

A Default Exim Configuration File

An Example configuration file taken from a default installation of Red Hat Enterprise Linux or CentOS. This file is similar to the one you would find in the /etc/exim/ directory on a server running exim v4.43.

```
#####
#           Runtime configuration file for Exim           #
#####

# This is a default configuration file which will operate correctly in
# uncomplicated installations. Please see the manual for a complete list
# of all the runtime configuration options that can be included in a
# configuration file. There are many more than are mentioned here. The
# manual is in the file doc/spec.txt in the Exim distribution as a plain
# ASCII file. Other formats (PostScript, Texinfo, HTML, PDF) are available
# from the Exim ftp sites. The manual is also online at the Exim web sites.

# This file is divided into several parts, all but the first of which are
# headed by a line starting with the word "begin". Only those parts that
# are required need to be present. Blank lines, and lines starting with #
# are ignored.

##### IMPORTANT ##### IMPORTANT ##### IMPORTANT #####
#
# Whenever you change Exim's configuration file, you *must* remember to #
# HUP the Exim daemon, because it will not pick up the new configuration #
# until you do. However, any other Exim processes that are started, for #
# example, a process started by an MUA in order to send a message, will #
# see the new configuration as soon as it is in place. #
#
# You do not need to HUP the daemon for changes in auxiliary files that #
# are referenced from this file. They are read every time they are used. #
#
# It is usually a good idea to test a new configuration for syntactic #
# correctness before installing it (for example, by running the command #
# "exim -C /config/file.new -bV"). #
#
##### IMPORTANT ##### IMPORTANT ##### IMPORTANT #####

#####
#           MAIN CONFIGURATION SETTINGS           #
#####

# Specify your host's canonical name here. This should normally be the fully
# qualified "official" name of your host. If this option is not set, the
# uname() function is called to obtain the name. In many cases this does
# the right thing and you need not set anything explicitly.

# primary_hostname =

# The next three settings create two lists of domains and one list of hosts.
```

These lists are referred to later in this configuration using the syntax
+local_domains, +relay_to_domains, and +relay_from_hosts, respectively. They
are all colon-separated lists:

```
domainlist local_domains = @  
domainlist relay_to_domains =  
hostlist relay_from_hosts = 127.0.0.1
```

Most straightforward access control requirements can be obtained by
appropriate settings of the above options. In more complicated situations, you
may need to modify the Access Control List (ACL) which appears later in this
file.

The first setting specifies your local domains, for example:

```
#  
# domainlist local_domains = my.first.domain : my.second.domain  
#
```

You can use "@" to mean "the name of the local host", as in the default
setting above. This is the name that is specified by primary_hostname,
as specified above (or defaulted). If you do not want to do any local
deliveries, remove the "@" from the setting above. If you want to accept mail
addressed to your host's literal IP address, for example, mail addressed to
"user@[192.168.23.44]", you can add "@[]" as an item in the local domains
list. You also need to uncomment "allow_domain_literals" below. This is not
recommended for today's Internet.

The second setting specifies domains for which your host is an incoming relay.
If you are not doing any relaying, you should leave the list empty. However,
if your host is an MX backup or gateway of some kind for some domains, you
must set relay_to_domains to match those domains. For example:

```
#  
# domainlist relay_to_domains = *.myco.com : my.friend.org  
#
```

This will allow any host to relay through your host to those domains.
See the section of the manual entitled "Control of relaying" for more
information.

The third setting specifies hosts that can use your host as an outgoing relay
to any other host on the Internet. Such a setting commonly refers to a
complete local network as well as the localhost. For example:

```
#  
# hostlist relay_from_hosts = 127.0.0.1 : 192.168.0.0/16  
#
```

The "/16" is a bit mask (CIDR notation), not a number of hosts. Note that you
have to include 127.0.0.1 if you want to allow processes on your host to send
SMTP mail by using the loopback address. A number of MUAs use this method of
sending mail.

All three of these lists may contain many different kinds of item, including
wildcarded names, regular expressions, and file lookups. See the reference
manual for details. The lists above are used in the access control list for
incoming messages. The name of this ACL is defined here:

```
acl_smtp_rcpt = acl_check_rcpt
```

You should not change that setting until you understand how ACLs work.

The following ACL entries are used if you want to do content scanning with
the exiscan-acl patch. When you uncomment one of these lines, you must also
review the respective entries in the ACL section further below.

```
# acl_smtp_mime = acl_check_mime  
# acl_smtp_data = acl_check_content
```

```
# This configuration variable defines the virus scanner that is used with
# the 'malware' ACL condition of the exiscan acl-patch. If you do not use
# virus scanning, leave it commented. Please read doc/exiscan-acl-readme.txt
# for a list of supported scanners.

# av_scanner = sophie:/var/run/sophie

# The following setting is only needed if you use the 'spam' ACL condition
# of the exiscan-acl patch. It specifies on which host and port the SpamAssassin
# "spamd" daemon is listening. If you do not use this condition, or you use
# the default of "127.0.0.1 783", you can omit this option.

# spamd_address = 127.0.0.1 783

# Specify the domain you want to be added to all unqualified addresses
# here. An unqualified address is one that does not contain an "@" character
# followed by a domain. For example, "caesar@rome.example" is a fully qualified
# address, but the string "caesar" (i.e. just a login name) is an unqualified
# email address. Unqualified addresses are accepted only from local callers by
# default. See the recipient_unqualified_hosts option if you want to permit
# unqualified addresses from remote sources. If this option is not set, the
# primary_hostname value is used for qualification.

# qualify_domain =

# If you want unqualified recipient addresses to be qualified with a different
# domain to unqualified sender addresses, specify the recipient domain here.
# If this option is not set, the qualify_domain value is used.

# qualify_recipient =

# The following line must be uncommented if you want Exim to recognize
# addresses of the form "user@[10.11.12.13]" that is, with a "domain literal"
# (an IP address) instead of a named domain. The RFCs still require this form,
# but it makes little sense to permit mail to be sent to specific hosts by
# their IP address in the modern Internet. This ancient format has been used
# by those seeking to abuse hosts by using them for unwanted relaying. If you
# really do want to support domain literals, uncomment the following line, and
# see also the "domain_literal" router below.

# allow_domain_literals

# No deliveries will ever be run under the uids of these users (a colon-
# separated list). An attempt to do so causes a panic error to be logged, and
# the delivery to be deferred. This is a paranoid safety catch. There is an
# even stronger safety catch in the form of the FIXED_NEVER_USERS setting
# in the configuration for building Exim. The list of users that it specifies
# is built into the binary, and cannot be changed. The option below just adds
# additional users to the list. The default for FIXED_NEVER_USERS is "root",
# but just to be absolutely sure, the default here is also "root".

# Note that the default setting means you cannot deliver mail addressed to root
# as if it were a normal user. This isn't usually a problem, as most sites have
# an alias for root that redirects such mail to a human administrator.

never_users = root

# The setting below causes Exim to do a reverse DNS lookup on all incoming
# IP calls, in order to get the true host name. If you feel this is too
# expensive, you can specify the networks for which a lookup is done, or
# remove the setting entirely.
```

```
host_lookup = *
```

```
# The settings below, which are actually the same as the defaults in the
# code, cause Exim to make RFC 1413 (ident) callbacks for all incoming SMTP
# calls. You can limit the hosts to which these calls are made, and/or change
# the timeout that is used. If you set the timeout to zero, all RFC 1413 calls
# are disabled. RFC 1413 calls are cheap and can provide useful information
# for tracing problem messages, but some hosts and firewalls have problems
# with them. This can result in a timeout instead of an immediate refused
# connection, leading to delays on starting up an SMTP session.
```

```
rfc1413_hosts = *
rfc1413_query_timeout = 30s
```

```
# By default, Exim expects all envelope addresses to be fully qualified, that
# is, they must contain both a local part and a domain. If you want to accept
# unqualified addresses (just a local part) from certain hosts, you can specify
# these hosts by setting one or both of
```

```
#
# sender_unqualified_hosts =
# recipient_unqualified_hosts =
#
# to control sender and recipient addresses, respectively. When this is done,
# unqualified addresses are qualified using the settings of qualify_domain
# and/or qualify_recipient (see above).
```

```
# If you want Exim to support the "percent hack" for certain domains,
# uncomment the following line and provide a list of domains. The "percent
# hack" is the feature by which mail addressed to x%y@z (where z is one of
# the domains listed) is locally rerouted to x@y and sent on. If z is not one
# of the "percent hack" domains, x%y is treated as an ordinary local part. This
# hack is rarely needed nowadays; you should not enable it unless you are sure
# that you really need it.
```

```
#
# percent_hack_domains =
#
# As well as setting this option you will also need to remove the test
# for local parts containing % in the ACL definition below.
```

```
# When Exim can neither deliver a message nor return it to sender, it "freezes"
# the delivery error message (aka "bounce message"). There are also other
# circumstances in which messages get frozen. They will stay on the queue for
# ever unless one of the following options is set.
```

```
# This option unfreezes frozen bounce messages after two days, tries
# once more to deliver them, and ignores any delivery failures.
```

```
ignore_bounce_errors_after = 2d
```

```
# This option cancels (removes) frozen messages that are older than a week.
```

```
timeout_frozen_after = 7d
```

```
#####
#           ACL CONFIGURATION           #
#   Specifies access control lists for incoming SMTP mail   #
#####
```

```
begin acl
```

```
# This access control list is used for every RCPT command in an incoming
# SMTP message. The tests are run in order until the address is either
# accepted or denied.
```

```
acl_check_rcpt:
```

```
# Accept if the source is local SMTP (i.e. not over TCP/IP). We do this by
# testing for an empty sending host field.
```

```
accept hosts = :
```

```
#####
```

```
# The following section of the ACL is concerned with local parts that contain
# @ or % or ! or / or | or dots in unusual places.
```

```
#
# The characters other than dots are rarely found in genuine local parts, but
# are often tried by people looking to circumvent relaying restrictions.
# Therefore, although they are valid in local parts, these rules lock them
# out, as a precaution.
```

```
#
# Empty components (two dots in a row) are not valid in RFC 2822, but Exim
# allows them because they have been encountered. (Consider local parts
# constructed as "firstinitial.secondinitial.familyname" when applied to
# someone like me, who has no second initial.) However, a local part starting
# with a dot or containing ../ can cause trouble if it is used as part of a
# file name (e.g. for a mailing list). This is also true for local parts that
# contain slashes. A pipe symbol can also be troublesome if the local part is
# incorporated unthinkingly into a shell command line.
```

```
#
# Two different rules are used. The first one is stricter, and is applied to
# messages that are addressed to one of the local domains handled by this
# host. It blocks local parts that begin with a dot or contain @ % ! / or |.
# If you have local accounts that include these characters, you will have to
# modify this rule.
```

```
deny message = Restricted characters in address
domains = +local_domains
local_parts = ^[.] : ^.*[!@%/|]
```

```
# The second rule applies to all other domains, and is less strict. This
# allows your own users to send outgoing messages to sites that use slashes
# and vertical bars in their local parts. It blocks local parts that begin
# with a dot, slash, or vertical bar, but allows these characters within the
# local part. However, the sequence ../ is barred. The use of @ % and ! is
# blocked, as before. The motivation here is to prevent your users (or
# your users' viruses) from mounting certain kinds of attack on remote sites.
```

```
deny message = Restricted characters in address
domains = !+local_domains
local_parts = ^[./|] : ^.*[!@%] : ^.*\.\./
```

```
#####
```

```
# Accept mail to postmaster in any local domain, regardless of the source,
# and without verifying the sender.
```

```
accept local_parts = postmaster
domains = +local_domains
```

```
# Deny unless the sender address can be verified.
```

```
require verify = sender
```

```
#####
```

```

# There are no checks on DNS "black" lists because the domains that contain
# these lists are changing all the time. However, here are two examples of
# how you could get Exim to perform a DNS black list lookup at this point.
# The first one denies, while the second just warns.
#
# deny message = rejected because $sender_host_address is in a black list at $dnslist_domain\n$dnslist_text
#   dnslists = black.list.example
#
# warn message = X-Warning: $sender_host_address is in a black list at $dnslist_domain
#   log_message = found in $dnslist_domain
#   dnslists = black.list.example
#####

# Accept if the address is in a local domain, but only if the recipient can
# be verified. Otherwise deny. The "endpass" line is the border between
# passing on to the next ACL statement (if tests above it fail) or denying
# access (if tests below it fail).

accept domains = +local_domains
  endpass
  verify = recipient

# Accept if the address is in a domain for which we are relaying, but again,
# only if the recipient can be verified.

accept domains = +relay_to_domains
  endpass
  verify = recipient

# If control reaches this point, the domain is neither in +local_domains
# nor in +relay_to_domains.

# Accept if the message comes from one of the hosts for which we are an
# outgoing relay. Recipient verification is omitted here, because in many
# cases the clients are dumb MUAs that don't cope well with SMTP error
# responses. If you are actually relaying out from MTAs, you should probably
# add recipient verification here.

accept hosts = +relay_from_hosts

# Accept if the message arrived over an authenticated connection, from
# any host. Again, these messages are usually from MUAs, so recipient
# verification is omitted.

accept authenticated = *

# Reaching the end of the ACL causes a "deny", but we might as well give
# an explicit message.

deny message = relay not permitted

# These access control lists are used for content scanning with the exiscan-acl
# patch. You must also uncomment the entries for acl_smtp_data and acl_smtp_mime
# (scroll up), otherwise the ACLs will not be used. IMPORTANT: the default entries here
# should be treated as EXAMPLES. You MUST read the file doc/exiscan-acl-spec.txt
# to fully understand what you are doing ...

acl_check_mime:

# Decode MIME parts to disk. This will support virus scanners later.
warn decode = default

# File extension filtering.
deny message = Blacklisted file extension detected

```

```

condition = ${if match \
    ${lc:$mime_filename} \
    {\N(\.exe|\.pif|\.bat|\.scr|\.lnk|\.com)$\N} \
    {1}{0}}

# Reject messages that carry chinese character sets.
# WARNING: This is an EXAMPLE.
deny message = Sorry, noone speaks chinese here
condition = ${if eq{$mime_charset}{gb2312}{1}{0}}

accept

acl_check_content:

# Reject virus infested messages.
deny message = This message contains malware ($malware_name)
malware = *

# Always add X-Spam-Score and X-Spam-Report headers, using SA system-wide settings
# (user "nobody"), no matter if over threshold or not.
warn message = X-Spam-Score: $spam_score ($spam_bar)
spam = nobody:true
warn message = X-Spam-Report: $spam_report
spam = nobody:true

# Add X-Spam-Flag if spam is over system-wide threshold
warn message = X-Spam-Flag: YES
spam = nobody

# Reject spam messages with score over 10, using an extra condition.
deny message = This message scored $spam_score points. Congratulations!
spam = nobody:true
condition = ${if >{$spam_score_int}{100}{1}{0}}

# finally accept all the rest
accept

#####
#          ROUTERS CONFIGURATION          #
#          Specifies how addresses are handled          #
#####
# THE ORDER IN WHICH THE ROUTERS ARE DEFINED IS IMPORTANT! #
# An address is passed to each router in turn until it is accepted. #
#####

begin routers

# This router routes to remote hosts over SMTP by explicit IP address,
# when an email address is given in "domain literal" form, for example,
# <user@[192.168.35.64]>. The RFCs require this facility. However, it is
# little-known these days, and has been exploited by evil people seeking
# to abuse SMTP relays. Consequently it is commented out in the default
# configuration. If you uncomment this router, you also need to uncomment
# allow_domain_literals above, so that Exim can recognize the syntax of
# domain literal addresses.

# domain_literal:
# driver = ipliteral
# domains = ! +local_domains
# transport = remote_smtp

# This router routes addresses that are not in local domains by doing a DNS
# lookup on the domain name. Any domain that resolves to 0.0.0.0 or to a

```

```
# loopback interface address (127.0.0.0/8) is treated as if it had no DNS
# entry. Note that 0.0.0.0 is the same as 0.0.0.0/32, which is commonly treated
# as the local host inside the network stack. It is not 0.0.0.0/0, the default
# route. If the DNS lookup fails, no further routers are tried because of
# the no_more setting, and consequently the address is unrouteable.
```

```
dnslookup:
```

```
driver = dnslookup
domains = ! +local_domains
transport = remote_smtp
ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
no_more
```

```
# The remaining routers handle addresses in the local domain(s).
```

```
# This router handles aliasing using a linearly searched alias file with the
# name SYSTEM_ALIASES_FILE. When this configuration is installed automatically,
# the name gets inserted into this file from whatever is set in Exim's
# build-time configuration. The default path is the traditional /etc/aliases.
# If you install this configuration by hand, you need to specify the correct
# path in the "data" setting below.
```

```
#
##### NB You must ensure that the alias file exists. It used to be the case
##### NB that every Unix had that file, because it was the Sendmail default.
##### NB These days, there are systems that don't have it. Your aliases
##### NB file should at least contain an alias for "postmaster".
```

```
#
# If any of your aliases expand to pipes or files, you will need to set
# up a user and a group for these deliveries to run under. You can do
# this by uncommenting the "user" option below (changing the user name
# as appropriate) and adding a "group" option if necessary. Alternatively, you
# can specify "user" on the transports that are used. Note that the transports
# listed below are the same as are used for .forward files; you might want
# to set up different ones for pipe and file deliveries from aliases.
```

```
system_aliases:
```

```
driver = redirect
allow_fail
allow_defer
data = ${lookup{$local_part}lsearch{/etc/aliases}}
# user = exim
file_transport = address_file
pipe_transport = address_pipe
```

```
# This router handles forwarding using traditional .forward files in users'
# home directories. If you want it also to allow mail filtering when a forward
# file starts with the string "# Exim filter" or "# Sieve filter", uncomment
# the "allow_filter" option.
```

```
# If you want this router to treat local parts with suffixes introduced by "-"
# or "+" characters as if the suffixes did not exist, uncomment the two local_
# part_suffix options. Then, for example, xxxx-foo@your.domain will be treated
# in the same way as xxxx@your.domain by this router. You probably want to make
# the same change to the localuser router.
```

```
# The no_verify setting means that this router is skipped when Exim is
# verifying addresses. Similarly, no_expn means that this router is skipped if
# Exim is processing an EXPN command.
```

```
# The check_ancestor option means that if the forward file generates an
# address that is an ancestor of the current one, the current one gets
# passed on instead. This covers the case where A is aliased to B and B
```

has a .forward file pointing to A.

The three transports specified at the end are those that are used when
forwarding generates a direct delivery to a file, or to a pipe, or sets
up an auto-reply, respectively.

```
userforward:
  driver = redirect
  check_local_user
# local_part_suffix = +* : -*
# local_part_suffix_optional
  file = $home/.forward
# allow_filter
  no_verify
  no_expn
  check_ancestor
  file_transport = address_file
  pipe_transport = address_pipe
  reply_transport = address_reply
```

This router matches local user mailboxes. If the router fails, the error
message is "Unknown user".

If you want this router to treat local parts with suffixes introduced by "-"
or "+" characters as if the suffixes did not exist, uncomment the two local_
part_suffix options. Then, for example, xxxx-foo@your.domain will be treated
in the same way as xxxx@your.domain by this router.

```
localuser:
  driver = accept
  check_local_user
# local_part_suffix = +* : -*
# local_part_suffix_optional
  transport = local_delivery
  cannot_route_message = Unknown user
```

```
#####
#          TRANSPORTS CONFIGURATION          #
#####
#          ORDER DOES NOT MATTER             #
#  Only one appropriate transport is called for each delivery.  #
#####
```

A transport is used only when referenced from a router that successfully
handles an address.

begin transports

This transport is used for delivering messages over SMTP connections.

```
remote_smtp:
  driver = smtp
```

This transport is used for local delivery to user mailboxes in traditional
BSD mailbox format. By default it will be run under the uid and gid of the
local user, and requires the sticky bit to be set on the /var/mail directory.
Some systems use the alternative approach of running mail deliveries under a
particular group instead of using the sticky bit. The commented options below
show how this can be done.

```
local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  envelope_to_add
  return_path_add
  group = mail
  mode = 0660
```

```
# This transport is used for handling pipe deliveries generated by alias or
# .forward files. If the pipe generates any standard output, it is returned
# to the sender of the message as a delivery error. Set return_fail_output
# instead of return_output if you want this to happen only when the pipe fails
# to complete normally. You can set different transports for aliases and
# forwards if you want to - see the references to address_pipe in the routers
# section above.
```

```
address_pipe:
  driver = pipe
  return_output
```

```
# This transport is used for handling deliveries directly to files that are
# generated by aliasing or forwarding.
```

```
address_file:
  driver = appendfile
  delivery_date_add
  envelope_to_add
  return_path_add
```

```
# This transport is used for handling autoreplies generated by the filtering
# option of the userforward router.
```

```
address_reply:
  driver = autoreply
```

```
# This transport is used to deliver local mail to cyrus IMAP server via UNIX
# socket.
```

```
#
#local_delivery:
# driver = lmtp
# command = "/usr/lib/cyrus-imapd/deliver -l"
# batch_max = 20
# user = cyrus
```

```
#####
#          RETRY CONFIGURATION          #
#####
```

```
begin retry
```

```
# This single retry rule applies to all domains and all errors. It specifies
# retries every 15 minutes for 2 hours, then increasing retry intervals,
# starting at 1 hour and increasing each time by a factor of 1.5, up to 16
# hours, then retries every 6 hours until 4 days have passed since the first
# failed delivery.
```

```
# Address or Domain  Error    Retries
# -----  ----  -
```

* * F,2h,15m; G,16h,1h,1.5; F,4d,6h

```
#####  
#           REWRITE CONFIGURATION           #  
#####
```

There are no rewriting specifications in this default configuration file.

begin rewrite

```
#####  
#           AUTHENTICATION CONFIGURATION     #  
#####
```

There are no authenticator specifications in this default configuration file.

begin authenticators

```
#####  
#           CONFIGURATION FOR local_scan()   #  
#####
```

If you have built Exim to include a local_scan() function that contains
tables for private options, you can define those options here. Remember to
uncomment the "begin" line. It is commented by default because it provokes
an error with Exim binaries that are not built with LOCAL_SCAN_HAS_OPTIONS
set in the Local/Makefile.

begin local_scan

End of Exim configuration file