Protocol Data Units, or APDUs.

To run an application, the smart card and terminal must exchange information. This sharing of information is accomplished in a command-response data exchange. The terminal creates and sends a command to the smart card, which then interprets the command and sends a response. This command-response message pair is known as an Application Protocol Data Unit (APDU). A specific command message sent by the terminal (C-APDU) will have a specific response message from the card (R-APDU). These messages are referred to as APDU command-response pairs. The EMV specification details the format of both of these message types, and their formats are described below.

**C-APDU Format**

The terminal initiates all Command APDUs. They consist of a required 4-byte header followed by an optional body of variable length that can contain data. The number of data bytes contained in the C-APDU is specified in the command byte Lc; the number of bytes which the terminal expects to receive from the card's response is specified in the command byte Le. Table 4 shows the C-APDU format, and the characters are described in Table 5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 4. Command APDU Structure** | | | | | | |
| **CLA** | **INS** | **P1** | **P2** | **Lc** | **Data** | **Le** |
| ←Mandatory Header→ | | | | ←Conditional Body→ | | |

|  |  |  |
| --- | --- | --- |
| **Table 5. Command APDU Content Description** | | |
| **Code** | **Description** | **Length** |
| CLA | Class of instruction | 1 |
| INS | Instruction code | 1 |
| P1 | Instruction parameter 1 | 1 |
| P2 | Instruction parameter 2 | 1 |
| Lc | Number of bytes present in command date field | 0 or 1 |
| Data | String of data bytes sent in command (= Lc) | Variable |
| Le | Maximum number of data bytes expected in data field of response | 0 or 1 |

The first byte of the Command APDU is defined as the instruction class, and is called CLA. This byte can take any 8-bit value except 0xFF. At present, however, only the values of the most significant nibble of 0 and 8 are used. The most significant nibble with a value of 0 is defined as an interindustry command, and the value of 8 is proprietary to the EMV specification.

The second byte of the Command APDU is the instruction code, and is called INS. This byte is valid only if the least significant bit is 0 and the most significant nibble is neither a 6 nor a 9.

The P1 and P2 bytes of the mandatory header contain parameters for the specific command, and can be any value. If not used, the parameter byte must have the value of 0x00.

**R-APDU**

After receiving and interpreting the APDU command from the terminal, the smart card will return a response. As defined in the specifications, this response consists of an optional body of variable length followed by a required trailer consisting of two bytes. This format is illustrated in Table 6, and the contents of the APDU response are described in Table 7.

|  |  |  |
| --- | --- | --- |
| **Table 6. APDU Card Response Format** | | |
| **Data** | **SW1** | **SW2** |
| ←Body→ | ←Trailer→ | |

|  |  |  |
| --- | --- | --- |
| **Table 7. APDU Command Response Contentsp** | | |
| **Code** | **Description** | **Length** |
| Data | String of data bytes received in APDU response | Var (= Lr) |
| SW1 | Command processing status | 1 |
| SW2 | Command processing qualifier | 1 |

The expected length of the card's response is transmitted as the Le code portion of the APDU command. The actual length of the response is called Lr. Although the card does not transmit the value of Lr, the terminal can calculate it if needed for the application.

For normal completion of a command, a smart card will return SW1 with a value of 0x90 and SW2 with a value of 0x00. Any other response indicates that either an error or warning occurred².