Using the Open ASN.1 Compiler

Lev Walkin <vlm@6ionet.info>

29th September 2004

Revision : 1.12

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:

-

e5(xample,)-220(this)-213(data)-212(structure)-213(may)-212(be)-213(encoded)-213(a)1(ccording)-213(to)-213(
them which will be described later.

The complete specification must be wrapped in a module, a

```
UsageExampleModule1
    { iso org(3) dod(6) internet(1) private(4)
```

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

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.

4.3 Invoking the ASN.1 helper code from the application

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, is the code
which frees the Rectangle_t structure:

```
Rectangle_t *rect = ...;

asn_DEF_Rectangle->free_struct(&asn_DEF_Rectangle,
    rect, 0);
```

This code defines a *rect* pointer which points to the Rectangle_t structure which needs to be freed. The second line invokes the generic free_struct routine created specifically for this Rectangle_t structure. The *asn_DEF_Rectangle* is the type descriptor, which holds a collection of generic routines to deal with the Rectangle_t structure.

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.

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.
bytesomd the buffer.

is is notverient it be ˝liakd, the BER encodeRould cosumed the
100bytes andepdthsefbytss ns bts ns case of
sed proce(ding)-250irtghrt actually he rhecoue.

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 after a BER decoding or after

```
        }
    }
```

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 data into appropriate storage. The optional argument *app_key* is opaque for the DER encoder code and just used by *_write_stream()* as the pointer to the appropriate output stream to be used.

If the custom write function is not given (passed as 0), then the DER encoder will essentially do the same thing (i.e., encode the data) but no callbacks will be invoked (so the data goes nowhere). It may prove useful to determine the size of the structure's encoding before actually doing the encoding[3].

Please look into der_encoder.h for the precise definition of der_encode() and related types.

4.3.3 Encoding XER

The XER stands for XML Encoding Rules, where XML, in turn, is eXtensible Markup

```
struct my_figure {       /* The custom structure */
    int flags;           /* <some custom member> */
    /* The type is generated by the ASN.1 compiler */
    Rectangle_t rect;
    /* other members of the structure */
};
```

In this example, the application programmer defined a custom structure with one ASN.1-derived member (rect). This member is not a reference to the Rectangle_t, but an in-place inclusion of the Rectan str0(stTd[(58(I)-333(of)-333(thfreeing333(2er)-39(thnecessary)65(ang35(of)-333(thusual-391

Bibliography

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