Web services and email

The two most popular services visibly provided by servers are email and web-type services. Full email setups generally consists of an MTA such as sendmail or postfix, a delivery agent such as procmail or dropmail, a pop/imap server, and perhaps a webmail interface such as openwebmail, Outlook Web Access (OWA), horde, or squirrelmail. They may also include various spam and virus programs, such as MailScanner, spamassassin, avis, clamav, dcc, razor, and many others, and other mail types of mail filters such as the popular milter library programs (e.g., milter-ahead).

Web services generally center around an Apache web server, some CGI-friendly regime such as Perl (anywhere from embedded Perl to mod_perl with any of the numerous CGI packages), Python, PHP, Ruby, JSP, ASP, and a database such as MySQL, Postgresql, Oracle, or SQLite. It may also include other bits such as SOAP or RSS services.

Email: sendmail

☞ Sendmail functions as a MTA (and also a RFC 2476 MSA). It is generally configured to listen to port 25 (and 587 for MSA functions), and the configuration files are now generally stored in `/etc/mail`.

☞ The primary configuration for administrators typically is `/etc/mail/sendmail.mc` This contains **m4** directives to control the creation of `/etc/mail/sendmail.cf`

☞ An example /etc/mail/sendmail.mc:

```
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package is
dnl # installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #
include(‘/usr/share/sendmail-cf/m4/cf.m4’)dnl
VERSIONID(‘setup for Red Hat Linux’)dnl
OSTYPE(‘linux’)dnl
dnl #
dnl # default logging level is 9, you might want to set it higher to
dnl # debug the configuration
dnl #
dnl define(‘confLOG_LEVEL’, ‘9’)dnl
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(‘SMART_HOST’,‘smtp.your.provider’)
dnl #
define(‘confDEF_USER_ID’,‘‘8:12’’)dnl
dnl define(‘confAUTO_REBUILD’)dnl
define(‘confTO_CONNECT’, ‘1m’)dnl
define(‘confTRY_NULL_MX_LIST’,true)dnl
define(‘confDONT_PROBE_INTERFACES’,true)dnl
dnl define(‘PROCMAIL_MAILER_PATH’,‘/usr/bin/procmail’)dnl
define(‘ALIAS_FILE’, ‘/etc/aliases’)dnl
define(‘STATUS_FILE’, ‘/var/log/mail/statistics’)dnl
define(‘UUCP_MAILER_MAX’, ‘2000000’)dnl
define(‘confUSERDB_SPEC’, ‘/etc/mail/userdb.db’)dnl
define(‘confPRIVACY_FLAGS’, ‘authwarnings,novrfy,noexpn,restrictqrun’)dnl
define(‘confAUTH_OPTIONS’, ‘A’)dnl
dnl #
dnl # The following allows relaying if the user authenticates, and disallows
dnl # plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(‘confAUTH_OPTIONS’, ‘A p’)dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl #
dnl TRUST_AUTH_MECH(‘EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN’)dnl
dnl define(‘confAUTH_MECHANISMS’, ‘EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAI
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #     make -C /usr/share/ssl/certs usage
dnl # or use the included makecert.sh script
dnl #
dnl define(‘confCACERT_PATH’,‘/usr/share/ssl/certs’)
dnl define(‘confCACERT’,‘/usr/share/ssl/certs/ca-bundle.crt’)
dnl define(‘confSERVER_CERT’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl define(‘confSERVER_KEY’,‘/usr/share/ssl/certs/sendmail.pem’)
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP’s
dnl # slapd, which requires the file to be readble by group ldap
dnl #
dnl define(‘confDONT_BLAME_SENDMAIL’,‘groupreadablekeyfile’)dnl
dnl #
dnl define(‘confTO_QUEUEWARN’, ‘4h’)dnl
dnl define(‘confTO_QUEUERETURN’, ‘5d’)dnl
dnl define(‘confQUEUE_LA’, ‘12’)dnl
dnl define(‘confREFUSE_LA’, ‘18’)dnl
define(‘confTO_IDENT’, ‘0’)dnl
dnl FEATURE(delay_checks)dnl
FEATURE(‘no_default_msa’,‘dnl’)dnl
FEATURE(‘smrsh’,‘/usr/sbin/smrsh’)dnl
FEATURE(‘mailertable’,‘hash -o /etc/mail/mailertable.db’)dnl
dnl FEATURE(‘virtusertable’,‘hash -o /etc/mail/virtusertable.db’)dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
dnl FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
```

```
define(‘PROCMAIL_MAILER_PATH’,‘/usr/bin/procmail, U=vmail:vmail’)dnl
VIRTUSER_DOMAIN_FILE(‘-o /etc/mail/virtuserdomains’)dnl
FEATURE(‘virtusertable’,‘hash -o /etc/mail/virtusertable.db’)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his quota.
dnl #
dnl FEATURE(local_procmail,‘’,‘procmail -t -Y -a $h -d $u’)dnl
FEATURE(local_procmail,‘/usr/bin/procmail’,‘procmail -t -Y -a $h -d $u’)dnl
FEATURE(‘access_db’,‘hash -T<TMPF> -o /etc/mail/access.db’)dnl
FEATURE(‘blacklist_recipients’)dnl
define(‘PROCMAIL_MAILER_ARGS’,‘procmail -t -Y -a $h -a $u’)dnl  according to docu
define(‘PROCMAIL_MAILER_FLAGS’,‘cl0’)dnl according to documentation, not used wit
EXPOSED_USER(‘root’)dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(‘Port=smtp,Addr=127.0.0.1, Name=MTA’)dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can’t reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected find
dnl # this useful.
dnl #
dnl DAEMON_OPTIONS(‘Port=submission, Name=MSA, M=Ea’)dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587 followed
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express can’t
dnl # do STARTTLS on ports other than 25. Mozilla Mail can ONLY use STARTTLS
dnl # and doesn’t support the deprecated smtps; Evolution <1.1.1 uses smtps
dnl # when SSL is enabled-- STARTTLS support is available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(‘Port=smtps, Name=TLSMTA, M=s’)dnl
dnl #
dnl # The following causes sendmail to additionally listen on the IPv6 loopback
dnl # device. Remove the loopback address restriction listen to the network.
dnl #
dnl DAEMON_OPTIONS(‘port=smtp,Addr=::1, Name=MTA-v6, Family=inet6’)dnl
dnl #
dnl # enable both ipv6 and ipv4 in sendmail:
dnl #
dnl DAEMON_OPTIONS(‘Name=MTA-v4, Family=inet, Name=MTA-v6, Family=inet6’)
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you want to
dnl # protect yourself from spam. However, the laptop and users on computers
dnl # that do not have 24x7 DNS do need this.
dnl #
FEATURE(‘accept_unresolvable_domains’)dnl
dnl #
dnl FEATURE(‘relay_based_on_MX’)dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
```

```
LOCAL_DOMAIN(‘localhost.localdomain’)dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
dnl MASQUERADE_AS(‘mydomain.com’)dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
dnl FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
dnl
```

```
dnl
dnl define(‘QUEUE_DIR’, ‘/var/spool/mqueue’)dnl
QUEUE_GROUP(‘mqueue’, ‘P=/var/spool/mqueue, F=f, r=1, R=8, I=2m’)
dnl
dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
LOCAL_RULE_0
R $* < @ $* .virtuser. > $*                        $#procmail $@ $2 $: $1
```

☞ An example of /etc/mail/sendmail.cf:

```
#
# Copyright (c) 1998-2003 Sendmail, Inc. and its suppliers.
#        All rights reserved.
# Copyright (c) 1983, 1995 Eric P. Allman.  All rights reserved.
# Copyright (c) 1988, 1993
#        The Regents of the University of California.  All rights reserved.
#
```

```
# By using this file, you agree to the terms and conditions set
# forth in the LICENSE file which can be found at the top level of
# the sendmail distribution.
#
#

######################################################################
######################################################################
#####
#####                SENDMAIL CONFIGURATION FILE
#####
##### built by root@sophie.cs.fsu.edu on Mon Nov 7 09:02:23 EST 2005
##### in /etc/mail
##### using /usr/share/sendmail-cf/ as configuration include directory
#####
######################################################################
#####
#####        DO NOT EDIT THIS FILE!  Only edit the source .mc file.
#####
######################################################################
```

```
######################################################################

#####  $Id: cfhead.m4,v 8.108.2.6 2003/12/05 02:26:47 ca Exp $  #####
#####  $Id: cf.m4,v 8.32 1999/02/07 07:26:14 gshapiro Exp $  #####
#####  setup for Red Hat Linux  #####
#####  $Id: linux.m4,v 8.13 2000/09/17 17:30:00 gshapiro Exp $  #####



#####  $Id: local_procmail.m4,v 8.21.42.1 2002/11/17 04:25:07 ca Exp $  #####


#####  $Id: no_default_msa.m4,v 8.2 2001/02/14 05:03:22 gshapiro Exp $  #####

#####  $Id: smrsh.m4,v 8.14 1999/11/18 05:06:23 ca Exp $  #####

#####  $Id: mailertable.m4,v 8.23 2001/03/16 00:51:26 gshapiro Exp $  #####

#####  $Id: redirect.m4,v 8.15 1999/08/06 01:47:36 gshapiro Exp $  #####
```

```
#####  $Id: always_add_domain.m4,v 8.11 2000/09/12 22:00:53 ca Exp $  #####

#####  $Id: use_ct_file.m4,v 8.11 2001/08/26 20:58:57 gshapiro Exp $  #####


#####  $Id: virtusertable.m4,v 8.21 2001/03/16 00:51:26 gshapiro Exp $  #####

#####  $Id: always_add_domain.m4,v 8.11 2000/09/12 22:00:53 ca Exp $  #####

#####  $Id: use_cw_file.m4,v 8.11 2001/08/26 20:58:57 gshapiro Exp $  #####


#####  $Id: local_procmail.m4,v 8.21.42.1 2002/11/17 04:25:07 ca Exp $  #####

#####  $Id: access_db.m4,v 8.24 2002/03/06 21:50:25 ca Exp $  #####

#####  $Id: blacklist_recipients.m4,v 8.13 1999/04/02 02:25:13 gshapiro Exp $  ##

#####  $Id: accept_unresolvable_domains.m4,v 8.10 1999/02/07 07:26:07 gshapiro Ex
```

```
#####  $Id: proto.m4,v 8.649.2.30 2004/01/11 17:54:06 ca Exp $  #####

# level 10 config file format
V10/Berkeley

# override file safeties - setting this option compromises system security,
# addressing the actual file configuration problem is preferred
# need to set this before any file actions are encountered in the cf file
#O DontBlameSendmail=safe

# default LDAP map specification
# need to set this now before any LDAP maps are defined
#O LDAPDefaultSpec=-h localhost

##################
#   local info   #
##################
```

```
# my LDAP cluster
# need to set this before any LDAP lookups are done (including classes)
#D{sendmailMTACluster}$m

Cwlocalhost
# file containing names of hosts for which we receive email
Fw/etc/mail/local-host-names

# my official domain name
# ... define this only if sendmail cannot automatically determine your domain
#Dj$w.Foo.COM

# host/domain names ending with a token in class P are canonical
CP.

# "Smart" relay host (may be null)
DS


# operators that cannot be in local usernames (i.e., network indicators)
```

```
CO @ % !

# a class with just dot (for identifying canonical names)
C..

# a class with just a left bracket (for identifying domain literals)
C[[

# access_db acceptance class
C{Accept}OK RELAY


C{ResOk}OKR


# Hosts for which relaying is permitted ($=R)
FR-o /etc/mail/relay-domains

# arithmetic map
Karith arith
```

```
# macro storage map
Kmacro macro
# possible values for TLS_connection in access map
C{tls}VERIFY ENCR
```

```
# dequoting map
Kdequote dequote

# class E: names that should be exposed as from this host, even if we masquerade
# class L: names that should be delivered locally, even if we have a relay
# class M: domains that should be converted to $M
# class N: domains that should not be converted to $M
#CL root
F{VirtHost}-o /etc/mail/virtuserdomains

C{E}root
```

```
C{w}localhost.localdomain
CR$={VirtHost}



# my name for error messages
DnMAILER-DAEMON


# Mailer table (overriding domains)
Kmailertable hash -o /etc/mail/mailertable.db

CPREDIRECT

# Virtual user table (maps incoming users)
Kvirtuser hash -o /etc/mail/virtusertable.db

# Access list database (for spam stomping)
Kaccess hash -T<TMPF> -o /etc/mail/access.db
```

```
# Configuration version number
DZ8.12.11


###############
#   Options   #
###############

# strip message body to 7 bits on input?
O SevenBitInput=False

# 8-bit data handling
#O EightBitMode=pass8

# wait for alias file rebuild (default units: minutes)
O AliasWait=10

# location of alias file
O AliasFile=/etc/aliases
```

```
# minimum number of free blocks on filesystem
O MinFreeBlocks=100

# maximum message size
#O MaxMessageSize=1000000

# substitution for space (blank) characters
O BlankSub=.

# avoid connecting to "expensive" mailers on initial submission?
O HoldExpensive=False

# checkpoint queue runs after every N successful deliveries
#O CheckpointInterval=10

# default delivery mode
O DeliveryMode=background

# error message header/file
#O ErrorHeader=/etc/mail/error-header
```

```
# error mode
#O ErrorMode=print

# save Unix-style "From_" lines at top of header?
#O SaveFromLine=False

# queue file mode (qf files)
#O QueueFileMode=0600

# temporary file mode
O TempFileMode=0600

# match recipients against GECOS field?
#O MatchGECOS=False

# maximum hop count
#O MaxHopCount=25

# location of help file
```

```
O HelpFile=/etc/mail/helpfile

# ignore dots as terminators in incoming messages?
#O IgnoreDots=False

# name resolver options
#O ResolverOptions=+AAONLY

# deliver MIME-encapsulated error messages?
O SendMimeErrors=True

# Forward file search path
O ForwardPath=$z/.forward.$w:$z/.forward

# open connection cache size
O ConnectionCacheSize=2

# open connection cache timeout
O ConnectionCacheTimeout=5m
```

```
# persistent host status directory
#O HostStatusDirectory=.hoststat

# single thread deliveries (requires HostStatusDirectory)?
#O SingleThreadDelivery=False

# use Errors-To: header?
O UseErrorsTo=False

# log level
O LogLevel=9

# send to me too, even in an alias expansion?
#O MeToo=True

# verify RHS in newaliases?
O CheckAliases=False

# default messages to old style headers if no special punctuation?
O OldStyleHeaders=True
```

```
# SMTP daemon options

O DaemonPortOptions=Port=smtp,Addr=127.0.0.1, Name=MTA

# SMTP client options
#O ClientPortOptions=Family=inet, Address=0.0.0.0

# Modifiers to define {daemon_flags} for direct submissions
#O DirectSubmissionModifiers

# Use as mail submission program? See sendmail/SECURITY
#O UseMSP

# privacy flags
O PrivacyOptions=authwarnings,novrfy,noexpn,restrictqrun

# who (if anyone) should get extra copies of error messages
#O PostmasterCopy=Postmaster
```

```
# slope of queue-only function
#O QueueFactor=600000

# limit on number of concurrent queue runners
#O MaxQueueChildren

# maximum number of queue-runners per queue-grouping with multiple queues
#O MaxRunnersPerQueue=1

# priority of queue runners (nice(3))
#O NiceQueueRun

# shall we sort the queue by hostname first?
#O QueueSortOrder=priority

# minimum time in queue before retry
#O MinQueueAge=30m

# how many jobs can you process in the queue?
#O MaxQueueRunSize=10000
```

```
# perform initial split of envelope without checking MX records
#O FastSplit=1

# queue directory
O QueueDirectory=/var/spool/mqueue

# key for shared memory; 0 to turn off
#O SharedMemoryKey=0



# timeouts (many of these)
#O Timeout.initial=5m
O Timeout.connect=1m
#O Timeout.aconnect=0s
#O Timeout.iconnect=5m
#O Timeout.helo=5m
#O Timeout.mail=10m
#O Timeout.rcpt=1h
```

```
#O Timeout.datainit=5m
#O Timeout.datablock=1h
#O Timeout.datafinal=1h
#O Timeout.rset=5m
#O Timeout.quit=2m
#O Timeout.misc=2m
#O Timeout.command=1h
O Timeout.ident=0
#O Timeout.fileopen=60s
#O Timeout.control=2m
O Timeout.queuereturn=5d
#O Timeout.queuereturn.normal=5d
#O Timeout.queuereturn.urgent=2d
#O Timeout.queuereturn.non-urgent=7d

O Timeout.queuewarn=4h
#O Timeout.queuewarn.normal=4h
#O Timeout.queuewarn.urgent=1h
#O Timeout.queuewarn.non-urgent=12h
```

```
#O Timeout.hoststatus=30m
#O Timeout.resolver.retrans=5s
#O Timeout.resolver.retrans.first=5s
#O Timeout.resolver.retrans.normal=5s
#O Timeout.resolver.retry=4
#O Timeout.resolver.retry.first=4
#O Timeout.resolver.retry.normal=4
#O Timeout.lhlo=2m
#O Timeout.auth=10m
#O Timeout.starttls=1h

# time for DeliverBy; extension disabled if less than 0
#O DeliverByMin=0

# should we not prune routes in route-addr syntax addresses?
#O DontPruneRoutes=False

# queue up everything before forking?
O SuperSafe=True
```

```
# status file
O StatusFile=/var/log/mail/statistics

# time zone handling:
#  if undefined, use system default
#  if defined but null, use TZ envariable passed in
#  if defined and non-null, use that info
#O TimeZoneSpec=

# default UID (can be username or userid:groupid)
O DefaultUser=8:12

# list of locations of user database file (null means no lookup)
O UserDatabaseSpec=/etc/mail/userdb.db

# fallback MX host
#O FallbackMXhost=fall.back.host.net

# if we are the best MX host for a site, try it directly instead of config err
O TryNullMXList=true
```

```
# load average at which we just queue messages
#O QueueLA=8

# load average at which we refuse connections
#O RefuseLA=12

# load average at which we delay connections; 0 means no limit
#O DelayLA=0

# maximum number of children we allow at one time
#O MaxDaemonChildren=0

# maximum number of new connections per second
#O ConnectionRateThrottle=0

# work recipient factor
#O RecipientFactor=30000

# deliver each queued job in a separate process?
```

```
#O ForkEachJob=False

# work class factor
#O ClassFactor=1800

# work time factor
#O RetryFactor=90000

# default character set
#O DefaultCharSet=iso-8859-1

# service switch file (name hardwired on Solaris, Ultrix, OSF/1, others)
#O ServiceSwitchFile=/etc/mail/service.switch

# hosts file (normally /etc/hosts)
#O HostsFile=/etc/hosts

# dialup line delay on connection failure
#O DialDelay=10s
```

```
# action to take if there are no recipients in the message
#O NoRecipientAction=add-to-undisclosed

# chrooted environment for writing to files
#O SafeFileEnvironment=/arch

# are colons OK in addresses?
#O ColonOkInAddr=True

# shall I avoid expanding CNAMEs (violates protocols)?
#O DontExpandCnames=False

# SMTP initial login message (old $e macro)
O SmtpGreetingMessage=$j Sendmail $v/$Z; $b

# UNIX initial From header format (old $l macro)
O UnixFromLine=From $g $d

# From: lines that have embedded newlines are unwrapped onto one line
#O SingleLineFromHeader=False
```

```
# Allow HELO SMTP command that does not include a host name
#O AllowBogusHELO=False

# Characters to be quoted in a full name phrase (@,;:\()[] are automatic)
#O MustQuoteChars=.

# delimiter (operator) characters (old $o macro)
O OperatorChars=.:%@!^/[]+

# shall I avoid calling initgroups(3) because of high NIS costs?
#O DontInitGroups=False

# are group-writable :include: and .forward files (un)trustworthy?
# True (the default) means they are not trustworthy.
#O UnsafeGroupWrites=True


# where do errors that occur when sending errors get sent?
#O DoubleBounceAddress=postmaster
```

```
# where to save bounces if all else fails
#O DeadLetterDrop=/var/tmp/dead.letter

# what user id do we assume for the majority of the processing?
#O RunAsUser=sendmail

# maximum number of recipients per SMTP envelope
#O MaxRecipientsPerMessage=100

# limit the rate recipients per SMTP envelope are accepted
# once the threshold number of recipients have been rejected
#O BadRcptThrottle=20

# shall we get local names from our installed interfaces?
O DontProbeInterfaces=true

# Return-Receipt-To: header implies DSN request
#O RrtImpliesDsn=False
```

```
# override connection address (for testing)
#O ConnectOnlyTo=0.0.0.0

# Trusted user for file ownership and starting the daemon
#O TrustedUser=root

# Control socket for daemon management
#O ControlSocketName=/var/spool/mqueue/.control

# Maximum MIME header length to protect MUAs
#O MaxMimeHeaderLength=2048/1024

# Maximum length of the sum of all headers
#O MaxHeadersLength=32768

# Maximum depth of alias recursion
#O MaxAliasRecursion=10

# location of pid file
#O PidFile=/var/run/sendmail.pid
```

```
# Prefix string for the process title shown on ’ps’ listings
#O ProcessTitlePrefix=prefix

# Data file (df) memory-buffer file maximum size
#O DataFileBufferSize=4096

 [  .... SKIPPED MATERIAL .... ]

############################
# QUEUE GROUP DEFINITIONS  #
############################

Qmqueue, P=/var/spool/mqueue, F=f, r=1, R=8, I=2m

###########################
#   Message precedences   #
###########################

Pfirst-class=0
```

```
Pspecial-delivery=100
Plist=-30
Pbulk=-60
Pjunk=-100

#####################
#   Trusted users   #
#####################

# this is equivalent to setting class "t"
Ft/etc/mail/trusted-users
Troot
Tdaemon
Tuucp

#########################
#   Format of headers   #
#########################

H?P?Return-Path: <$g>
```

```
HReceived: $?sfrom $s $.$?_($?s$|from $.$_)
        $.$?{auth_type}(authenticated$?{auth_ssf} bits=${auth_ssf}$.)
        $.by $j ($v/$Z)$?r with $r$. id $i$?{tls_version}
        (version=${tls_version} cipher=${cipher} bits=${cipher_bits} verify=${ver
        for $u; $|;
        $.$b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $?x$x <$g>$|$g$.
H?F?From: $?x$x <$g>$|$g$.
H?x?Full-Name: $x
# HPosted-Date: $a
# H?l?Received-Date: $b
H?M?Resent-Message-Id: <$t.$i@$j>
H?M?Message-Id: <$t.$i@$j>

#
######################################################################
######################################################################
#####
```

```
#####                        REWRITING RULES
#####
######################################################################
######################################################################

############################################
###  Ruleset 3 -- Name Canonicalization  ###
############################################
Scanonify=3

# handle null input (translate to <@> special case)
R$@                        $@ <@>

# strip group: syntax (not inside angle brackets!) and trailing semicolon
R$*                        $: $1 <@>                        mark addresses
R$* < $* > $* <@>        $: $1 < $2 > $3                         unmark <addr>
R@ $* <@>                $: @ $1                                 unmark @host:...
R$* [ IPv6 : $+ ] <@>        $: $1 [ IPv6 : $2 ]                 unmark IPv6 addr
R$* :: $* <@>                $: $1 :: $2                         unmark node::addr
R:include: $* <@>        $: :include: $1                         unmark :include:
```

```
R$* : $* [ $* ]                $: $1 : $2 [ $3 ] <@>                remark if lea
R$* : $* <@>                $: $2                                strip colon if m
R$* <@>                        $: $1                                unmark
R$* ;                           $1                                strip trailing
R$* < $+ :; > $*        $@ $2 :; <@>                        catch <list:;>
R$* < $* ; >                   $1 < $2 >                        bogus bracketed s

# null input now results from list:; syntax
R$@                        $@ :; <@>

# strip angle brackets -- note RFC733 heuristic to get innermost item
R$*                        $: < $1 >                        housekeeping <>
R$+ < $* >                   < $2 >                        strip excess on left
R< $* > $+                   < $1 >                        strip excess on right
R<>                        $@ < @ >                        MAIL FROM:<> case
R< $+ >                        $: $1                                remove housek

 [ .... SKIPPED MATERIAL .... ]

######################################
```

```
###   Ruleset 0 -- Parse Address   ###
######################################

Sparse=0

R$*                        $: $>Parse0 $1                initial parsing
R<@>                        $#local $: <@>                special case error msgs
R$*                        $: $>ParseLocal $1        handle local hacks
R$*                        $: $>Parse1 $1                final parsing

#
#  Parse0 -- do initial syntax checking and eliminate local addresses.
#        This should either return with the (possibly modified) input
#        or return with a #error mailer.  It should not return with a
#        #mailer other than the #error mailer.
#

SParse0
R<@>                        $@ <@>                        special case error msgs
R$* : $* ; <@>                $#error $@ 5.1.3 $: "553 List:; syntax illegal for
```

```
R@ <@ $* >                < @ $1 >                catch "@@host" bogosity
R<@ $+>                        $#error $@ 5.1.3 $: "553 User address required"
R$+ <@>                        $#error $@ 5.1.3 $: "553 Hostname required"
R$*                        $: <> $1
R<> $* < @ [ $* ] : $+ > $*        $1 < @ [ $2 ] : $3 > $4
R<> $* < @ [ $* ] , $+ > $*        $1 < @ [ $2 ] , $3 > $4
R<> $* < @ [ $* ] $+ > $*        $#error $@ 5.1.2 $: "553 Invalid address"
R<> $* < @ [ $+ ] > $*                $1 < @ [ $2 ] > $3
R<> $* <$* : $* > $*        $#error $@ 5.1.3 $: "553 Colon illegal in host name p
R<> $*                        $1
R$* < @ . $* > $*        $#error $@ 5.1.2 $: "553 Invalid host name"
R$* < @ $* .. $* > $*        $#error $@ 5.1.2 $: "553 Invalid host name"
R$* < @ $* @ > $*        $#error $@ 5.1.2 $: "553 Invalid route address"
R$* @ $* < @ $* > $*        $#error $@ 5.1.3 $: "553 Invalid route address"
R$* , $~O $*                $#error $@ 5.1.3 $: "553 Invalid route address"


# now delete the local info -- note $=O to find characters that cause forwarding
R$* < @ > $*                $@ $>Parse0 $>canonify $1        user@ => user
R< @ $=w . > : $*        $@ $>Parse0 $>canonify $2        @here:... -> ...
```

```
R$- < @ $=w . >                 $: $(dequote $1 $) < @ $2 . >         dequote "foo'
R< @ $+ >                 $#error $@ 5.1.3 $: "553 User address required"
R$* $=O $* < @ $=w . >         $@ $>Parse0 $>canonify $1 $2 $3         ...@here ->
R$-                         $: $(dequote $1 $) < @ *LOCAL* >         dequote "foo'
R< @ *LOCAL* >                 $#error $@ 5.1.3 $: "553 User address required"
R$* $=O $* < @ *LOCAL* >
                          $@ $>Parse0 $>canonify $1 $2 $3         ...@*LOCAL* -> ..

R$* < @ *LOCAL* >          $: $1

#
#  Parse1 -- the bottom half of ruleset 0.
#

SParse1

# handle numeric address spec
R$* < @ [ $+ ] > $*         $: $>ParseLocal $1 < @ [ $2 ] > $3         numeric inte
R$* < @ [ $+ ] > $*         $: $1 < @ [ $2 ] : $S > $3         Add smart host to pa
R$* < @ [ $+ ] : > $*                 $#esmtp $@ [$2] $: $1 < @ [$2] > $3         r
R$* < @ [ $+ ] : $- : $*> $*         $#$3 $@ $4 $: $1 < @ [$2] > $5         smartho
```

```
R$* < @ [ $+ ] : $+ > $*        $#esmtp $@ $3 $: $1 < @ [$2] > $4       smarthos

# handle virtual users
R$+                        $: <!> $1                Mark for lookup
R<!> $+ < @ $={VirtHost} . >         $: < $(virtuser $1 @ $2 $@ $1 $: @ $) > $1 <
R<!> $+ < @ $=w . >         $: < $(virtuser $1 @ $2 $@ $1 $: @ $) > $1 < @ $2 . >
R<@> $+ + $+ < @ $* . >
                        $: < $(virtuser $1 + + @ $3 $@ $1 $@ $2 $@ +$2 $: @ $) >
R<@> $+ + $* < @ $* . >
                        $: < $(virtuser $1 + * @ $3 $@ $1 $@ $2 $@ +$2 $: @ $) >
R<@> $+ + $* < @ $* . >
                        $: < $(virtuser $1 @ $3 $@ $1 $@ $2 $@ +$2 $: @ $) > $1 +
R<@> $+ + $+ < @ $+ . >        $: < $(virtuser + + @ $3 $@ $1 $@ $2 $@ +$2 $: @ $
R<@> $+ + $* < @ $+ . >        $: < $(virtuser + * @ $3 $@ $1 $@ $2 $@ +$2 $: @ $
R<@> $+ + $* < @ $+ . >        $: < $(virtuser @ $3 $@ $1 $@ $2 $@ +$2 $: ! $) >
R<@> $+ < @ $+ . >        $: < $(virtuser @ $2 $@ $1 $: @ $) > $1 < @ $2 . >
R<@> $+                        $: $1
R<!> $+                        $: $1
R< error : $-.$-.$- : $+ > $*         $#error $@ $1.$2.$3 $: $4
R< error : $- $+ > $*         $#error $@ $(dequote $1 $) $: $2
```

```
R< $+ > $+ < @ $+ >        $: $>Recurse $1

# short circuit local delivery so forwarded email works


R$=L < @ $=w . >        $#local $: @ $1                        special local name
R$+ < @ $=w . >                $#local $: $1                        regular local

# not local -- try mailer table lookup
R$* <@ $+ > $*                $: < $2 > $1 < @ $2 > $3        extract host name
R< $+ . > $*                $: < $1 > $2                        strip trailing do
R< $+ > $*                $: < $(mailertable $1 $) > $2        lookup
R< $~[ : $* > $*         $>MailerToTriple < $1 : $2 > $3                check --
R< $+ > $*                $: $>Mailertable <$1> $2                try domain

# resolve remotely connected UUCP links (if any)

# resolve fake top level domains by forwarding to other hosts
# pass names that still have a host to a smarthost (if defined)
R$* < @ $* > $*                $: $>MailerToTriple < $S > $1 < @ $2 > $3        g

# deal with other remote names
R$* < @$* > $*                $#esmtp $@ $2 $: $1 < @ $2 > $3        user@host.do

# handle locally delivered names
R$=L                        $#local $: @ $1                special local names
R$+                        $#local $: $1                        regular local nam

[ .... SKIPPED MATERIAL .... ]

###################################################################
###  Ruleset 98 -- local part of ruleset zero (can be null)       ###
###################################################################

SParseLocal=98

# addresses sent to foo@host.REDIRECT will give a 551 error code
```

```
R$* < @ $+ .REDIRECT. >                $: $1 < @ $2 . REDIRECT . > < ${opMode} >
R$* < @ $+ .REDIRECT. > <i>        $: $1 < @ $2 . REDIRECT. >
R$* < @ $+ .REDIRECT. > < $- >        $#error $@ 5.1.1 $: "551 User has moved; pl


R $* < @ $* .virtuser. > $*                        $#procmail $@ $2 $: $1


[ .... LOTS OF SKIPPED MATERIAL .... ]

#
######################################################################
######################################################################
#####
#####                        MAILER DEFINITIONS
#####
######################################################################
######################################################################

#####################################
```

```
###   SMTP Mailer specification   ###
#####################################

#####  $Id: smtp.m4,v 8.64 2001/04/03 01:52:54 gshapiro Exp $  #####

#
#  common sender and masquerading recipient rewriting
#
SMasqSMTP
R$* < @ $* > $*                $@ $1 < @ $2 > $3                already fully qua
R$+                        $@ $1 < @ *LOCAL* >                add local qualifica

#
#  convert pseudo-domain addresses to real domain addresses
#
SPseudoToReal

# pass <route-addr>s through
R< @ $+ > $*                $@ < @ $1 > $2                      resolve <route-
```

```
# output fake domains as user%fake@relay

# do UUCP heuristics; note that these are shared with UUCP mailers
R$+ < @ $+ .UUCP. >        $: < $2 ! > $1                        convert to UUCP
R$+ < @ $* > $*                $@ $1 < @ $2 > $3                not UUCP form

# leave these in .UUCP form to avoid further tampering
R< $&h ! > $- ! $+        $@ $2 < @ $1 .UUCP. >
R< $&h ! > $-.$+ ! $+        $@ $3 < @ $1.$2 >
R< $&h ! > $+                $@ $1 < @ $&h .UUCP. >
R< $+ ! > $+                $: $1 ! $2 < @ $Y >                use UUCP_RELAY
R$+ < @ $~[ $* : $+ >        $@ $1 < @ $4 >                        strip mailer:
R$+ < @ >                $: $1 < @ *LOCAL* >                if no UUCP_RELAY


#
#  envelope sender rewriting
#
SEnvFromSMTP
R$+                        $: $>PseudoToReal $1                sender/recipient c
```

```
R$* :; <@>                $@                                list:; special case
R$*                        $: $>MasqSMTP $1                qualify unqual'ed name
R$+                        $: $>MasqEnv $1                        do masquerading


#
#  envelope recipient rewriting --
#  also header recipient if not masquerading recipients
#
SEnvToSMTP
R$+                        $: $>PseudoToReal $1                sender/recipient c
R$+                        $: $>MasqSMTP $1                qualify unqual'ed name
R$* < @ *LOCAL* > $*        $: $1 < @ $j . > $2

#
#  header sender and masquerading header recipient rewriting
#
SHdrFromSMTP
R$+                        $: $>PseudoToReal $1                sender/recipient c
R:; <@>                        $@                                list:; special c
```

```
# do special header rewriting
R$* <@> $*                $@ $1 <@> $2                        pass null host thro
R< @ $* > $*                $@ < @ $1 > $2                        pass route-addr
R$*                        $: $>MasqSMTP $1                qualify unqual’ed name
R$+                        $: $>MasqHdr $1                        do masquerading


#
#  relay mailer header masquerading recipient rewriting
#
SMasqRelay
R$+                        $: $>MasqSMTP $1
R$+                        $: $>MasqHdr $1

Msmtp,                      P=[IPC], F=mDFMuX, S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMTP,
                T=DNS/RFC822/SMTP,
                A=TCP $h
Mesmtp,                      P=[IPC], F=mDFMuXa, S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMTP
                T=DNS/RFC822/SMTP,
                A=TCP $h
Msmtp8,                P=[IPC], F=mDFMuX8, S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMT
                T=DNS/RFC822/SMTP,
                A=TCP $h
Mdsmtp,                P=[IPC], F=mDFMuXa%, S=EnvFromSMTP/HdrFromSMTP, R=EnvToSM
                T=DNS/RFC822/SMTP,
                A=TCP $h
Mrelay,                P=[IPC], F=mDFMuXa8, S=EnvFromSMTP/HdrFromSMTP, R=MasqSMT
                T=DNS/RFC822/SMTP,
                A=TCP $h


######################*****##############
###   PROCMAIL Mailer specification   ###
##################*****##################

#####  $Id: procmail.m4,v 8.22 2001/11/12 23:11:34 ca Exp $  #####

Mprocmail,        P=/usr/bin/procmail, U=vmail:vmail, F=DFMcl0,
                S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMTP/HdrFromSMTP,
                T=DNS/RFC822/X-Unix,
                A=procmail -t -Y -a $h -a $u


##################################################
###   Local and Program Mailer specification   ###
##################################################

#####  $Id: local.m4,v 8.58 2000/10/26 01:58:29 ca Exp $  #####

#
#  Envelope sender rewriting
#
SEnvFromL
R<@>                        $n                        errors to mailer-daemon
R@ <@ $*>                $n                        temporarily bypass Sun bogosit
R$+                        $: $>AddDomain $1        add local domain if needed
R$*                        $: $>MasqEnv $1                do masquerading

#
```

```
#  Envelope recipient rewriting
#
SEnvToL
R$+ < @ $* >                $: $1                            strip host part

#
#  Header sender rewriting
#
SHdrFromL
R<@>                        $n                          errors to mailer-daemon
R@ <@ $*>                $n                          temporarily bypass Sun bogosit
R$+                        $: $>AddDomain $1        add local domain if needed
R$*                        $: $>MasqHdr $1                do masquerading

#
#  Header recipient rewriting
#
SHdrToL
R$+                        $: $>AddDomain $1        add local domain if needed
R$* < @ *LOCAL* > $*        $: $1 < @ $j . > $2
```

```
#
#  Common code to add local domain name (only if always-add-domain)
#
SAddDomain
R$* < @ $* > $*         $@ $1 < @ $2 > $3        already fully qualified

R$+                        $@ $1 < @ *LOCAL* >        add local qualification

Mlocal,                P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9, S=EnvFromL/Hdr
                T=DNS/RFC822/X-Unix,
                A=procmail -t -Y -a $h -d $u
Mprog,                P=/usr/sbin/smrsh, F=lsDFMoqeu9, S=EnvFromL/HdrFromL, R=Env
                T=X-Unix/X-Unix/X-Unix,
                A=smrsh -c $u
```

☞ sendmail is quite powerful. A common application for sendmail is to serve as a gateway mail server.

(You can also do this type of thing with Exchange; see Microsoft's website for a document called "Using a Windows SMTP Relay Server in a Perimeter Network" which gives an overview, and for details, look at "How to Configure a Windows Server 2003 Server as a Relay Server or Smart Host".)

☞ One quite clever idea came from MailScanner's author, Julian Field at the University of Southampton. Email going into sendmail is put into a queue, and instead of the usual process of another sendmail process acting as a queue handler to deliver it, MailScanner first processes the mail (looking for spam and viruses, and comparing it against blacklists and whitelists), and then enqueues the message into a different queue directory for the second sendmail queue handler to find. (You can often view mail queues with the alias "mailq" which actually is "sendmail -bp" (or postfix's "postqueue -f".)

☞ As we saw from the .mc files, sendmail doesn't actually do local delivery of email. Ordinary delivery is typically by procmail (other candidates include the old **binmail** program or **dropmail**.

☞ **procmail** is a very powerful mail delivery agent; it can be configured to do many, many things. See http://www.procmail.org for “recipes”. For instance, a typical **procmail** recipe might look like:

```
:0
* ^From: unpleasant@user\.com
/dev/null

:0:
${DEFAULT}
```

- ☞ Headsup: **procmail** is very picky about such items as colons. A single missing colon can be very bad since it might be one that indicates that a mailbox is to be locked before it receives a delivery – and failing to lock a shared mailbox file might prove unpleasant.

- ☞ Finally, you have to decide one (or perhaps two more) things about delivery: do you want email to go into a traditional mbox, which is just one long file of email separated by the delimiter "\nFrom .*\n" or do you want to use the more modern maildir approach, where each email is written to a separate file? The latter is preferred. If you do choose to go with mbox format, you will also have to make sure that your locking mechanisms for procmail, imap/pop, and any other client software such as openwebmail all agree to a common locking mechanism.

Main SMTP commands

- HELO / EHLO
- MAIL FROM: $< `someone@somewhere` >$
- RCPT TO: $< `someone@somewhere` >$
- DATA
- QUIT

Maildirs, from Dr. Bernstein (see http://cr.yp.to/proto/maildir.html)

- ☞ Maildirs are safer in many ways that the traditional mbox format. On USAH p. 549, the problems with traditional mailbox locking are discussed, as they are on the maildir webpage.

- ☞ Maildirs keep every email message in a separate file, and never use any type of locking mechanism.

- Traditional mailbox (mbox) format is not safe over NFS, even nowadays.

- Every maildir setup will have the subdirectories `tmp`, `new`, and `cur`, and may have others. Mail is first delivered to `tmp`, then safely moved to `new`. It may have others, also.

- Here's a good description from the **qmail** man page for Maildirs:

```
     HOW A MESSAGE IS DELIVERED

          The tmp directory is used to ensure reliable delivery, as
```

discussed here.

A program delivers a mail message in six steps. First, it chdir()s to the maildir directory. Second, it stat()s the name tmp/time.pid.host, where time is the number of seconds since the beginning of 1970 GMT, pid is the program's process ID, and host is the host name. Third, if stat() returned anything other than ENOENT, the program sleeps for two seconds, updates time, and tries the stat() again, a limited number of times. Fourth, the program creates tmp/time.pid.host. Fifth, the program NFS-writes the message to the file. Sixth, the program link()s the file to new/time.pid.host. At that instant the message has been successfully delivered.

[...]

NFS-writing means (1) as usual, checking the number of bytes returned from each write() call; (2) calling fsync() and checking its return value; (3) calling close() and checking its return value. (Standard NFS implementations handle fsync() incorrectly but make up for it by abusing close().)

imap and pop

- ☞ **dovecot**: an increasingly popular imap and pop server is **dovecot**, which handles mbox and maildir format with aplomb. It also handles virtual users quite well, including those existing only in databases.
- ☞ **courier**: also popular.
- ☞ **cyrus**: uses its own mailbox format; it is more formidable to configure than other imap setups.

☞ What is imap/pop? These are protocols that allow a user to remotely retrieve email from a mailhost. imap (RFC 3501), unlike pop (RFC 1939), supports the idea of separate folders on the server machine, and it has more functionality built in. Generally, you leave your mail messages on an imap server, and you retrieve them from a pop server.

☞ The main commands for POP are

➳ USER username

➳ PASS password

  - LIST
  - RETR item
  - DELE item
  - QUIT
  - RSET

- IMAP commands are “tagged”. This means that you need to put a short, unique identifier before you use a command; the response to that command will use the same tag. The main commands for IMAP checking are

  tag LOGIN username password

```
tag SELECT mailbox
tag LIST "" *
tag LOGOUT
```

Clients

- ☞ There are two types of clients: (1) those that read email via a protocol such as IMAP, POP, or the "Microsoft" way, and (2) those that access mail via a filesystem.

- ☞ Web clients: The very popular squirrelmail (http://www.squirrelmail.org) is an example of type (1) that uses IMAP. openwebmail (http://www.openwebmail.or is an example of (2). It reads directly from either MBOX or Maildir format.

- ☞ Dedicated interface clients: most of these now handle both file stores and IMAP/POP. Examples include Outlook, Thunderbird, Evolution, Sylpheed, Eudora, Pegasus, and a host of others.

- ☞ Working on the latter setups can be interesting since the client can silently be going to entirely different machines also for its email.

- ☞ I have worked on a setup where just determining where the client email was coming from required using **tcpdump** and lots of patience; in that case, a single user was having a problem accessing his mailbox: it turned out that the client interface (a very old version of a web email client) could not handle bad headers in email messages; it could not handle very large messages; and it was configured to terminate any handler that took longer than 30 seconds, so it could not ever handle a mailbox that had a large number of messages to move – it used POP instead of IMAP, and thus ended up initially doing RETR, then DELE after it had pulled the messages into a maildir-like format.