Using the Open Source ASN.1 Compiler

Lev Walkin <

Chapter 1

Introduction to the ASN.1 Compiler

-

Chapter 2

Using the ASN.1 Compiler

2.1 Invoking the ASN.1 helper code

First of all, you should to include one or more header files into your application. For our Rectangle module, including the Rectangle.h file is enough:

```
#include <Rectangle.h>
```

The header files defines the C structure corresponding to the ASN.1 definition of a rectangle and the declaration of the ASN.1 type descriptor, which is used as an argument to most of the functions provided by the.1 For example,s the code
which frees the Rectangle_t structure:

```
Rectangle_t *rect = ...;

asn_DEF_Rectangle-> 106le      =as5 Td[(which)600(,...;)]0=419 Td[(The)-2130(enou
```

xer_encoder

-

```
/* 1. Rectangle_t is defined within my_figure */
```

```
#include <stdio.h>
#include <sys/types.h>
#include <Rectangle.h>   /* Rectangle ASN.1 type  */

/*
 * This is a custom function which writes the
 * encoded output into some FILE stream.
 */
static int
write_out(const void *buffer, size_t size, void *app_key) {
    FILE *out_fp = app_key;
    size_t wrote;

    wrote = fwrite(buffer, 1, size, out_fp);

    return (wrote == size) ? 0 : -1;
}

int main(int ac, char **av) {
    Rectangle_t *rectangle; /* Type to encode        */
    asn_enc_rval_t ec;/* Encoder return value  */

    /* Allocate the Rectangle_t */
```

3.2 A ”Rectangle” Decoder

This example will help you to create a simple BER decoder of a simple ”Rectangle” type used throughout this document.

1.

Chapter 4

Constraint validation examples

This chapter shows how to define ASN.1 constraints and use the generated validation code.

4.1 Adding constraints into ”Rectangle” type

s;

Chapter 5

Abstract Syntax Notation: ASN.1

This chapter defines some basic ASN.1 concepts and describes several most widely used types. It is by no meacs an authoritative or complete reference. F(or)-333(mor)37(e)-334(complete)]TJ 0 -11.956 Td[(ASN.

5.1.3 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,          -- acquires value 0
    Windows,          -- acquires value 1
    Solaris(5),       -- remains 5
    Linux,            -- becomes 6
    MacOS             -- becomes 7
}
```

5.1.4 The OCTET STRING type

This type models the sequence of 8-bit bytes. This may be used to transmit some

5.3 ASN.1 Constructed Types

5.3.1 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
}
```

5.3.2 The SET type

This is a collection of other simple or constructed types. Ordering is not important. The data may arrive in the order which is different from the order of specification. Data is encoded in the order not necessarily corresponding to the order of specification.

5.3.3 The CHOICE type

This type is just a choice between the subtypes specified in it. The CHOICE type

Bibliography

[ASN1C] The Open Source ASN.1 Compiler. http://lionet.info/asn1c

[AONL] Online ASN.1 Compiler. http://lionet.info/asn1c/asn1c.cgi

[Dub00] Olivier Dubuisson — *ASN.1 Communication between heterogeneous systems* — Morgan Kaufmann Publishers, 2000. http://asn1.elibel.tm.fr/en/book/. ISBN:0-12-6333361-0.

[ITU-T/ASN.1] ITU-T Study Group 17 – Languages for Telecommunication Systems http://www.itu.int/ITU-T/studygroups/com17/languages/