

Using the OpenSource ASN.1 Compiler

Lev Walkin <vlm@lionet.info>

13th October 2004

Contents

I	ASN.1 Basics	5
1	Abstract Syntax Notation: ASN.1	7
1.1	Some of the ASN.1 Basic Types	8
1.1.1	The BOOLEAN type	8
1.1.2	The INTEGER type	
	1.1.3 The ENUMERATED type	

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 Duboisson's book [\[Dub00\]](#) or the ASN.1 body of standards itself [\[ITU-T/ASN.1\]](#)

1.3 ASN.1 Constructed Types

1.3.1 The SEQUENCE type

This is an ordered collection of other simple or constructed types. The SEQUENCE constructed type resembles the C "struct" statement.

```
Address ::= SEQUENCE {  
    -- The apartment number may be omitted  
    apartmentNumber    NumericString OPTIONAL,  
    streetName          PrintableString,  
    cityName            PrintableString,  
    stateName           PrintableString,  
    -- This one may be omitted too  
    zipNo               NumericString OPTIONAL  
}
```

1.3.2 The SET type

This is a collection of other simple or constructed types. Ordering is not important. The

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

1.3.5 The SET OF type

The SET OF type models the bag of structures. It resembles the SEQUENCE OF type, but the order is not important: i.e. the elements may arrive in the order which is not

Part II

ASN.1 Compiler

Chapter 2

Introduction to the ASN.1 Compiler

Chapter 3

Quick start

After building and installing the compiler, the *asn1c*¹

Chapter 4

Overall Options	Description
-E	Stop after the parsing stage and print the reconstructed ASN.1 specification code to the standard output.
-F	Used together with -E, instructs the compiler to stop after the ASN.1 syntax tree fixing stage and dump the reconstructed ASN.1 specification to the standard output.
-P	Dump the compiled output to the standard output instead of

`check_constraints` Check that the contents of the target structure are semantically valid and constrained to appropriate implicit or explicit subtype constraints. Please refer to Section 4.3.4 on page 26.

Each of the above function takes the type descriptor (*asn_DEF_...*) and the target structure (*rect*, in the above example). The target structure is typically created by the

4.3.2 Encoding DER

The Distinguished Encoding Rules is the *canonical* variant of BER encoding rules. The DER is best suited to encode the structures where all the lengths are known beforehand.

This is probably exactly how you want to encode: either[(v)25Ather[(v)25r[(v)2(BER)-247dencodingv
manucalfiall1(-up,l)-187(the)-34((t)1ar)187gete(structure)-34(containse)-34((the)-33(data:)-34(whiche)-34((ize)

SN.1 ypde787(asn_DEF_Reacat787fromy thewhiche
ishats


```
    }  
}
```

As you see, the DER encoder does not write into some sort of buffer or something. It just invokes the custom function (possible, multiple times) which would save the

