NAME

    gensnmptree - generate C and header files from a MIB description file

SYNOPSIS

    gensnmptree [**-dEeFfhlt**] [**-I** *directory*] [**-i** *infile*] [**-p** *prefix*] [*name ...*]

DESCRIPTION

    The **gensnmptree** utility is used to either generate C language tables and header files from a MIB description or to numeric OIDs from MIB descriptions. The first form is used only for maintaining the bsnmpd(1) daemon or for module writers. The second form may be used by SNMP client program writers.

    If none of the options **-e**, **-E** or **-t** are used **gensnmptree** reads a MIB description from its standard input and creates two files: a C-file *prefix*tree.c containing a table used by bsnmpd(1) during PDU processing and a header file *prefix*tree.h containing appropriate declarations of the callback functions used in this table, the table itself and definitions for all enums.

    The following options are available:

    -d

      Switch on debugging.

    -E

      Extract enumerations and bit constructs. In this mode the tool emits a header file that contains for each type given on the command line a C-enum definition and a preprocessor define that may be used to map values to strings.

    -e

      gensnmptree expects MIB variable names (only the last component) on its command line. It reads a MIB specification from standard input and for each MIB variable name emits three C preprocessor defines on its standard output:

    OIDX_name   This define can be used to initialize a *struct asn_oid* in the following way:

                    const struct asn_oid oid_sysDescr = OIDX_sysDescr;

    OIDLEN_name   is the length of the OID.

    OID_name   is the last component of the OID.

-F

emit definitions for C-functions includeable in a C-file that do some basic stuff on enums like value checking and conversion between value and strings.

-f

emit definitions for inline C-functions that do some basic stuff on enums like value checking and conversion between value and strings.

-h

Print a short help page.

-I *directory*

Add the named directory to the include path just before the standard include directories.

-i *infile*

Read from the named file instead of standard input.

-l Generate local preprocessor includes. This is used for bootstrapping bsnmpd(1).

-t Instead of normal output print the resulting tree.

-p *prefix*

Prefix the file names and the table name with *prefix*.

MIBS

The syntax of the MIB description file can formally be specified as follows:

file := top | top file

top := tree | typedef | include

tree := head elements ’)’

entry := head ’:’ index STRING elements ’)’

leaf := head type STRING ACCESS ’)’

column := head type ACCESS ’)’

type := BASETYPE | BASETYPE ’|’ subtype | enum | bits

subtype := STRING

enum := ENUM '(' value ')'

bits := BITS '(' value ')'

value := INT STRING | INT STRING value

head := '(' INT STRING

elements := EMPTY | elements element

element := tree | leaf | column

index := type | index type

typedef := 'typedef' STRING type

include := 'include' filespec

filespec := '"' STRING '"' | '<' STRING '>'

BASETYPE specifies a SNMP data type and may be one of
- NULL
- INTEGER
- INTEGER32 (same as INTEGER)
- UNSIGNED32 (same as GAUGE)
- OCTETSTRING
- IPADDRESS
- OID
- TIMETICKS
- COUNTER
- GAUGE
- COUNTER64

ACCESS specifies the accessibility of the MIB variable (which operation can be performed) and is one of
- GET
- SET

INT is a decimal integer and *STRING* is any string starting with a letter or underscore and consisting of letters, digits, underscores and minuses, that is not one of the keywords.

The *typedef* directive associates a type with a single name.

The *include* directive is replaced by the contents of the named file.

EXAMPLES
The following MIB description describes the system group:

```
include "tc.def"

typedef AdminStatus ENUM (
        1 up
        2 down
)

(1 internet
 (2 mgmt
  (1 mib-2
   (1 system
     (1 sysDescr OCTETSTRING op_system_group GET)
     (2 sysObjectId OID op_system_group GET)
     (3 sysUpTime TIMETICKS op_system_group GET)
     (4 sysContact OCTETSTRING op_system_group GET SET)
     (5 sysName OCTETSTRING op_system_group GET SET)
     (6 sysLocation OCTETSTRING op_system_group GET SET)
     (7 sysServices INTEGER op_system_group GET)
     (8 sysORLastChange TIMETICKS op_system_group GET)
     (9 sysORTable
      (1 sysOREntry : INTEGER op_or_table
        (1 sysORIndex INTEGER)
        (2 sysORID OID GET)
        (3 sysORDescr OCTETSTRING GET)
        (4 sysORUpTime TIMETICKS GET)
     ))
   )
  )
 )
)
```

SEE ALSO

bsnmpd(1)

AUTHORS

Hartmut Brandt <harti@FreeBSD.org>