```
#
# /etc/syslog-ng/syslog-ng.conf
#
# File format description can be found in syslog-ng.conf(5)
# and in /usr/share/doc/packages/syslog-ng/syslog-ng.txt.
#
# NOTE: The SuSEconfig script and its syslog-ng.conf.in
#       configuration template aren't used any more.
#
#       Feel free to edit this file directly.
#
#       Additional log sockets for chroot environments can
#       be declared in the /etc/sysconfig/syslog file using
#               SYSLOGD_ADDITIONAL_SOCKET<NAME>
#       variables. This way allows to define a socket from
#       RPM scripts and is used by several services, e.g.
#       bind and dhcpd.
#
#       The sockets defined in /etc/sysconfig/syslog file
#       are added by the /etc/ini.d/syslog init-script using
#       "-a path" command line options while syslog-ng is
#       started.
#
#       This syslog-ng contains an extension and appends the
#       sockets added by "-a" option to the same source group
#       and using the same socket type (unix-dgram) as the
#       "/dev/log" socket.
#       If one of the sockets added by "-a" option already
#       exists in any (other) source group in the config file,
#       then the socket added by "-a" option is ignored.
#

#
# Global options.
#
#options { long_hostnames(off); sync(0); perm(0640); stats(3600); };

#
# 'src' is our main source definition. you can add
# more sources driver definitions to it, or define
# your own sources, i.e.:
#
#source my_src { .... };
#
source src {
        #
        # include internal syslog-ng messages
        # note: the internal() soure is required!
        #
        internal();

        #
        # the default log socket for local logging:
        #
        unix-dgram("/dev/log");

        #
        # uncomment to process log messages from network:
        #
        #udp(ip("0.0.0.0") port(514));
};


#
# Filter definitions
#
filter f_iptables   { facility(kern) and match("IN=") and match("OUT="); };
```

```
filter f_console    { level(warn) and facility(kern) and not filter(f_i
                      or level(err) and not facility(authpriv); };

filter f_newsnotice { level(notice) and facility(news); };
filter f_newscrit   { level(crit)   and facility(news); };
filter f_newserr    { level(err)    and facility(news); };
filter f_news       { facility(news); };

filter f_mailinfo   { level(info)      and facility(mail); };
filter f_mailwarn   { level(warn)      and facility(mail); };
filter f_mailerr    { level(err, crit) and facility(mail); };
filter f_mail       { facility(mail); };

filter f_cron       { facility(cron); };

filter f_local      { facility(local0, local1, local2, local3,
                               local4, local5, local6, local7); };

#
# acpid messages
#
filter f_acpid_full { match('^\acpid:'); };
filter f_acpid      { level(emerg...notice) and match('^\acpid:'); };

# this is for the old acpid < 1.0.6
filter f_acpid_old  { match('^\[acpid\]:'); };

filter f_netmgm     { match('^NetworkManager:'); };

filter f_messages   { not (facility(news) or filter(f_iptables)) or
filter(f_mailwarn); };
filter f_warn       { level(warn, err, crit) and not filter(f_iptables); };
filter f_alert      { level(alert); };


#
# Enable this and adopt IP to send log messages to a log server.
#
#destination logserver { udp("10.10.10.10" port(514)); };
#log { source(src); destination(logserver); };

#
# Enable this, if you want to keep all messages in one file:
# (don't forget to provide logrotation config)
#
#destination allmessages { file("/var/log/allmessages"); };
#log { source(src); destination(allmessages); };

#
# Most warning and errors on tty10 and on the xconsole pipe:
#
destination console  { pipe("/dev/tty10"    owner(-1) group(-1) perm(-1)); };
log { source(src); filter(f_console); destination(console); };

destination xconsole { pipe("/dev/xconsole" owner(-1) group(-1) perm(-1)); };
log { source(src); filter(f_console); destination(xconsole); };

# Enable this, if you want that root is informed immediately,
# e.g. of logins:
#
#destination root { usertty("root"); };
#log { source(src); filter(f_alert); destination(root); };


#
# News-messages in separate files:
#
destination newscrit   { file("/var/log/news/news.crit"
```

```
                                owner(news) group(news)); };
log { source(src); filter(f_newscrit); destination(newscrit); };

destination newserr    { file("/var/log/news/news.err"
                                owner(news) group(news)); };
log { source(src); filter(f_newserr); destination(newserr); };

destination newsnotice { file("/var/log/news/news.notice"
                                owner(news) group(news)); };
log { source(src); filter(f_newsnotice); destination(newsnotice); };

#
# and optionally also all in one file:
# (don't forget to provide logrotation config)
#
#destination news { file("/var/log/news.all"); };
#log { source(src); filter(f_news); destination(news); };


#
# Mail-messages in separate files:
#
destination mailinfo { file("/var/log/mail.info"); };
log { source(src); filter(f_mailinfo); destination(mailinfo); };

destination mailwarn { file("/var/log/mail.warn"); };
log { source(src); filter(f_mailwarn); destination(mailwarn); };

destination mailerr  { file("/var/log/mail.err" fsync(yes)); };
log { source(src); filter(f_mailerr);  destination(mailerr); };

#
# and also all in one file:
#
destination mail { file("/usr/local/psa/var/log/maillog"); };
log { source(src); filter(f_mail); destination(mail); };


#
# acpid messages in one file:
#
destination acpid { file("/var/log/acpid"); };
destination null { };
log { source(src); filter(f_acpid); destination(acpid); flags(final); };
#
# if you want more verbose acpid logging, comment the destination(null)
# line and uncomment the destination(acpid) line
#
log { source(src); filter(f_acpid_full); destination(null); flags(final); };
# log { source(src); filter(f_acpid_full); destination(acpid); flags(final); };
#
# old acpid < 1.0.6
log { source(src); filter(f_acpid_old); destination(acpid); flags(final); };

#
# NetworkManager messages in one file:
#
destination netmgm { file("/var/log/NetworkManager"); };
log { source(src); filter(f_netmgm); destination(netmgm); flags(final); };


#
# Cron-messages in one file:
# (don't forget to provide logrotation config)
#
#destination cron { file("/var/log/cron"); };
#log { source(src); filter(f_cron); destination(cron); };
```

```
#
# Some boot scripts use/require local[1-7]:
#
destination localmessages { file("/var/log/localmessages"); };
log { source(src); filter(f_local); destination(localmessages); };


#
# All messages except iptables and the facilities news and mail:
#
destination messages { file("/var/log/messages"); };
log { source(src); filter(f_messages); destination(messages); };


#
# Firewall (iptables) messages in one file:
#
destination firewall { file("/var/log/firewall"); };
log { source(src); filter(f_iptables); destination(firewall); };


#
# Warnings (except iptables) in one file:
#
destination warn { file("/var/log/warn" fsync(yes)); };
log { source(src); filter(f_warn); destination(warn); };
```