

**NAME**

zmailer.conf – file format

**SYNOPSIS**

These are generated from inline XMLishly tagged descriptions in generated file: *SiteConfig*.

**DESCRIPTION**

ZMailer 'ZENV' Environment Variables:

A config file for storing various *site config* parameters

These are generic material that **do not depend upon compilers et.al!**

This is generated with autoconfiguration process from *SiteConfig.in* file, and the configuration parameters.

Following paragraph appears to be outdated:

The standard installation procedure **will not** overwrite the destination **ZCONFIG** file with the new one. This file contains one parameter that is compiled in the binaries, and is not overridable with this file, namely the **ZCONFIG** variable, alias the final location of the processed result of this SiteConfig file.

**ZCONFIG**

ZCONFIG is the pathname of the configuration file specifying all the other host-dependent information needed by ZMailer programs. This file is created from the *SiteConfig* file in the distribution (the file you are reading right now), and contains variable assignments in an sh-compatible format.

Several top-level components of ZMailer do have runtime options to choose some other path, than compiled-in default for this: (/opt/mail/zmailer.conf).

Some programs (notably '*sendmail*(8zm)', and transport agents) do check for standard UNIX environment variables for overrider of this value.)

**MAILBIN**

MAILBIN is the directory hierarchy containing all ZMailer binaries.

Configured as: /opt/mail/bin

**MAILSHARE**

MAILSHARE is the directory hierarchy containing site-wide configuration files and databases.

Configured as: /opt/mail

**MAILVAR**

MAILVAR is the directory that will contain machine-specific configuration files and databases.

Configured as: /opt/mail

**MAILBOX**

MAILBOX is the directory containing all the user mailboxes. This is defaulted inside the *mailbox*(8zm) program and may be overridden here.

Configured as: /var/spool/mail

**POSTOFFICE**

POSTOFFICE is the directory hierarchy used to manipulate message files, where runtime activity takes places.

Configured as: /var/spool/postoffice

**LOGDIR**

LOGDIR is the directory where log files will appear.

Configured as: /var/log/mail

**INPUTDIRS**

INPUTDIRS are alternate directories to receive files **instead** of standard "router" and \$ROUTERDIRS set.

**INPUTDIRHASH**

When defined, INPUTDIRHASH submits messages immediately into the **If no such "hash subdirs" exist, message submission will fail!** (The value must be "1" for this to take effect!)

**INPUTNOTIFY**

The INPUTNOTIFY defines, where is a socket at which some pre-router queue processor listens for PF\_UNIX/SOCK\_DGRAM messages telling paths to new jobs.

The injection library informs the router queuing subsystem of new jobs, and does it with same message as ROUTERNOTIFY= uses.

**ROUTEROPTIONS**

ROUTEROPTIONS are command line options given to the *router*(8zm) when started from the *zmailer*(1zm) shell script.

The default values are "-dWkn 4"

**ROUTERDIRS**

Multiple LOWER priorities on message routing can be defined by creating \$POSTOFFICE/<component-of-\$ROUTERDIRS> -directories.

Routers process first \$POSTOFFICE/router/ -directory, and once it is empty, files from subsequent dirs. See *mail*(3) *mail\_priority*

These can be only under the \$POSTOFFICE.

**ROUTERDIRHASH**

When defined, ROUTERDIRHASH submits messages immediately into the **If no such "hash subdirs" exist, message submission will fail!** (The value must be "1" for this to take effect!)

**ROUTERNOTIFY**

The ROUTERNOTIFY defines, where is a socket at which the router listens for PF\_UNIX/SOCK\_DGRAM messages telling paths to new jobs. The injection library informs the router queuing subsystem of new jobs.

**SMTPOPTIONS**

SMTPOPTIONS are command line options given to the smtpserver when started from the zmailer shell script. The intent is that if you want non-default address verification options they can be specified here. The default value is "-sve". This is also used, when invoking "sendmail" with "-bs" option.

**ALLOWSOURCEROUTE**

VALID ANYMORE ??? ALLOWSOURCEROUTE (when present) stops the system from ignoring the old RFC821/822 source routes of type: @a,@b:c@d; By "ignoring" we mean here that system chops away "@a,@b:" and uses only: c@d This is done at all input portals; smtpserver, and at sendmail/rmail. System behaviour is prone to problems in case this is ever enabled!

**SCHEDULEROPTIONS**

SCHEDULEROPTIONS are command line options given to the scheduler when started from the zmailer shell script. The intent is that if you want non-default logging options, the can be specified here.

The configured default value is: "-S -H"

**SCHEDULERDIRHASH**

The SCHEDULERDIRHASH is magic thing to tell to the router that it should move resulting files directly into hash subdir(s) of the scheduler subsystem, and not only to the

main-level. Existence of this variable also overrides -H option(s) to the scheduler. Value is the number of -H options. If these hash subdirectories don't exist, system failure happens! **Systems with low loads can do with value "1", while systems with large queues should definitely use value "2" !**

#### SCHEDULERNOTIFY

The SCHEDULERNOTIFY defines, where is a socket at which the scheduler listens for PF\_UNIX/SOCK\_DGRAM messages telling paths to new jobs. The router(s) inform the scheduler of new jobs.

The configured default is: /var/spool/postoffice/.scheduler.notify

Following few ZM "environment" variables are used by the *router*(8zm) configuration at your site.

#### MAILSERVER

MAILSERVER is the hostname of the remote machine where the postoffice is located. This value is only needed in an environment with distributed file systems, and if it exists will be used by the mail queue querying program as the default name of the host to query. It is a way of overriding the algorithm used by mailq in an NFS environment, or when you are running a different kind of DFS. Usually undefined or a hostname.

#### PUNTHOST

PUNTHOST is where mail that is supposed to go to a local address, but no such address exists, is punted to.

#### FORCEPUNT

FORCEPUNT is for cases when the local machine under no circumstances is to store any email locally, but send all such to this given address (local host is a member on a "cluster" whose message store is at some other cluster server, and said node handles "local" delivery for all cluster members... \*including\* running pipes..)

#### FORCEPUNT

SMARTHOST is where mail that cannot be resolved or routed is punted to.

There used to be a variable for this, now a better way is to use 'routes' database at which you put line: . smtp!smart.host.name (That is: dot, white-space(s), "smtp!smart.host.name " )

#### NOBODY

NOBODY is the unprivileged UID value.

This is absolutely necessary if setuid() will fail on your "nobody" account uid (if it is -2, for example). Make sure that whatever value you give here will work with setuid(). Values between 1 and 29999 will usually work.

**Be carefull with this! The system relies on it very much indeed!**

(On SunOS 4.1.x, the value of "-2" works the best, on Solaris the default for nobody is 60001! If your system has "nobody" "account", use here the name instead of number -- it should (usually) work)

-- Use a mapping via /etc/passwd, this is most generic..

#### LOGLEVEL

LOGLEVEL may be set to restrict the log output of the router to entries whose tags are found in the specified string value. The currently known tags are:

address: deferred: file: header\_defer: info: recipient:

#### NNTPSERVER

Builtin USENET channel uses NNTPSERVER variable (depending upon your inews ..) to send the article to..

**SENDMAILPATH**

Where the sendmail (compability one) shall be located ?

The configured default is: /usr/sbin/sendmail

**RMAILPATH**

Where is the rmail to be located at ?

The configured default is: /bin/rmail

**TA\_USE\_MMAP**

TA\_USE\_MMAP - value "1" in this variable mmap()s control-, and message body files into transport agent process memories. This may, or may not give performance boost.

This has no effect, if the system does not have functional *mmap(2)* system call.

The configured default is: 0

**TALOCKMODE**

TALOCKMODE=[TFW] -- don't use!

**MBOXLOCKS**

MAILBOX locking scheme -- no configuration option (yet) See man-page of *mailbox(8zm)* for details; the order of key-chars is meaningfull:

‘.’ Dotlock scheme for mailboxes at \$MAILBOX/ directory

‘F’ flock() locking of files (and perhaps mailboxes)

‘L’ lockf() locking of files (and perhaps mailboxes)

‘.’ Separates the two parts of the parameter; left part is for the mailbox locking, and right part is for all other kinds of files.

We use compiled-in defaults at the mailbox program! Following examples are for flock(), and lockf() systems with their respective defaults. ( Systems capable to use both will use lockf() )

**SELFADDRESSES**

The SELFADDRESSES is a comma separated list of IP address literals listing all of our acceptable IP addresses (Comma because IPv6 uses colon for short-hand notation..):

For usual (IPv4) universe, no addresses are needed listing, however for IPv6 it may be necessary - likewise if you want to use cluster-mode, you may want to list all *\*cluster\** addresses here - nodes know only their local ones, after all.. (See: doc/guides/etrn-cluster)

**DBTYPE**

What kind of DB type we prefer to use ? We can support several, simultaneously, after all; btree/ndbm/gdbm ...

Configured default value: btree

**DBEXT**

What extension that particular DB type wants for open operations ?

Configured default value: .db

**DBEXTtest**

What extension that particular DB type will actually have ? This is used in *file age tests*.

Configured default value: .db

**DEFCHARSET**

The charset to be used as a default when turning 8-bit containing headers to MIME-2 headers -- and what to say at the default generated "Content-Type: text/plain; charset=XXXX" -header in case the original message was not of MIME, and still had 8-bit chars...

## RFC822TABS

We want those nice tabs between the header field name and value. The task of generating TABS or SPACES is at `TA *writeheaders()`. Value '0' here yields expansion of possibly existing header resident line-start TABS. There is no mechanism to turn line-start SPACES to TABS with any other value stored here.

## NORECEIVEDFORCLAUSE

Existence `NORECEIVEDFORCLAUSE` ZENV variable forbids the transport agents from adding '(ORCPT <...> ...)' or 'for <...>' clauses to the top-most "Received:" header.

## SYSLOGFLG

`SYSLOGFLG` tells which systems use syslog to log things: Set of chars which are as follows:

S smtpserver and /usr/sbin/sendmail  
 R router  
 T transport agents  
 C scheduler completion of a message

## TRUSTEDUSER

Per default, ZMailer uses "daemon" userid when it wants to operate in "runastrusted-user()" mode. Finding that userid (or rather its numeric uid) can be a bit difficult, and if it *fails*, apparently uid 65535 will be used.

## ORGDOMAIN

Use `ORGDOMAIN` in ZENV if the system can't generate MIME multipart boundary string contained host/domain ids automatically...

## ROUTEUSER\_IN\_ABNORMAL\_UNIX

Depending, are you running strange private customer account databases hooked (only) into 'mailbox', or not, make sure following is non-empty if you \*are\* using private databases, as then ZMailer's router won't claim wrongly userid to be nonexistent.. These shunted tests look for `HOMEDIRECTORY`, which might be nonexistent thing at such funny systems... An EMPTY string means "this is NORMAL unix":

(A "bug" is that this isn't automatically substituted, but non-void content gives behaviour that has been around for quite a while...)

## LISTSERV

Some sites (well, one FUNET site), has `LISTSERV`, this is for configuring that subpart of the `aliases.cf` scripts:

## BINDADDR

If defined, `BINDADDR` specifies to which local interface to bind smtpserver, smtp transport agent and scheduler. Possible specification formats are:

```
[0.0.0.0]
[IPv6.0.:0]
iface:eth0:1
```

## SNMPSHAREDFILE

SNMP-like global system instance monitoring datablock path: This `_file_` has absolute path (no substitutions are allowed), it is shared in between all principal subsystem components in `mmap(MAP_SHARED, MAP_READ|MAP_WRITE)` mode. Counters in this file are NEVER reset. Gauges are managed as shadows of subsystem internal state.

## DOMAIN\_AWARE\_GETPWNAM

Define this to "1" if you use (replacement) `getpwnam()` that handles username together with domain. This is a convenient way to support virtual domains: users in different domains automatically have different userids, different homes etc. Actually, this often can be arranged even within standard `/etc/passwd` model, at least on Linux.

**SEE ALSO**

*zmailer*(1zm),

**AUTHOR**

This document authored and copyright by:  
Matti Aarnio <mea@nic.funet.fi>