*Revision* : 1.10 – describes asn1c-0.9.6

# Part I

# ASN.1 Basics

# Chapter 1

# Abstract Syntax Notation: ASN.1

*This chapter defines some basic ASN.1 concepts and describes several most widely used types. It is by no means an authoritative or complete reference. For more complete ASN.1 description, please refer to Olivier Dubuisson's book [Dub00] or the ASN.1 body of standards itself [ITU-T/ASN.1].*

The Abstract Syntax Notation One is used to formally describe the semantics of data transmitted across the network. Two communicating parties may have different formats of their native data types (i.e. number of bits in the integer type), thus it is important to have a way to describe the data in a manner which is independent from the particular machine's representation. The ASN.1 specifications is used to achieve one or more of the following:

-

### 1.1.3 The ENUMERATED type

The ENUMERATED type is semantically equivalent to the INTEGER type with some integer values explicitly named.

```
FruitId ::= ENUMERATED { apple(1), orange(2) }

-- The numbers in braces are optional,
-- the enumeration can be performed
-- automatically by the compiler
ComputerOSType ::= ENUMERATED {
    FreeBSD,          -- will be 0
    Windows,          -- will be 1
    Solaris(5),       -- will remain 5
    Linux,            -- will be 6
    MacOS             -- will be 7
}
```

### 1.1.4 The OCTET STRING type

This type models the sequence of 8-bit bytes. This may be used to transmit some opaque data or data serialized by other types of encoders (i.e. video file, photo picture, etc).

### 1.1.5 The OBJECT IDENTIFIER type

The OBJECT IDENTIFIER is used to represent the unique identifier of any object, starting from the very root of the registration tree. If your organization needs to uniquely identify something (a router, a room, a person, a standard, or whatever), you are encouraged to get your own identification subtree at `http://www.iana.org/protocols/forms.htm`ComputerOIaquepernet.5(alues)-250(e)15(xplicrnet-iapple used)-36955 77(0.2aqu42 Tders)9.963

```
-- an array of structures defined in place.
ManyCircles ::= SEQUENCE OF SEQUENCE {
                            radius INTEGER
                            }
```

### 1.3.5 The SET OF type

# Part II

# Using the ASN.1 Compiler

# Chapter 2

# Introduction to the ASN.1 Compiler

The purpose of the ASN.1 compiler, of which this document is part, is to convert the

# Chapter 3

# Chapter 4

# Using the ASN.1 Compiler

## 4.1 Command-line options

of BER, so the generic BER parser is also capable of decoding the data encoded by CER and DER encoders. The opposite is not true.

### 4.3.2 Encoding DER

```
        }
    }
```

See Section 4.3.5 for the example of stdio-based XML encoder and other pretty-printing suggestions.

### 4.3.4 Validating the target structure

Sometimes the target structure needs to be validated. For example, if the structure was
created by the application (as opposed to being decoded from some externad05((epce(˙,s)]TJ 0 -11.965 Td[(som
the otherh(ane,)377[(the)-530(sccessfule)-521(decoeing)-521(fg)-521(the)-530idaag from some et0epcee oeas
ncee21(sarily.)-320meaon tat. the valiedIte the tat.
thespecificcationssomesubtypen(tat.)-09(weren)-310not.(akw)10eonaccount.
decodeitime,uld.tusefuleno.per(fome)-37(nhe.)-37(last.)-37(check.)-37(wheon)-37(nhe.)]TJ 0 -11.955 Td[(ldwag

˝(˙e)-880func(t)1tionor vriousn ey-
pliitIte˝ise(˙e)-306func(tion)2305(fteor)-cata_ch.
primetarget structu:onvakw pri_0(strume)-56membthe.pris(˝(˙e)-266func(tine,)834whi_che)-260isepris(˙(soutne,6100&

```
struct my_figure {       /* The custom structure */
    int flags;           /* <some custom member> */
    /* The type is generated by si2608er>IN(ome)-600(custom)-6300(cust -3 Tf -308
```

# Bibliography

[ASN1C] OpenSource ASN.1 Compiler. `http://lionet.info/`