**Data Structure**

**ASN.1 - Abstract Syntax Notation One**

ASN.1 is a language to describe data structure.  
There are different data types: primitiv (boolean, integer, UTF8 string) and constructed (sequence).  
  
e.g.

Sequence

{

Name::= UTF8 String

Age::= Integer

Vegetarian::= Boolean OPTIONAL

Smoker::= [0] Boolean OPTIONAL

}

Sequence

{

Name::= "Müller"

Age::= "30"

Vegetarian::= "false"

Smoker::= "false"

}

**TLV - [T]ag [L]ength [V]alue**

To encode the ASN.1 data we use the TLV Structure. Every data object consists of a tag, a length byte and the value/data.  
The tag defines if the object is an integer, boolean or something else.

Tag

30 : Sequence

OC : UTF8 String

02 : Integer

01 : Boolean

Here we have the ASN.1 example encodet to the TLV structure:

30 11 OC 06 4D 7E 6C 6C 65 72 02 01 1E 01 01 00 80 01 00

Legend/Explanation

Tag Length Value

30 11 Sequence with a length of 17

OC 06 4D 7E 6C 6C 65 72 UTF8 String with a length of 6 and the value Müller

02 01 1E Integer with a length of 1 and the value 30

01 01 00 Optional Boolean with a length of 1 and the value false

80 01 00 Optional Boolean with a length of 1 and the value false

**Coding of the Tag**

| **b8** | **b7** | **b6** | **b5** | **b4** | **b3** | **b2** | **b1** | **explanation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 |  |  |  |  |  |  | universal class |
| 0 | 1 |  |  |  |  |  |  | application class |
| 1 | 0 |  |  |  |  |  |  | context specific class |
| 1 | 1 |  |  |  |  |  |  | private class |
|  |  | 0 |  |  |  |  |  | primitiv data object |
|  |  | 1 |  |  |  |  |  | constructed data object |
|  |  |  | x | x | x | x | x | tag value |
|  |  |  | 1 | 1 | 1 | 1 | 1 | there is a 2nd byte with tag value |

Optional Byte 2:

| **b8** | **b7** | **b6** | **b5** | **b4** | **b3** | **b2** | **b1** | **explanation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x |  |  |  |  |  |  |  | b8 = 0 : this is the last byte |
|  | x | x | x | x | x | x | x | tag value |

Tags are divided into four classes. b8 and b7 are defining the class.  
Universal classes are basic data types like integer, boolean, etc.  
The application class belongs to an application or norm. Context specific and private are not in the norm and created for special cases. For example the second optional boolean "smoker" is a context specific class so that you don't mix up the two booleans "vegetarian" and "smoker".  
  
e.g.

The Sequence tag '30' decoded:

b8 b7 b6 b5 b4 b3 b2 b1

0 0 = universal class

1 = constructed

1 0 0 0 0 = tag value

0011 0000

'30'

---------------

The UTF8 String tag '0C' decoded:

b8 b7 b6 b5 b4 b3 b2 b1

0 0 = universal class

0 = primitiv

0 1 1 0 0 = tag value

0000 1100

'0C'

**Length byte**

Depending on the value we have one or more length bytes.  
The byte has the following structure:

|  |  |  |
| --- | --- | --- |
| **No of Bytes** | **Length** | **Coding** |
| 1 | 0-127 | 0xxxxxx |
| 2 | 128-255 | 10000001 xxxxxxxx |
| 3 | 256-65535 | 10000010 xxxxxxxx xxxxxxxx |

According to this structure you can have any more length bytes.

Here we have two examples which are output with the reademv script from a Master Card  
  
Example 1:

A2 C: 00 A4 04 00 - SELECT Lc=7

0005 A0 00 00 00 04 10 10 .......

Le=0

R: SW1/SW2=9000 (Normal processing: No error) Lr=28

0000 6F 1A 84 07 A0 00 00 00 04 10 10 A5 0F 50 0A 4D o............P.M

0010 61 73 74 65 72 43 61 72 64 87 01 01 asterCard...

// We will concentrate us on the Response Value:

6F 1A 84 07 A0 00 00 00 04 10 10 A5 0F 50 0A 4D 61 73 74 65 72 43 61 72 64 87 01 01

Tag: "6F"  
Application class  
Primitive data object  
  
Length: "1A"  
'1A' = binary 00011010 = decimal 26 . There are 26 value bytes  
  
Value:  
"84 07 A0 00 00 00 04 10 10 A5 0F 50 0A 4D 61 73 74 65 72 43 61 72 64 87 01 01"

Example 2:

A2 C: 00 B2 01 1C - READ RECORD Le=0

R: SW1/SW2=9000 (Normal processing: No error) Lr=195

0000 70 81 C0 8F 01 04 9F 32 01 03 92 24 9F FB FB 7F p......2...$....

0010 EE C7 B0 43 67 B3 E4 C6 71 C3 0B 4A EE AD A2 C1 ...Cg...q..J....

0020 93 49 58 DD 61 04 D1 50 EA FD 3C 05 2C 97 0E 8D .IX.a..P..<.,...

0030 90 81 90 52 D7 78 E3 33 2B 72 0F 4F E4 1D 7C 1B ...R.x.3+r.O..|.

0040 ED 06 45 EA 7D EF 14 E4 E3 6F 80 90 A1 42 B7 E1 ..E.}....o...B..

0050 B1 7D DE CF AA 80 FC B4 BF 04 1C 2D 44 04 AD 1E .}.........-D...

0060 7F 19 C9 56 5B 93 7F 5E B5 02 90 6F EE 32 F5 21 ...V[..^...o.2.!

0070 E5 32 AB FC 37 F0 46 1E 91 AA 46 79 8D 74 C4 BA .2..7.F...Fy.t..

0080 FA 08 81 A3 0E 1F 9B B8 7A B7 85 C7 E2 9A 45 46 ........z.....EF

0090 D1 B7 FD 6F 98 A4 65 19 FB 7F 53 20 3A 93 AA C9 ...o..e...S :...

00A0 5C 5B 53 B8 CC 6E 9A D3 DB C9 25 CC 72 B9 6E DD \[S..n....%.r.n.

00B0 78 3B B0 D7 B6 E8 E9 78 BB 35 5E 45 5E 7A 5B CA x;.....x.5^E^z[.

00C0 57 C4 95 W..

Tag: "70"  
Application class  
Constructed data object  
  
Length: "81 C0"  
1st byte '81' = binary 10000001 . There are two length bytes.  
2nd byte 'C0' = binary 11000000 = decimal 192. There are 192 value bytes.  
  
Value:  
