NAME

bsnmpd - simple and extensible SNMP daemon

SYNOPSIS

bsnmpd [**-dh**] [**-c** *file*] [**-D** *options*] [**-e** *file*] [**-I** *paths*] [**-l** *prefix*] [**-m** *variable*[=*value*]] [**-p** *file*]

DESCRIPTION

The **bsnmpd** daemon serves the internet SNMP (Simple Network Management Protocol). It is intended to serve only the absolute basic MIBs and implement all other MIBs through loadable modules. In this way the **bsnmpd** can be used in unexpected ways.

The options are as follows:

-d Do not daemonize. Used for debugging.

-h Print a short usage message.

-c *file* Use *file* as configuration file instead of the standard one.

-D *options* Debugging options are specified with a **-o** flag followed by a comma separated string of options. The following options are available.

dump Dump all sent and received PDUs to the terminal.

events Set the debugging level of the event library (see eventlib(3)) to 10.

trace=level Set the snmp library trace flag to the specified value. The value can be specified in the usual C-syntax for numbers.

-e *file* Specify an alternate file where the agent’s engine id and number of boots are saved.

-I *paths* Specify a colon separated list of directories to search for configuration include files. The default is */etc:/usr/etc/:/usr/local/etc*. These paths are only searched for include specified within <> parentheses.

-l *prefix* Use *prefix* as the default basename for the pid and the configuration files.

-m *variable*[=*value*]
Define a configuration variable.

-p *file* Specify an alternate pid file instead of the default one.

CONFIGURATION

bsnmpd reads its configuration from either the default or the user specified configuration file. The configuration file consists of the following types of lines:

- variable assignments
- section separators
- include directives
- MIB variable assignments

If a line is too long it can be continued on the next line by ending it with a backslash. Empty lines and lines in which the first non-blank character is a "#" sign are ignored.

All MIB variable assignments of the entire configuration (including nested configuration files) are handled as one transaction, i.e., as if they arrived in a single SET PDU. Any failure during the initial configuration read causes **bsnmpd** to exit. A failure during the configuration read caused by a module load causes the loading of the module to fail.

The configuration is read during initialization of **bsnmpd**, when a module is loaded and when **bsnmpd** receives a SIGHUP.

VARIABLE ASSIGNMENTS

Variable assignments can take one of two forms:

```
variable := string
variable ?= string
```

The string reaches from the first non-blank character after the equal sign until the first new line or "#" character. In the first case the string is assigned to the variable unconditionally, in the second case the variable is only assigned if it does not exist yet.

Variable names must begin with a letter or underscore and contain only letters, digits or underscores.

SECTION SEPARATORS

The configuration consists of named sections. The MIB variable assignments in the section named "snmpd" are executed only during initial setup or when **bsnmpd** receives a SIGHUP. All other sections are executed when either a module with the same name as the section is loaded or **bsnmpd** receives a SIGHUP and that module is already loaded. The default section at the start of the configuration is "snmpd". One can switch to another section with the syntax

```
%secname
```

Where *secname* is the name of the section. The same *secname* can be used in more than one place in the configuration. All of these parts are collected into one section.

INCLUDE DIRECTIVES

Another configuration file can be included into the current one with the include directive that takes one of two forms:

```
.include "file"
.include <"file">
```

The first form causes the file to be searched in the current directory, the second form causes the file to be searched in the directories specified in the system include path. Nesting depth is only restricted by available memory.

MIB VARIABLE ASSIGNMENTS

A MIB variable is assigned with the syntax

```
oid [ suboids ] = value
```

oid is the name of the variable to be set. Only the last component of the entire name is used here. If the variable is a scalar, the index (.0) is automatically appended and need not to be specified. If the variable is a table column, the index (*suboids*) must be specified. The index consist of elements each separated from the previous one by a dot. Elements may be either numbers, strings or hostnames enclosed in [] brackets. If the element is a number it is appended to the current oid. If the element is a string, its length and the ASCII code of each of its characters are appended to the current oid. If the element is a hostname, the IP address of the host is looked up and the four elements of the IP address are appended to the oid.

For example, an oid of

```
myvariable.27.foooll.[localhost]."&^!"
```

results in the oid

```
myvariable.27.102.111.111.111.108.108.127.0.0.1.38.94.33
```

The value of the assignment may be either empty, a string or a number. If a string starts with a letter or an underscore and consists only of letters, digits, underscores and minus signs, it can be written without quotes. In all other cases the string must be enclosed in double quotes.

SUBSTITUTIONS

A variable substitution is written as

```
$(variable)
```

where *variable* is the name of the variable to substitute. Using an undefined variable is considered an error.

FILES

/etc/<prefix>.config	Default configuration file, where the default <prefix> is "snmpd".
/var/<prefix>.engine	Default engine id file.
/var/run/<prefix>.pid	Default pid file.
/etc:/usr/etc/:/usr/local/etc	Default search path for system include files.
/usr/share/snmp/mibs/FOKUS-MIB.txt */usr/share/snmp/mibs/BEGEMOT-MIB.txt* */usr/share/snmp/mibs/BEGEMOT-SNMPD.txt*	Definitions for the MIBs implemented in the daemon.
/etc/hosts.allow, */etc/hosts.deny*	Access controls that should be enforced by TCP wrappers are defined here. Further details are described in hosts_access(5).

SEE ALSO

gensnmptree(1), hosts_access(5)

STANDARDS

The **bsnmpd** conforms to the applicable IETF RFCs.

AUTHORS

Hartmut Brandt <harti@FreeBSD.org>

BUGS

Sure.