

NAME

snmp_config - handling of Net-SNMP configuration files

DESCRIPTION

The Net-SNMP package uses various configuration files to configure its applications. This manual page merely describes the overall nature of them, so that the other manual pages don't have to.

DIRECTORIES SEARCHED

First off, there are numerous places that configuration files can be found and read from. By default, the applications look for configuration files in the following 4 directories, in order: /etc/snmp, /usr/share/snmp, /usr/lib/x86_64-linux-gnu/snmp, and \$HOME/.snmp. In each of these directories, it looks for files snmp.conf, snmpd.conf and/or snmptrapd.conf, as well as snmp.local.conf, snmpd.local.conf and/or snmptrapd.local.conf. *.local.conf are always read last. In this manner, there are 8 default places a configuration file can exist for any given configuration file type.

Additionally, the above default search path can be overridden by setting the environment variable SNMP_CONFPATH to a colon-separated list of directories to search for. The path for the persistent data should be included when running applications that use persistent storage, such as snmpd.

Applications will read persistent configuration files in the following order of preference:

- file in **SNMP_PERSISTENT_FILE** environment variable
- directories in **SNMPCONFPATH** environment variable
- directory defined by **persistentDir** snmp.conf variable
- directory in **SNMP_PERSISTENT_DIR** environment variable
- default **/var/lib/snmp** directory

Finally, applications will write persistent configuration files in the following order of preference:

- file in **SNMP_PERSISTENT_FILE** environment variable
- directory defined by **persistentDir** snmp.conf variable
- directory in **SNMP_PERSISTENT_DIR** environment variable
- default **/var/lib/snmp** directory

Note: When using SNMP_PERSISTENT_FILE, the filename should match the application name. For example, /var/net-snmp/snmpd.conf.

CONFIGURATION FILE TYPES

Each application may use multiple configuration files, which will configure various different aspects of the application. For instance, the SNMP agent (**snmpd**) knows how to understand configuration directives in both the snmpd.conf and the snmp.conf files. In fact, most applications understand how to read the contents of the snmp.conf files. Note, however, that configuration directives understood in one file may not be understood in another file. For further information, read the associated manual page with each configuration file type. Also, most of the applications support a **-H** switch on the command line that will list the configuration files it will look for and the directives in each one that it understands.

The snmp.conf configuration file is intended to be a application suite wide configuration file that supports directives that are useful for controlling the fundamental nature of all of the SNMP applications, such as how they all manipulate and parse the textual SNMP MIB files.

SWITCHING CONFIGURATION TYPES IN MID-FILE

It's possible to switch in mid-file the configuration type that the parser is supposed to be reading. Since that sentence doesn't make much sense, let's give you an example: say that you wanted to turn on packet dumping output for the agent by default, but you didn't want to do that for the rest of the applications (ie, snmpget, snmpwalk, ...). Normally to enable packet dumping in the configuration file you'd need to put a line like:

```
dumpPacket true
```

into the snmp.conf file. But, this would turn it on for all of the applications. So, instead, you can put the same line in the snmpd.conf file so that it only applies to the snmpd daemon. However, you need to tell the parser to expect this line. You do this by putting a special type specification token inside a [] set. In other

words, inside your `snmpd.conf` file you could put the above `snmp.conf` directive by adding a line like so:

```
[snmp] dumpPacket true
```

This tells the parser to parse the above line as if it were inside a `snmp.conf` file instead of an `snmpd.conf` file. If you want to parse a bunch of lines rather than just one then you can make the context switch apply to the remainder of the file or until the next context switch directive by putting the special token on a line by itself:

```
# make this file handle snmp.conf tokens:
[snmp]
dumpPacket true
logTimestamp true
# return to our original snmpd.conf tokens:
[snmpd]
rocommunity mypublic
```

The same approach can be used to set configuration directives for a particular client application (or group of applications). For example, any program that uses the `'snmp_parse_args()'` call to handle command-line arguments (including the standard command-line tools shipped as part of the Net-SNMP distributions) will automatically read the config file `'snmpapp.conf'`. To set library-level settings for these applications (but not other more-specific tools), use configuration such as the following:

```
[snmp] defCommunity myCommunity
```

for a single directive, or

```
# make this file handle snmp.conf tokens:
[snmp]
defCommunity myCommunity
defVersion 2c
# return to our original snmpapp.conf tokens:
[snmpapp]
```

for multiple settings. Similarly for any other application token (as passed to `init_snmp()`).

COMMENTS

Any lines beginning with the character `'#'` in the configuration files are treated as a comment and are not parsed.

INCLUDING OTHER CONFIGURATION FILES

It is possible to include other configuration files for processing during normal configuration file processing.:

```
# include site specific config
includeFile site.conf
```

This will load the specified configuration file. The path to file must be either absolute, starting with `'/'`, or relative. The relative path is then relative to the directory where the parent file with `'includeFile'` directive resides.

The included file name does not need to have `'.conf'` suffix.

```
# include all *.conf files in a directory
includeDir /etc/snmp/config.d
```

This will search specified directory for all files with `'.conf'` suffix and process them as if they were included using `includeFile` directive. The configuration files are not processed in any particular order.

The specified directory must be absolute directory path.

API INTERFACE

Information about writing C code that makes use of this system in either the agent's MIB modules or in applications can be found in the *net-snmp-config-api(3)* manual page.

SEE ALSO

snmpconf(1), net-snmp-config-api(3), snmp.conf(5), snmpd.conf(5)