



OpenMail Technical Fundamentals

The 300-level Essentials

Richi Jennings, Technical Product Manager
Hewlett-Packard
Nine Mile Ride
Wokingham
Berkshire RG40 3LL
United Kingdom

T +44 (0)1 344-36 5870
F +44 (0)1 344-76 3429
richi@hp.com

OpenMail®



Introduction/Agenda

OpenMail

- **This is a broad technical introduction to OpenMail**
- **Covers some (but not all) essential fundamentals that architects and sysadmins need to know**
- **Broad brush taster (deep in some areas, *not* a substitute for RTFM, 200- and 300-level training!)**
- **Architectural overview**
- **OpenMail concepts**
- **How to provide an OpenMail service**
- **What to do, what *not* to do**

UNIX is a registered trademark of The Open Group.
Microsoft is a U.S. registered trademark of Microsoft Corp.
Windows NT is a U.S. registered trademark of Microsoft Corp.



OpenMail is...

- OpenMail
- ◆ Scalable
 - ◆ Flexible
 - ◆ Reliable
 - ◆ Low-cost
 - ◆ Feature-rich
 - ◆ Future-proof

... *Why?*



OpenMail "Architecture"

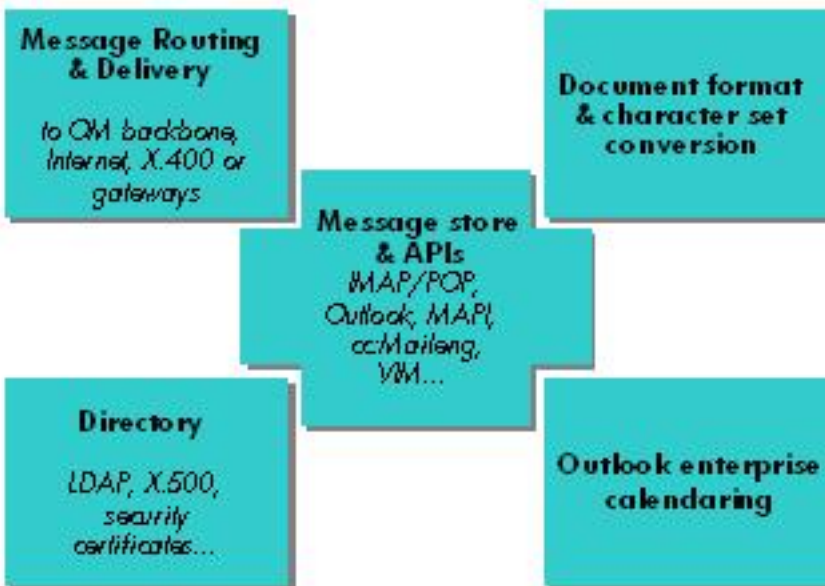
OpenMail[®]





OpenMail "Architecture"

OpenMail[®]





Message Store

6

openMail

- ◆ Single message store for all types of client
- ◆ Superset of Internet and X.400
- ◆ Flexible, controllable synchronization of public folders
- ◆ Highly scalable
- ◆ Access control lists on public folders
- ◆ Per-user store size quotas
- ◆ Data migration (cc:Mail, MS-Mail, Exchange, PROFS, HP DeskManager) and OpenMail-OpenMail moving
- ◆ Server-based searching
- ◆ Restore a single user from a backup



Routing & Delivery

7

OpenMail

- ◆ Flexible routing configuration, can route via public Internet
- ◆ Secure using server-server and/or end-end encryption
- ◆ Localized failure messages - including "similar" names
- ◆ Integrated SMS/Pager gateways (bi-directional)
- ◆ Integrated Lotus Notes/Domino connectivity
- ◆ Access control lists on services
- ◆ Route testing service
- ◆ Request server (users send mail to applications, receive replies)
- ◆ "Application Link" service for Business-Quality Messaging ("message queuing"), using the same OpenMail architecture



Directory

8

OpenMail

- **Single directory for all types of client**
- **Superset of Internet and X.500**
- **Flexible, fast, controllable synchronization to LDAP, X.500, Exchange, Domino/Notes, or your own custom data**
- **Scales to millions of entries with sub-second response time: suitable for "type-down" style access**
- **Multiple directories**
- **Soundex**
- **Extendable (e.g. photos, vCard)**
- **Access control lists on distribution lists and directories**



Microsoft Outlook Enterprise Scheduling

OpenMail

- ◆ **Assign tasks**
- ◆ **Publish your free/busy time to everyone**
- ◆ **Share calendar details with people you trust**
- ◆ **Real-time (not workgroup-replicated) so it's always up-to-date!**



What Else?

OpenMail

- ◆ Support for 64-bit OS (larger memory = bigger systems)
- ◆ Easy-to-use admin. Tools, tailored for different types of user
- ◆ Host several virtual systems on one machine
- ◆ "Fail-over" clustering
- ◆ High reliability - proven in the field
- ◆ Flexible, fair network licensing
- ◆ Commitment Request
- ◆ Available on different OS platforms
- ◆ Strong file types, "coercion", server-based converters
- ◆ Password controls (e.g. expiration, flexible composition rules)
- ◆ Rich server-based rules



Administration: the Basic Basics

OpenMail

- ◆ OpenView on HP-UX workstation
- ◆ MS Windows workgroup admin tool
- ◆ Web-based admin tool (7.0)
- ◆ or, Command lines, *e.g.*...
 - ◆ **omrc** Starts the OpenMail server
 - ◆ **omshut** Shuts down the OpenMail server
 - ◆ **omaddmn** Creates an OpenMail mailnode
 - ◆ **omaddu** Creates an OpenMail user
 - ◆ **omaddrt** Creates a route from OpenMail to another OpenMail or foreign email system
 - ◆ **omstat** Shows the overall status of OpenMail
 - ◆ **omsetsvc** Shows detailed information on OpenMail components
 - ◆ **omconflvl** Configures Event logging levels
 - ◆ **omconfaud** Configures Audit logging levels
 - ◆ **man openmail**
- ◆ "Real" sysadmin use *om...* commands
 - ◆ ...or at least understand their use



"Mailnodes"

12

openMail

- Central to OpenMail is the idea of a "mailnode"
- A mailnode typically consists of up to 4 Organisation Units (OUs), expressed as "OU1,OU2,OU3,OU4"

Some examples:

- rabbit
- rabbit,class
- HP-Pinewood,om1
- Non-HP-Pinewood,om1
- Marketing,enquiries,uk

To create a mailnode:

```
$ omaddmn -m "rabbit,class"
```

(this creates the mailnode as being local to this machine)

- **OU structure based on X.400**

- Can use other X.400 attributes (e.g. C/A/P/O), but careful!



Users

OpenMail

- ◆ **Each OpenMail user needs a UNIX login**

- you can choose to have this assigned automatically
- when any processes are run on behalf of the OpenMail user they will normally be run as the associated UNIX user

- ◆ **Assign to a local mailnode**

- ◆ **Assume you have a UX user called richi - to create an OpenMail user, specify the name and the mailnode they belong to:**

- \$ `omaddu -n "Richi Jennings/acme,uk,lab" -u richi -p password`
- (there are plenty more options for this command... check out the man page :-)



Internet Addressing

OpenMail

◆ Add the OpenMail mailbox

- `$ omaddu -n "Richi Jennings/acme,uk,lab" -u richi -p password`

◆ Assign it an Internet address

- `$ ommodent -e "Richi Jennings" -n "internet-addr=richi@acme.com"`

◆ Arrange for name to be resolved correctly on acme.com

- `$ echo "richi richi.jennings/ou=lab/ou=uk/ou=acme" >> /usr/lib/aliases`
- `$ newaliases`



ENU: the Error Notification User

- **Having an ENU is optional, but a Good Idea™**
- **ENU receives notification whenever some problem occurs (e.g. routing problems)**
- **Depending on the problem, it may be that that error message sent to the ENU is the only indication that something has gone wrong**
 - ENU tends to receive messages when something is "logically" incorrect (e.g. no return route configured, so error message can't be returned to the sender)
- **Could be a remote user, but is that a good idea?**
- **`omconfenu -n postmaster`**
- **`omshowenu`**



Routing: Local Users

openMail

- **We're now able to send messages between local users. But what about connecting two OpenMail machines together?**

- | | |
|---------------|----------------|
| HostA | HostB |
| acme,uk,sales | acme,usa,mktg |
| acme,uk,lab | acme,usa,sales |

- **HostA knows about mail to users at the mailnodes acme,uk,sales acme,uk,lab and acme,uk,accounts - but how do we tell it about the mailnodes at HostB?**

- We need to add a route to these other mailnodes!
- Hold that thought...

- **When we created the local mailnodes with the omaddmn command, we actually built an OpenMail "Routing Table". On HostA the routing table is actually more like:**

- LOCAL acme,uk,sales
- LOCAL acme,uk,lab



Routing: Services

OpenMail

✦ **Within OpenMail, everything revolves around “Services.” All the various tasks of a messaging system are separated into services that can be started and stopped by the operator (via the “omon/omoff” commands)**

- Service Router
- Local Delivery
- Directory Synchronisation
- Remote Client Interface
- Internet Mail Gateway
- Sendmail Interface
- X.400 Interface
- Notes Gateway
- ...etc...

✦ **All “routing” decisions are handled by the Service Router. All the mechanics of delivering a message to someone's Inray are handled by Local Delivery. etc...**



Routing: Queues

- OpenMail**
- ◆ **Generally speaking, the way services work is that they each read their own "queue."**
 - ◆ **For example, when a user hits send, the message is placed on the Service Router queue, the Service Router will read this and determine whom it is to go to next.**
 - If it's for a local mailbox, the router will place the message on the Local Delivery queue. Local Delivery then sees it and attaches this message to that person's In tray.
 - If it's for another machine, the router will place the message on the appropriate outbound queue...



Routing: Remote Users

OpenMail

- ◆ `$ omaddrt -m "acme,usa,*" -q smintfc -i openmail@HostB`
- ◆ **Now, mail addressed to any mailnode "acme,usa,..." will be placed on the SMINTFC queue, and will be sent to "openmail@HostB"**
- ◆ **"SMINTFC" is the name of the queue read by OpenMail's "Sendmail Interface"**
 - ◆ This takes the message, wraps it up into a form suitable for sending across the internet (we call it a Serialized/Encoded message), and sends it (via SMTP) to openmail@HostB
 - ◆ On HostB, sendmail will have been configured to recognize a message addressed to the special user "openmail", unwrap it back to OpenMail format and place it on the OpenMail Service Router queue
 - ◆ When the Service Router on HostB sees this message it will recognize the mailnode as a local mailnode and give it to local Delivery for delivery to the recipient's InTray



More on Services and Queues

- **In the last section I said that “services each read their own queue.” This is not always true...**
 - There are some services that are not associated with any queue
 - There are some queues that are not associated with any services
- **With each new release of OpenMail it is quite possible that new queues and services are added. So too, it's very easy for third party developers to add new services and queues.**
- **Some key ones and some of the easily forgotten, but very important ones...**



More on Services and Queues (2)

OpenMail[®]

Queue	Associated Service
ROUTER	Service Router
LOCAL	Local Delivery
SMINTFC	Sendmail i/f
UNIX	Internet Mail g/w
X400	X400 i/f
ERRO R	<i>none</i>
ERRMGR	Error Manager
SMERR	<i>none</i>
<i>none</i>	Remote Client i/f
<i>none</i>	Local Client i/f

Notes

- ♦ **Can be one-to-one relationship**
 - ...or a one-to-many
 - ...or one-to-none
 - ...or none-to-one!
- ♦ `omsetsvc-e` (see **next page**)
- ♦ **The two queues "ERRO R" and "SMERR" have no associated service**
 - They are there to handle any "corruption" or unexpected errors found by other OpenMail services
 - You *must* monitor these queues manually and deal with the message individually (see later)



omstat -s; omstat -x service

```

$ omstat -s
Service Router           Started           10.22.98         5
Local Delivery           Started           10.22.98         10
Internet Mail Gateway    Started           10.22.98         0
X400 Interface           Stopped           11.05.98         3
HPDesk Gateway           Stopped           0                0
Sendmail Interface       Started           10.22.98         0
Local Client Interface   Disabled           10.22.98         0
Remote Client Interface  Enabled           10.22.98         6
Test Server              Stopped           0                0
Request Server           Stopped           0                0
Print Server             Stopped           0                0
Directory Synchronization Stopped           0                0

$ omstat -x router
2 Service Router 6 Started 914102553 19.12.98 ROUTER 0

```



Error Queues

OpenMail

- ◆ **omstat only shows services that have queues**
- ◆ **You must remember to monitor queues that don't have services attached, particularly:**
 - ◆ **ERROR**—where any OpenMail service will place a message if it encounters a “corruption” (if it cannot be explained or corrected)
 - ◆ **SMERR**—where `xport.in` places any “bounced” **Serialised/Encoded (S/E) messages**
 - ◆ When we send out a S/E Message (via the `SMINTFC` queue) we send it via `sendmail` with the option “`-oem`” which says “Mail any errors back to me.” These will be placed on the `SMERR` queue.
 - ◆ **Cron job left as exercise for reader!**
 - ◆ `omstat -q error`
 - ◆ `omstat -q smerr`



Starting and Stopping Services with omon and omoff

- `omon -s service`
 - Start it
 - `omoff -s service`
 - Stop it
 - `omoff -s service [-d delay]`
 - Schedule it to stop
 - `omoff -c -s service`
 - Cancel the scheduled stop
 - **service can often be "all"**
- `/etc/omrc and /etc/omshut`
 - Start or stop *everything*





Directory Relay Service (drs)

- ◆ **Important for cross-server scheduling for Outlook clients**
- ◆ **Outlook free/busy information is stored in a special directory**
- ◆ **Allows direct connection between OpenMail servers for real-time free/busy lookup**
- ◆ **Alternative: use directory sync (but this can be out of date)**
- ◆ **See one of my other sessions for more details:**
 - "OpenMail support of Outlook and MAPI"



OpenMail Dæmons

- ◆ Treated similarly to services, but distinct
- ◆ `ommon` / `omoff` work similarly, except "non-stop" dæmons can't be `omoff`'ed

```
$ omstat -a
```

PC Monitor	Started	NON-STOP	0
Notification Server	Started	19.12.98	0
Shared memory daemon	Started	NON-STOP	
Notification Monitor	Started	NON-STOP	
Container Access Monitor	Started	NON-STOP	
Item Structure Server	Started	19.12.98	
Database Monitor	Started	19.12.98	
Licence Monitor Daemon	Started	NON-STOP	
P7 Daemon	Aborted	NON-STOP	
LDAP Daemon	Started	19.12.98	



Shared Memory Daemon (smd)

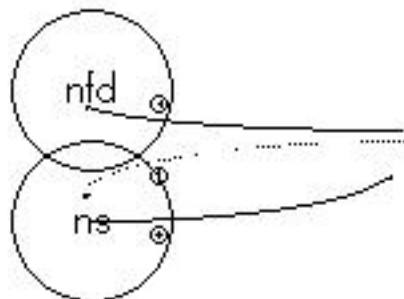
- Owner of shared memory for decoded config files**
- Holds important configuration in memory**
- Remember to use `omconfsm -f filename` if you change an in-memory file**
- re-reads the specified file(s) and creates a new shared memory segment
 - old-existing segment will have a delete placed against it and it remains there until the last process accessing it goes away
- You may wish to re-start any dependent services**

```
$ omconfsm -h
LangMap
CharMap
LCMap.bin
audit.cfg
filetype
convert
conv.noloss
convert.cs
files.intcs
dir.attribs
diratt.loc
ualth.cnf
cnvprogs
csetconv
toupper
CSInfo.cfg
CSMaps
CSTables.1d
T61Info.cfg
LMBCSInfo.cfg
suspend
```



Notification Service (ns) and Monitor (nfd) Dæmons

- Clients can request notification of events (*e.g.* new mail arriving)
- No expensive polling: greatly improves scalability



- ③ Client registers its interest in a particular notification
- ① Container library tells nfd about all changes ; nfd filters out those which are not interesting
- ① Client sends datagram when it's ready to receive notifications (not necessary for newer clients)
- ③ ns replies if new notifications are available; client can then fetch details via RPC



Shared Memory and Notifications

OpenMail

- ◆ **Note that the notification daemons allocate a block of shared memory, big enough to cope with the number of configured users + 10%**
- ◆ **If you plan to add a bunch of new users, it's worth restarting the client services and the notification daemons!**



Item Structure Service Dæmon (iss)

- **Logs Message Store activity**
- **Allows single user restore, invaluable troubleshooting, etc.**
- **Logs activity at the "Container" level**
 - On add or delete to/from a container, this information is logged
- **Log keeps track of both a container's children and parents**
 - It tracks *both* directions of the hierarchy
- **Main components are:**
 - The Item Structure Service (iss)
 - a dæmon (**omissdm**) that records/logs structural changes
 - log files are in **~openmail/structlog/IS***
 - The Item Structure DataBase (isdb)
 - in **~openmail/is/O/***
 - **omupdtis** command
 - the utility that applies the logs against the isdb



Using the Item Structure Service For the First Time

- **Although the daemon is running, logging is *not* enabled by default!**
- 1. **Ensure iss is running** (`omstat -a`)
- 2. **Write down the system date and time:** `date | lp`
- 3. **Get the initial state:** `omscan -a -l /tmp/scanlog`
- 4. **Initialize the database:** `omupdtis -l -l /tmp/scanlog`
- 5. **Incorporate recent activity and remove the old logs:** `omupdtis -r -f mm.dd.yy -F hh:mm:ss`
 - Use the date and time that you wrote down earlier



Single User Restore

or, "Help! I deleted a really important message!"

OpenMail

• Now that you have an isdb, you can do SURs!

• A (very quick) overview:

- Decide on a date/time where you know the message existed; apply the logs up to this point
- Generate a list of files that need to be restored
 - `omprepsur` (use same date/time)
- Restore the files into a temporary directory
 - `omgetsur` (or your own method)
- Check that all's OK
 - `omprepsur` again, but without `-s` or `-o` options
- Re-create the mailbox
 - `omdosur` (use same date/time)
 - `omdelu/omaddu`
 - `omcpinu`



Kernel, Swap and Filesystem Configuration

OpenMail

◆ Kernel

- Calculate new values for `maxdsize`, `maxswapchunks`, `maxuprc`, `nccallout`, `nproc`, `nfile`, `maxfiles`, `ninode`, `nflocks`, `msgssz`, `msgtql`, `msgmap`, `msgmnb`, `sम्मnu`, `shmmni`, `db-max-pct` (see the *OpenMail Technical Guide* for formulæ)

◆ Swap

- Allow about 1MB per logon (more or less, depending on the client mix—see the *Guide* for more details)

◆ Inodes for `/var/opt/openmail/data/`

- Allow 1 inode per 2KB



HP-UX Tip for Spotting Swap Problems

- **Set kernel global parameter `mman_elog` to 1**
- **Logs problems in `dmesg` and `syslog`, e.g...**
 - *Process "stackhog" Pid 1659 Uid 409 : received a SIGSEGV for stack growth failure*
 - *Apr 17 09:38:46 hal vmunix: Process "stackhog" Pid 1659 Uid 409 : received a SIGSEGV for stack growth failure*
 - *Process "memhog" Pid 1718 Uid 409 : brk/sbrk failure as insufficient swap space - returning ENOMEM*
 - *Apr 17 09:50:46 hal vmunix: Process "memhog" Pid 1718 Uid 409 : brk/sbrk failure as insufficient swap space - returning ENOMEM*
- **Requires `PHKL_11340` for 10.20**
- **Speaking of logging...**



Event Logging

• You can modify the logging level of each product component separately

• Standard levels shown below (there are others, but they are probably not useful without a support call)

0 Logging not enabled

1 Catastrophic error/abend

3 Error but the process can continue

5 Warning of possible problem

7 Command has executed successfully

9 Something has been done to a message

11 Log debug information in binary code

13 Log debug information in library code

15 Log procedure call in binary code

17 Firewall routine procedure call

• Use `omshowlvl/omconfvl` to show/set the levels, `omshowlog` to output a log

- You can filter by level, the date/time range, the component and request logs pertaining to a particular user

• `omcommon` is a useful quick check—why not run it from cron twice a day?

• Where are the log files? `~openmail/logs/...`

- `log.[012] fatal flvix.log daemon.stderr`

OpenMail



Audit Logging

OpenMail

- ◆ **Great for billing (that's what it was designed for)**
- ◆ **Also useful for debugging**
- ◆ **Use `omshowaud/omconfaud` to show/set audit levels**
- ◆ **Written to `~openmail/logs/audit`**
- ◆ ***Ridiculously* configurable!**



Patch Process

OpenMail

• **Periodic patches**

- Cumulative
- Nominally once per month
- Automatically regression tested (in addition to normal QA)
- Simpler patch management
- Can be used proactively
- Uses swinstall
- Ensure you have enough disk space before you begin!

• **Point patches**

- For high priority problems (e.g. machine down)
- Dependent on a PP (generally the most recent)
- Not usually regression tested
- Should not be used proactively



Checklist!

39

- Configure appropriate Kernel parameters
- Plan filesystem layout and swap space
- Filesystems have enough inodes?
- Create UNIX logins for the local users (optional at 6.0)
- Add local mailnode(s)
- Add local users
- Configure user password controls*
- Configure the Error Notification User
- Configure Public Distribution Lists*
- Define any new or localized Directory attributes*
- Configure the file type coercion settings*
- Configure the PrintServer*
- Configure the Request Server*
- Configure the converters and body part identifications*
- Create the top-level Bulletin Boards*
- Schedule the OpenMail maintenance commands in cron
- Schedule the back up procedures to run regularly
- Set the general configuration file options (general.cfg)*
- Set the user specific configuration options (user.cfg)*
- Set the client configuration options (userconf)*
- Schedule omprint for users without a client*
- Configure Access Control Lists*
- Configure the Logs*

* - denotes optional items



Checklist! (2)

- Set up isdb and iss logging
- Configure sendmail (or other MTA)
- Configure routes to the other OpenMail systems
- Configure Directory Synchronization
- Configure Bulletin Board Synchronization*
- Configure the default Internet Mail Gateway mailnode
- Configure routes to the Internet Mail Gateway
- Add internet mail users to the Directories*
- Configure the address encoding and decoding for the Internet Mail Gateway*
- Configure the special character encoding for the Internet Mail Gateway*
- Configure the name conversions for the Internet Mail Gateway*
- Configure the address domain mappings for the Internet Mail Gateway*
- Configure the body part conversions for the Internet Mail Gateway*
- Set up other gateways/connectors:
 - MS Exchange
 - Lotus Notes/Domino
 - SMS/paging
 - HPDeskManager

* - denotes optional items



Periodic Administration Tasks

- ◆ Check all the services are running
- ◆ Check the In Tray of the Error Manager
- ◆ Check messages are moving through the system
- ◆ Check for messages that could not be transmitted
- ◆ Check for system errors
- ◆ Monitor disk usage
- ◆ Back up the system
- ◆ Check and repair data inconsistencies
- ◆ Add and remove local users
- ◆ Maintain the Directory of local and remote users
- ◆ Maintain the Public Distribution Lists
- ◆ Maintain the Routing Table
- ◆ Maintain the Bulletin Boards
- ◆ Maintain the Message Store (omtidyallu)

OpenMail



Super Tweaks

`/var/opt/openmail/sys/general.cfg`

openMail

- **Limit auto-reply frequency to same address (cf vacation)**
 - `ID_AUTOREPLY_CHECK_ON=TRUE`
 - `ID_AUTOREPLY_EXPIRY_TIME=# days`
- **Limit severity of non-delivery reports to ENU**
 - `NDN_BM_SERIOUS_ONLY=TRUE`
- **Don't show "similar" names from the directory in non-delivery reports**
 - `NDN_NO_ALTERNATES=TRUE`
- **Keep an archived copy of messages that go through the service router**
 - `SR_DUMP_MESSAGES=BEFORE (or AFTER)`
- **Expand public distribution list members in routed messages**
 - `SR_EXPAND_FDL=TRUE`
- **Set loop ping check hop count (default 100)**
 - `SR_MAX_HOP_COUNT=count`
- **Optimize the CDA server rebuilds**
 - `CDA_USE_CHANGE_LOG=TRUE`
- **User must type password**
 - `UAL_DISALLOW_AUTO_PASSWORD=TRUE`
 - ... or ...
 - `UAL_DISALLOW_NON_USER_PASSWORD=TRUE`



Super Tweaks (2)

`/var/opt/openmail/sys/general.cfg`

OpenMail

Use distributed licensing

- `LML_USE_DIST_LIC=TRUE`

Enforce compatibility between different clients

- `UAL_DISABLE_NESTED_BBS=TRUE`

Mail a warning to user if password expires soon

- `UAL_PWD_WARNING_DAYS=days`

Mail a warning to users who log on with an old password (i.e. don't enforce aging)

- `UAL_PASSWORD_AGED=WARN`

Allow aliases to be used for signon

- `UAL_SIGNON_ALIAS=YES`
- `UAL_USE_SIGNON_ALIAS=TRUE`

Mail a warning to users and/or ENU when approaching and exceeding quotas

- `UAL_SIZE_MSG_TO_USER=TRUE`
- `UAL_SIZE_MSG_TO_ENU=TRUE`
- `UAL_SIZE_WARNING_LIMIT=% max_limit`
- `UAL_SIZE_WARNING_BOUNDS=% increase`

Use UNIX users' passwords

- `UAL_UNIX_PASSWORD=TRUE`



Big Tweaks for Small Systems

also in general.cfg

OpenMail

- ✦ **Maximize throughput at the expense of delivery speed**
 - Q_WAKE_READER=FALSE
 - Q_TIMEOUT=seconds (120 perhaps?)
 - SR_Q_TIMEOUT=seconds (ditto)
- ✦ **Limit number of concurrent searches**
 - SE_MAX_CHILDREN=# (default 20)
- ✦ **Make CDA less aggressive**
 - CDA_CHECKTIME=minutes (default 5)
- ✦ **Limit number of concurrent remote free/busy lookups**
 - DRS_MAX_CHILDREN=# (default 16)
- ✦ **Bypass LD for special messages [small]**
 - SR_LD_BYPASS_LSERV=TRUE
- ✦ **Don't tidy trash in background after client signoff (don't forget to schedule `antidyallu` at night!)**
 - UAL_NO_WB_EMPTY=TRUE
- ✦ **Kill idle/dead client connections**
 - UAL_IDLE_TIMEOUT=minutes
 - UAL_DEAD_TIMEOUT=minutes
- ✦ **Kill idle Internet clients (default 300)**
 - UAL_POP3_TIMEOUT=seconds
 - UAL_MAP4_TIMEOUT=seconds



What's New in 6.0?

OpenMail

- ◆ **PQA** (OTN 300-0150)
- ◆ **MAPI/Outlook & Exchange** (300-0154)
- ◆ **RTF** (300-0160)
- ◆ **IMAP** (300-0170)
- ◆ **Other internet support** (300-0173)
- ◆ **/etc/passwd, NIS and UIDs** (300-0157)
- ◆ **Message Ids** (300-0156)



PQA

OpenMail

- ◆ **Why change it?**
- ◆ **New queue manager**
- ◆ **Message pool**
- ◆ **POISON / IDEL, lazy deletion**
- ◆ **Multiple queue readers**



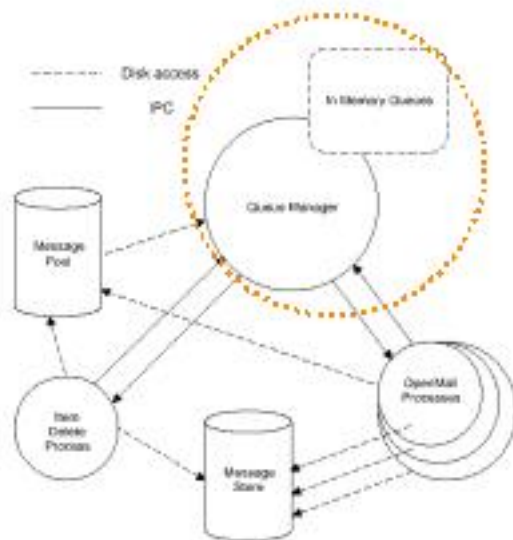
1. Why Change It?

- ◆ **On large systems, ROUTER and LOCAL queues could back up, at peak times**
 - ◆ Even if there was spare CPU and I/O bandwidth
 - ◆ Because the SR and LD services were single-threaded
- ◆ **Queues could become I/O bound**
- ◆ **Scope to make queues multi-reader**
- ◆ **Scope to cache more state in memory**
- ◆ **Scope to do lazy deletes**
- ◆ **Improved handling of "poisoned" messages**



2. New Queue Manager

- Previously, services accessed queues directly**
- In-process, using shared libraries
- Now, it's *only* the queue manager daemon which accesses the queues**
- Much faster: can keep state in memory
 - Services talk to QM via IPC



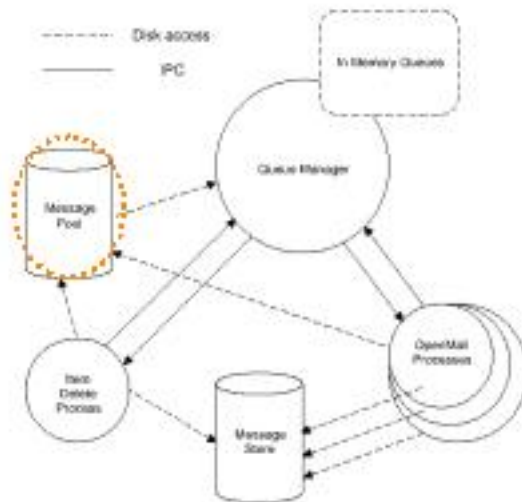


3. Message Pool

49

OpenMail

- **Holds in-transit messages**
- **Accessed mainly by services**
- **Stored in ~openmail/msgpool/**
 - Potential hotspot, but designed to be compatible with striped arrays





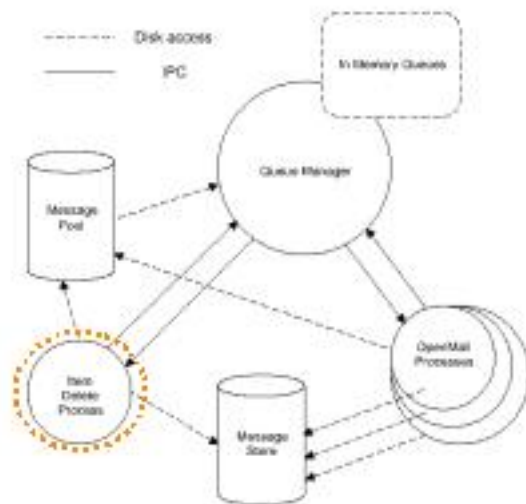
4. POISON and IDEL

POISON holds onto any "poisoned" messages

- These are corrupted messages which are causing a service to abort
- QM will try three times and then move it to POISON

IDEL receives messages for deletion

- idel.server will delete them asynchronously, at low priority
- QM notes when message processing is complete (orphan) and puts it on IDEL





5. Multiple Queue Readers

- **Previously, could have many writers, but only one reader**
- **Now, we can have multiple:**
 - Service router
 - Local delivery
 - SMTP "transport" out (xport.out)
 - SMTP "pure" out (unix.out)
- **"Main" readers and "Auxiliary" readers**
 - `omsetsvc -x ld 3` (gives 4 simultaneous local deliveries)
 - `omstat -s -p` (get status for auxiliaries)
- **Can configure up to 20 auxiliaries**
 - Careful: you could slow your system down
 - Tip: monitor system and add auxiliaries slowly!

OpenMail



MAPI/Outlook & Exchange Connector

- ◆ **New functionality**
- ◆ **Automatic profile generation**
- ◆ **Direct references**
- ◆ **MAPI property storage**
- ◆ **Notifications and Reminders**
- ◆ **Appointment and Free/Busy storage**
- ◆ **Cross-server lookup**
- ◆ **Delegates**
- ◆ **Goodbye to WINMAIL.DAT**
- ◆ **MAPI.CFG**

OpenMail



1. New Functionality

OpenMail

5.20

- Default store
- Calendaring
- Outlook 98
- Performance

5.30

- Full delegates
- Remote mail improvements
- Better progress indicators
- Performance

5.40

- Outlook 2000
- Receive HTML body
- Entrust Express
- Windows CE and Palm sync.
- Performance!



2. Automatic Profile Generation

- **MAPI forces you to create “profiles” for each user**
- **Stored in the registry**
- **Contains configuration information for each MAPI “service”**
- **Creating and/or supporting is time-consuming**
- **Two utilities to make this easier**
 - **NEWPROF.EXE**
 - Simple tool
 - Free from Microsoft
 - **Profile Maker**
 - More complete tool
 - Product from Automatic Profile Management, LLC
 - http://www.autoprof.com/guide/hp_openmail.htm



3. Direct References

- ◆ **AKA *Direct Index***
- ◆ **Handle to any OpenMail object in the store**
- ◆ **Non-hierarchic (unlike container/item pairs)**
- ◆ **Used extensively by MAPI**
- ◆ **~openmail/dits/ contains the mapping data**
 - Don't delete anything in this directory!

OpenMail



4. MAPI Property Storage

- **In MAPI, an object is defined as a collection of properties**
 - Persistent or Temporary
 - Read/write or read-only
 - List, Relationship or Computed
 - Defined or Custom
- **OpenMail stores the properties in the normal places, for interoperability**
 - Container records
 - Transaction files
 - Basic items (e.g. PR_BODY)
- **Also, TNEF object file for MAPI-specific information**
 - (e.g. calendar data)



5. Notifications and Reminders

OpenMail

◆ We don't poll for new mail

- OpenMail server sends notification packet to MAPI provider
 - known as *Server Push*
- MAPI provider then fetches information when convenient
- Handled by nfd (the notification daemon)

◆ In 5.20, Reminders were triggered on the server

- This is the standard MAPI model...
- ...but it performs poorly on big servers

◆ In 5.30, we trigger them on the clients



6. Appointment, Free/Busy storage

OpenMail

◆ **Appointments**

- Stored as messages in the Calendar folder
- Most of the properties in the TNEF object file

◆ **Free/Busy**

- Stored as "LocalFREEBUSY" message in the Calendar folder
- An extract is "published" to FREEBUSY on client schedule
- Don't forget to use `omaddfb`
- Also `-F` option to `omaddu`



7. Cross-Server Lookup

OpenMail

- **Because FREEBUSY is actually a hidden directory (address book), we can do some clever things to work over wide areas**
 - We could use directory synchronization to share the information out
 - but this is how Microsoft do it, and it causes problems
 - So, instead, we use the Directory Relay service (drs)
 - makes real-time contact with remote server
- **mnMapFile stores mapping between mailnodes and server FQDNs**
 - Built automatically
 - commands to edit it, if you must, or just use vi!
 - `omadd mnmp`, `omdelmnmp`, `ommodmn mp`, `omshowmn mp...`



8. Delegates

OpenMail

◆ New in 5.30

◆ “Principals” can delegate some/all of their mailbox to “delegates”

- Send messages
- Send meeting requests
- Send task requests
- Respond to meeting requests
- Respond to task requests
- Respond to messages
- Access some or all private folders



9. Goodbye To All .DAT

- **From 5.20, we no longer attach a WINMAIL.DAT file**
- **Instead, the TNEF is stored as a hidden object file**
- **You can revert to the old behavior if you wish (a MAPI.CFG setting)**
- **TNEF route**
 - For connecting to an Exchange server with high fidelity
 - Similar to a MIME route, but builds TNEF file for each message
 - `omconfux -t exchange`
 - (omaddr not necessary, as done automatically by omconfux)
 - Set `UX_PRE_5_20_COMPATIBILITY_MODE=FALSE`
 - Use `tnefout.str` (not `mimeout.str`)

OpenMail



10. MAPI.CFG

- Stored on the server, enforces some client configuration settings**
- Client checks for new version at logon
- [Mail]**
- Compatible Messages=1
- [Display]**
- ShowMailnodes=1
 - ShowCompleteInternetAddress=1
- [Directories]**
- 1=LOCALREPLICA
 - 2=SYSTEM



RTF

63

- OpenMail
-
- 1. General**
 - 2. cc:Mail**
 - 3. Request Server**
 - 4. Filetyping**



1. General

- ◆ **Used to convert MAPI/Outlook RTF body text to plain text, for client interoperability**
- ◆ **In previous releases *rtf.browse* was a separate process**
 - Now a library (although *rtf.browse* binary provided for backwards compatibility)
- ◆ **Now supports multi-byte character sets**
 - Charset conversion is configured in the steering files
 - *mimeout.str*, *brwmime.str*, etc.
 - Searching also honors charset



2. cc:Mail

- ◆ **A form of RTF is used in OpenMail as a storage format for cc:Mail “Highlighted text”**
- ◆ **This allows non-cc:Mail users to read messages from cc:Mail**
 - Set `ConvertTextToRTF=1` in cc:Mail’s `HPMAIL.INI`
- ◆ **Likewise, RTF messages from Outlook can be displayed with higher fidelity in cc:Mail**
 - Set `RTFConvertIfColoured=1` and `EditConvertedRTF=1`



3. Request Server

◆ A request server script can now accept input in RTF

- Use the request server steering file to convert the RTF item to text
 - in `/opt/openmail/req/scriptnameout.str...`
 - 2130 1167.ISO8859_1 R
- `REQ_IGNORE_TYPES_OF_ATT` tweak is deprecated

OpenMail



4. Filetyping

- Some Microsoft internet clients send both .DOC and .RTF files as application/msword**
- Not helpful for incoming RTF files, as they'll be typed as DOC files
- Filetype coercion now supports OLE2 formats (which don't have a simple magic number to distinguish between them)**
- Before 6.0, all Office 97 types appeared the same!
 - Now, we can distinguish between Word, PowerPoint, Excel, etc.



IMAP

- 1. Improvements**
- 2. Object file cache**
- 3. IMAP4rev1 and extensions**
- 4. Tweaks**

OpenMail



1. Improvements

openMail

Now an OpenMail daemon

- {omon -a imap, omoff -a imap, omstat -a [imap]}

New connections start faster

- Forked from main process, not from inetd, so no exec or initialization required

Heavy use of memory-mapped I/O internally

New cache design

- Based on actual client usage, rather than theory
- Memory and object files
- Removes disk space overhead of 5.10

Supports server-push notifications

MDNs and DSNs

Searching performance optimizations

Exhaustive size checking, in attempt to prevent buffer overflow attacks



2. Object File Cache

- ◆ **An object file called `imapBodyStrc`**
- ◆ **Skeleton RFC822 format message**
- ◆ **Just the headers, plus enough information to quickly reconstruct the rest of the message**
 - Data structures optimized to quickly allow partial rendering
- ◆ **Automatically maintained**
- ◆ **`IMAP_CACHE_RFC_HEADER` and `IMAP_CACHE_RFC_MESSAGE` tweaks are now deprecated**



3. IMAP4rev1 & Extensions

openMail

STATUS

- Get overview information about a mailbox (folder) without opening it

FETCH

- Now supports sections and octet ranges ("partial")
- e.g. FETCH 1 body[2.2.2.text]

Multi-byte character sets

- Modified UTF-7 for mailbox names
- UTF-8 for everything else
- Note: avoid non-ASCII login names (standard is vague)

Also other "capabilities"

- NAMESPACE; IDLE; X-NETSCAPE; X-OPENMAIL; CHILDREN; LITERAL+



4. Tweaks

- **IMAP_MIN_SIZE_ESTIMATE=10**
 - Only return exact size for messages under 10K
- **IMAP_SEARCH_TIMEOUT=10**
 - Kill searches that take longer than 10 seconds
- **IMAP_AUTOMATIC_MDN=TRUE**
 - Automatically generate read receipts (IMAP clients *should* do this, but some don't)
- **IMAP_CONNECTION_LIMIT=5000**
 - Start no more than 5000 IMAP sessions
- **IMAP_CONNRATE_LIMIT=3**
 - Start no more than 3 sessions per second



4a. More Tweaks!

OpenMail

- **IMAP_USE_ITEM_BROWSER_SELECTION=TRUE**
 - Can work around obscure client compatibility problems
 - Beware performance hit!
- **IMAP_IDLE_TIMEOUT=31**
 - Kill sessions idle for more than 31 minutes
 - UAL_IMAP4_TIMEOUT deprecated
- **IMAP_BB_FOLDER_PREFIX=pubfolders**
 - Set prefix for public folder access
- **IMAP_FOLDER_SEPARATOR=. and IMAP_BB_FOLDER_SEPARATOR=.**
 - Use '.' as folder separator, instead of '/'
- **IMAP_DELETE_SUBFOLDERS=TRUE**
 - Work around non-conformance of some clients



Other Internet Support

- 1. Preserving headers**
- 2. DSNs and MDNs**
- 3. Katakana, and other tweaks**

OpenMail



1. Preserving headers

OpenMail

- ◆ **Incoming SMTP message headers retained**
- ◆ **Good for POP and IMAP clients**
- ◆ **Examples where this is helpful:**
 - Content-disposition: [inline | attachment]
 - Content-ID:
 - used for including inline images for HTML body
 - X-...



2. DSNs and MDNs

OpenMail

- ◆ **DSNs inform sender when message has been delivered by final MTA**
 - ...or non-delivered, expanded
- ◆ **MDNs inform sender what the recipient does with the message**
 - Delete, read
 - Sent manually or automatically
 - Due to manual or automatic action



3. Tweaks

OpenMail

◆ **CONV_UNIXJIS_ALLOW_HALFW_KANA=TRUE**

- Retain old single-byte Japanese format (JIS X 0201-1976)
- Don't convert to the current standard (RFC1468) double-byte JIS format (ISO-2022-JP)

◆ **UXI_NO_INET_OBJFILES=TRUE**

- Suppress the new header retention behavior

◆ **INET_INLINE_FNAME_ALLOWED=FALSE**

- Don't generate filenames for inline bodies
- Many clients are confused by bodies with filenames (although this is perfectly legal)

◆ **UXO_MIME_SPACE_OK_IN_FNAME=FALSE and BRW_MIME_SPACE_OK_IN_FNAME=FALSE**

- Suppress special characters in item filenames (e.g. spaces)

◆ **UXI_NO_CONVERT_REPORTS=TRUE**

- Don't convert incoming DSN/MDN

◆ **IMAP_AUTOMATIC_MDN=TRUE**

- Automatically generate read receipts (IMAP clients should do this, but some don't)



/etc/passwd, NIS and UIDs

- 1. What changed?**
- 2. The UID pool**
- 3. Changes to commands**
- 4. Notes & Tweaks**

OpenMail



1. What Changed?

79

- **Previously, each local OpenMail user needed an account**
- **In 6.0, it is only necessary to assign an unused UID**
 - No username is necessary
 - No home directory necessary
 - No entry in /etc/passwd or NIS
- **This means:**
 - Performance improved
 - Easier to administer
- **Still need a UID, for security reasons and for OS scheduling**
- **OpenMail maintains a “pool” of UIDs that will not be used for real accounts**



2. The UID Pool

80

OpenMail

- ◆ **Pool stored in `~openmail/sys/idpool`**
- ◆ **Set up and manage the pool:**
 - `omadmip -a 50000 -n 5000`
 - Create a pool, 5000 big, starting at UID 50000
 - `omadmip -u`
 - Show statistics
 - `omadmip -f`
 - Fix pool (clear up orphans)
 - `omadmip -j`
 - Show total free UIDs in pool (*undocumented*)



3. Changes to Commands

OpenMail

◆ **omaddu**

- -u no longer mandatory!

◆ **omshowu**

- Displays UID, if no /etc/passwd or NIS entry
- -U option can take UID (no change here)



4. Notes and Tweaks

- ◆ **Users running omadmin must have a home directory**
- ◆ **If you use UAL_UNIX_PASSWORD=TRUE then you can't use the UID pool**
- ◆ **Auditing now logs the UID where it once logged the account name**
 - AUD_LOG_UX_NAME=TRUE reverts to the previous behavior (but slows system down a little)
- ◆ **ITO/ITA doesn't support the UID pool**
- ◆ **AdvMail/TL users must have a logon**
- ◆ **Print server *may* not work correctly**



Message IDs

OpenMail

- **Previously, 'H' + 7-byte-hex-user-id + 8-byte-hex-item-number**

- E.g. "H00000720001f8d3"
- 'C' instead of 'H' if client specifies ID (rare)
- Remote possibility of duplicates
- Made tracking hard

- **New format is:**

- Old-style-ID + '.' + seconds-since-epoch + '.' + FQDN
 - e.g. H0000066000004a1.0929522826.om.example.com
- Maximum size 284 bytes

- **IM_MAKE_MSG_ID_GLOBAL_UNIQUE=FALSE**

- Turns it off (not necessary for backwards compatibility)



Other Resources

OpenMail

◆ **<http://www.hp.com/go/openmail>**

- 200- and 300-level courses!
- Technotes
- TFM's
- White papers



OpenMail is...

- OpenMail**
- ◆ **Scalable**
 - ◆ **Flexible**
 - ◆ **Reliable**
 - ◆ **Low-cost**
 - ◆ **Feature-rich**
 - ◆ **Future-proof**



OpenMail

**Business Messaging for the next E.
E-services.**

www.hp.com/go/openmail