NAME

syslog.conf − syslogd(8) configuration file

DESCRIPTION

The *syslog.conf* file is the main configuration file for the **syslogd**(8) which logs system messages on *nix systems. This file specifies rules for logging. For special features see the **sysklogd**(8) manpage.

Every rule consists of two fields, a *selector* field and an *action* field. These two fields are separated by one or more spaces or tabs. The selector field specifies a pattern of facilities and priorities belonging to the specified action.

Lines starting with a hash mark (‘‘#’’) and empty lines are ignored.

This release of **syslogd** is able to understand an extended syntax. One rule can be divided into several lines if the leading line is terminated with an backslash (‘‘\’’).

SELECTORS

The selector field itself again consists of two parts, a *facility* and a *priority*, separated by a period (‘‘.’’). Both parts are case insensitive and can also be specified as decimal numbers, but don’t do that, you have been warned. Both facilities and priorities are described in **syslog**(3). The names mentioned below correspond to the similar **LOG_**-values in */usr/include/syslog.h*.

The *facility* is one of the following keywords: **auth**, **authpriv**, **cron**, **daemon**, **kern**, **lpr**, **mail**, **mark**, **news**, **security** (same as **auth**), **syslog**, **user**, **uucp** and **local0** through **local7**. The keyword **security** should not be used anymore and **mark** is only for internal use and therefore should not be used in applications. Anyway, you may want to specify and redirect these messages here. The *facility* specifies the subsystem that produced the message, i.e. all mail programs log with the mail facility (**LOG_MAIL**) if they log using syslog.

The *priority* is one of the following keywords, in ascending order: **debug**, **info**, **notice**, **warning**, **warn** (same as **warning**), **err**, **error** (same as **err**), **crit**, **alert**, **emerg**, **panic** (same as **emerg**). The keywords **error**, **warn** and **panic** are deprecated and should not be used anymore. The *priority* defines the severity of the message

The behavior of the original BSD syslogd is that all messages of the specified priority and higher are logged according to the given action. This **syslogd**(8) behaves the same, but has some extensions.

In addition to the above mentioned names the **syslogd**(8) understands the following extensions: An asterisk (‘‘*’’) stands for all facilities or all priorities, depending on where it is used (before or after the period). The keyword **none** stands for no priority of the given facility.

You can specify multiple facilities with the same priority pattern in one statement using the comma (‘‘,’’) operator. You may specify as much facilities as you want. Remember that only the facility part from such a statement is taken, a priority part would be skipped.

Multiple selectors may be specified for a single *action* using the semicolon (‘‘;’’) separator. Remember that each selector in the *selector* field is capable to overwrite the preceding ones. Using this behavior you can exclude some priorities from the pattern.

This **syslogd**(8) has a syntax extension to the original BSD source, that makes its use more intuitively. You may precede every priority with an equation sign (‘‘=’’) to specify only this single priority and not any of the above. You may also (both is valid, too) precede the priority with an exclamation mark (‘‘!’’) to ignore all that priorities, either exact this one or this and any higher priority. If you use both extensions than the exclamation mark must occur before the equation sign, just use it intuitively.

ACTIONS

The action field of a rule describes the abstract term ‘‘logfile’’. A ‘‘logfile’’ need not to be a real file, btw. The **syslogd**(8) provides the following actions.

Regular File

Typically messages are logged to real files. The file has to be specified with full pathname, beginning with a slash ‘‘/’’.

You may prefix each entry with the minus ‘‘-’’ sign to omit syncing the file after every logging. Note that you might lose information if the system crashes right behind a write attempt. Nevertheless this might give you back some performance, especially if you run programs that use logging in a very verbose manner.

Named Pipes

This version of **syslogd**(8) has support for logging output to named pipes (fifos). A fifo or named pipe can be used as a destination for log messages by prepending a pipe symbol (‘‘|’’) to the name of the file. This is handy for debugging. Note that the fifo must be created with the **mkfifo**(1) command before **syslogd**(8) is started.

Terminal and Console

If the file you specified is a tty, special tty-handling is done, same with */dev/console*.

Remote Machine

This **syslogd**(8) provides full remote logging, i.e. is able to send messages to a remote host running **syslogd**(8) and to receive messages from remote hosts. The remote host won’t forward the message again, it will just log them locally. To forward messages to another host, prepend the hostname with the at sign (‘‘@’’).

Using this feature you’re able to control all syslog messages on one host, if all other machines will log remotely to that. This tears down administration needs.

List of Users

Usually critical messages are also directed to ‘‘root’’ on that machine. You can specify a list of users that shall get the message by simply writing the login. You may specify more than one user by separating them with commas (‘‘,’’). If they’re logged in they get the message. Don’t think a mail would be sent, that might be too late.

Everyone logged on

Emergency messages often go to all users currently online to notify them that something strange is happening with the system. To specify this *wall*(1)-feature use an asterisk (‘‘*’’).

EXAMPLES

Here are some example, partially taken from a real existing site and configuration. Hopefully they rub out all questions to the configuration, if not, drop me (Joey) a line.

```
# Store critical stuff in critical
#
*.=crit;kern.none          /var/adm/critical
```

This will store all messages with the priority **crit** in the file */var/adm/critical*, except for any kernel message.

```
# Kernel messages are first, stored in the kernel
# file, critical messages and higher ones also go
# to another host and to the console
#
kern.*                  /var/adm/kernel
kern.crit               @finlandia
kern.crit               /dev/console
kern.info;kern.!err     /var/adm/kernel-info
```

The first rule direct any message that has the kernel facility to the file */var/adm/kernel*.

The second statement directs all kernel messages of the priority **crit** and higher to the remote host finlandia. This is useful, because if the host crashes and the disks get irreparable errors you might not be able to read the stored messages. If they're on a remote host, too, you still can try to find out the reason for the crash.

The third rule directs these messages to the actual console, so the person who works on the machine will get them, too.

The fourth line tells the syslogd to save all kernel messages that come with priorities from **info** up to **warning** in the file */var/adm/kernel-info*. Everything from *err* and higher is excluded.

```
# The tcp wrapper loggs with mail.info, we display
# all the connections on tty12
#
mail.=info              /dev/tty12
```

This directs all messages that uses **mail.info** (in source **LOG_MAIL** | **LOG_INFO**) to */dev/tty12*, the 12th console. For example the tcpwrapper **tcpd**(8) uses this as it's default.

```
# Store all mail concerning stuff in a file
#
mail.*;mail.!=info      /var/adm/mail
```

This pattern matches all messages that come with the **mail** facility, except for the **info** priority. These will be stored in the file */var/adm/mail*.

```
# Log all mail.info and news.info messages to info
#
mail,news.=info         /var/adm/info
```

This will extract all messages that come either with **mail.info** or with **news.info** and store them in the file */var/adm/info*.

```
# Log info and notice messages to messages file
#
*.=info;*.=notice;\
        mail.none  /var/log/messages
```

This lets the **syslogd** log all messages that come with either the **info** or the **notice** facility into the file */var/log/messages*, except for all messages that use the **mail** facility.

```
# Log info messages to messages file
#
*.=info;\
        mail,news.none      /var/log/messages
```

This statement causes the **syslogd** to log all messages that come with the **info** priority to the file */var/log/messages*. But any message coming either with the **mail** or the **news** facility will not be stored.

```
# Emergency messages will be displayed using wall
#
*.=emerg                *
```

This rule tells the **syslogd** to write all emergency messages to all currently logged in users. This is the wall action.

```
# Messages of the priority alert will be directed
# to the operator
#
*.alert                 root,joey
```

This rule directs all messages with a priority of **alert** or higher to the terminals of the operator, i.e. of the users ‘‘root’’ and ‘‘joey’’ if they’re logged in.

```
*.*                     @finlandia
```

This rule would redirect all messages to a remote host called finlandia. This is useful especially in a cluster of machines where all syslog messages will be stored on only one machine.

CONFIGURATION FILE SYNTAX DIFFERENCES

Syslogd uses a slightly different syntax for its configuration file than the original BSD sources. Originally all messages of a specific priority and above were forwarded to the log file. The modifiers ‘‘=’’, ‘‘!’’ and ‘‘-’’ were added to make the **syslogd** more flexible and to use it in a more intuitive manner.

The original BSD syslogd doesn’t understand spaces as separators between the selector and the action field.

FILES

/etc/syslog.conf
: Configuration file for **syslogd**

BUGS

The effects of multiple selectors are sometimes not intuitive. For example ‘‘mail.crit,*.err’’ will select ‘‘mail’’ facility messages at the level of ‘‘err’’ or higher, not at the level of ‘‘crit’’ or higher.

SEE ALSO

sysklogd(8), **klogd**(8), **logger**(1), **syslog**(2), **syslog**(3)

AUTHORS

The **syslogd** is taken from BSD sources, Greg Wettstein (greg@wind.enjellic.com) performed the port to Linux, Martin Schulze (joey@linux.de) made some bugfixes and added some new features.