NAME

asn1parse - ASN.1 parsing tool

SYNOPSIS

openssl asn1parse [**-inform PEM|DER**] [**-in filename**] [**-out filename**] [**-noout**] [**-offset number**] [**-length number**] [**-i**] [**-oid filename**] [**-dump**] [**-dlimit num**] [**-strparse offset**] [**-genstr string**] [**-genconf file**]

DESCRIPTION

The **asn1parse** command is a diagnostic utility that can parse ASN.1 structures. It can also be used to extract data from ASN.1 formatted data.

OPTIONS

-inform DER|PEM
: the input format. **DER** is binary format and **PEM** (the default) is base64 encoded.

-in filename
: the input file, default is standard input

-out filename
: output file to place the DER encoded data into. If this option is not present then no data will be output. This is most useful when combined with the **-strparse** option.

-noout
: don't output the parsed version of the input file.

-offset number
: starting offset to begin parsing, default is start of file.

-length number
: number of bytes to parse, default is until end of file.

-i indents the output according to the ''depth'' of the structures.

-oid filename
: a file containing additional OBJECT IDENTIFIERs (OIDs). The format of this file is described in the NOTES section below.

-dump
: dump unknown data in hex format.

-dlimit num
: like **-dump**, but only the first **num** bytes are output.

-strparse offset
: parse the contents octets of the ASN.1 object starting at **offset**. This option can be used multiple times to ''drill down'' into a nested structure.

-genstr string, **-genconf file**
: generate encoded data based on **string**, **file** or both using *ASN1_generate_nconf(3)* format. If **file** only is present then the string is obtained from the default section using the name **asn1**. The encoded data is passed through the ASN1 parser and printed out as though it came from a file, the contents can thus be examined and written to a file using the **out** option.

OUTPUT

The output will typically contain lines like this:

```
 0:d=0 hl=4 l= 681 cons: SEQUENCE
```

.....

```
 229:d=3 hl=3 l= 141 prim: BIT STRING
 373:d=2 hl=3 l= 162 cons: cont [ 3 ]
 376:d=3 hl=3 l= 159 cons: SEQUENCE
 379:d=4 hl=2 l= 29 cons: SEQUENCE
 381:d=5 hl=2 l= 3 prim: OBJECT :X509v3 Subject Key Identifier
 386:d=5 hl=2 l= 22 prim: OCTET STRING
 410:d=4 hl=2 l= 112 cons: SEQUENCE
 412:d=5 hl=2 l= 3 prim: OBJECT :X509v3 Authority Key Identifier
 417:d=5 hl=2 l= 105 prim: OCTET STRING
 524:d=4 hl=2 l= 12 cons: SEQUENCE
```

.....

This example is part of a self signed certificate. Each line starts with the offset in decimal. **d=XX** specifies the current depth. The depth is increased within the scope of any SET or SEQUENCE. **hl=XX** gives the header length (tag and length octets) of the current type. **l=XX** gives the length of the contents octets.

The **-i** option can be used to make the output more readable.

Some knowledge of the ASN.1 structure is needed to interpret the output.

In this example the BIT STRING at offset 229 is the certificate public key. The contents octets of this will contain the public key information. This can be examined using the option **-strparse 229** to yield:

```
 0:d=0 hl=3 l= 137 cons: SEQUENCE
 3:d=1 hl=3 l= 129 prim: INTEGER :E5D21E1F5C8D208EA7A2166C7FAF9F6BDF2059669C60876DDB70840F1A5
 135:d=1 hl=2 l= 3 prim: INTEGER :010001
```

NOTES

If an OID is not part of OpenSSL's internal table it will be represented in numerical form (for example 1.2.3.4). The file passed to the **-oid** option allows additional OIDs to be included. Each line consists of three columns, the first column is the OID in numerical format and should be followed by white space. The second column is the ''short name'' which is a single word followed by white space. The final column is the rest of the line and is the ''long name''. **asn1parse** displays the long name. Example:

```
1.2.3.4 shortName A long name
```

EXAMPLES

Parse a file:

```
 openssl asn1parse -in file.pem
```

Parse a DER file:

```
 openssl asn1parse -inform DER -in file.der
```

Generate a simple UTF8String:

```
 openssl asn1parse -genstr 'UTF8:Hello World'
```

Generate and write out a UTF8String, don't print parsed output:

```
 openssl asn1parse -genstr 'UTF8:Hello World' -noout -out utf8.der
```

Generate using a config file:

```
 openssl asn1parse -genconf asn1.cnf -noout -out asn1.der
```

Example config file:

```
 asn1=SEQUENCE:seq_sect

 [seq_sect]
```

```
field1=BOOL:TRUE
field2=EXP:0, UTF8:some random string
```

BUGS

There should be options to change the format of output lines. The output of some ASN.1 types is not well handled (if at all).

SEE ALSO

ASN1_generate_nconf(3)