7.9 Adding Filters, Shapers + EITs

7.9.1 Adding A New Encoded Information Type

If you wish to add a new encoded information type, then some editing of the *tailor* file is necessary.

Suppose we wish to add a new body type "binary" which we want to be able to submit and have converted to `ia5` before transmission.

1. Add a new body type to the list of body types in the *tailor* file.

   ```
   bodypart    binary
   ```

2. Now specify which outbound channels can handle such a body part by adding this type to the `bptout` list associated with the channel. If no outbound channel will be delivering this type then this step is unecessary. This is the case for `binary`.

3. To allow *submit* to work out how to convert from `binary` to `ia5` it is necessary to add a new reformatting channel, such as the following:

   ```
   chan bin2ia5   prog=fcontrol, type=shaper, bptin=binary,
                  bptout=ia5, outinfo="uuencode binary",
                  show="Binary -> Text"
   ```

   This specifies that to convert input bodypart type `binary`, to output body type `ia5` the channel *fcontrol* should run the process `uuencode program` with standard input from the `binary` body part and standard output into the `ia5` body part.

   In practise, it might be better not to call *uuencode* directly, but wrap it up in a shell script; we could then print out some preliminary remarks instructing the recipient how to unwrap the binary part.

Now, when *submit* receives a message which has flagged that it contains a body part of `binary`, it will be able to convert that part into `ia5`, which most output channels can deal with.

7.9.2 Adding a New Content Type

Adding a new content type for PP to relay is similar in style to adding a new encoded information type. The steps that need to be taken are dependent on whether conversion is involved or not.

If no conversion is involved, then to add a new content type such as `p22-encrypted`, for instance, would simply consist of registering that content type with an outbound channel. This is done by adding a new channel with the `content-out` part set to `p22-encrypted`. In this case, *submit* will pass the message straight through, unaltered, providing the recipient is reachable via that channel.

If conversion is required, then two things are necessary. The first is to add a new unflattening channel which is capable of splitting the message into component parts. Secondly, the component parts must be able to be reformatted into whatever is required on the outbound channel.

For example, suppose we have a new message content called "whizbang". This consists of a "whizbang" specific header and a standard `ia5` body part. To allow the type of mail to be input, the following would be required.

- An exploder channel that can split up "whizbang" format mail into its components.

  ```
  chan WBexplode    prog=WBexplode,type=shaper,
                    content-in=whizbang,
  ```

  This channel is defined to split up the message into two components, a file `hdr.wb` and a body part `1.ia5`.

- The new encoded information type `hdr.wb` needs to be registered in the tailor file

  ```
  headertype wb
  ```

- The `ia5` body part is okay as it stands, but to relay the message to X.400 or RFC-822 the header will need converting, therefore a filter to convert `hdr.wb` into either `hdr.822` or `hdr.p2` will be necessary. Only one need be written, because if *submit* has a `hdr.822` it can convert it to a `hdr.p2` using the *822top2* channel. Such a channel definition might look like this:

  ```
  chan wbhdr-2-822 prog=wbto822,type=shaper,hdrin=wb,
                   hdrout=822,
                   show="WhizBang to 822 Header converter"
  ```

7.10 The Day–to–Day Running of PP

7.10.1 How to Start Up the PP System

Each time the UNIX machine is started up, the *rc.local* script in your UNIX system is executed. An entry for PP should be placed in *rc.local*. This entry should refer to a script which starts up various programs related to the running of PP. Typically this script is similar to *pp.start* (see appendix B) which starts up:

- QMGR: The queue manager which runs as the PP userID.

- *pptsapd*: The PP TSAP Daemon which also runs as the PP userID.

- *smtpsrvr* (optional): The SMTP listener which runs as the superuser. This is only necessary if:

  1. You are using the SMTP protocol.
  2. You are not running SMTP from *inetd* (see Section 3.8.2 on page 34 for details of this).

- *tsapd* (optional): is necessary if you are expecting incoming X.400 messages. This is assumed to be part of the general ISODE configuration; see
  isode-gen (8) for details.

Normally, these programs will run without interruption unless something serious happens, such as the machine going down. But if necessary you can always stop them by using the UNIX *kill* command. You can restart them by running the script manually or by invoking the programs separately, then letting them run in the background.

7.10.2 What to Run Each Night

In addition to running *pp.start* whenever the machine starts up, there are other things you should run regularly, ideally each night.

For example, if you build the PP tables from other sources, such as the NRS database, local system databases or other external sources, then running *dbmbuild* each night is a useful thing to do.

All the nightly routines can be run from a shell script, conventionally named something along the lines of *pp.night* (see appendix C for an example).

The sequence of operations should be :

- Archive the day's logs and remove outdated logs (see Section 7.10.3 on page 133).

- Collect statistics.

- Import new table source data files if necessary.

- Build new tables.

- *dbmbuild.*

7.10.3 Saving Logs

Saving logs depends very much on the individual policies for each site. For many cases the logs can just be set to truncate themselves when a suitable size is reached. Alternatively, if you are more concerned about keeping the information you should backup the logs each night. Two important exceptions to these general rules above are the *oper* log and the *stat* log.

The *oper* log should under normal conditions be empty, or at least very small. For this reason it is worth checking this overnight and perhaps mailing it to the postmaster if it contains anything.

The *stat* log can be processed to show statistics. There is a program in the *tools* directory call *statp.* This program processes the *stat* log file into a more structured format that is suitable for processing with *awk* (1). A sample awk script is supplied which shows the sort of output that can be produced from this.

7.11 Examining the PP Queue

Occasionally there may be problems encountered by PP that cannot be resolved by normal means, and it will be necessary to look at the queue directory to find out what is going wrong. This section is a brief guide to this.

The PP Queue is stored under one directory, the *quedir* directory. An item in the queue is contained within a subdirectory of *quedir* with a unique name. This subdirectory may contain the following:

addr: a text encoded file containing information corresponding approximately to P1 functionality.

base: a directory which contains the message as it arrived.

filter.number: other directories which contain transformed versions of the message content.

report.number: files containing text encoded delivery reports associated with the message.

Of these components only the *addr* file must be present.

Each message content is a (possibly degenerate) directory tree. This hierarchy represents the structure of a message, with each body part being represented by a single file. Subdirectories represent forwarded IP messages. This structure is to allow for straightforward transformation of messages. When a message is converted, the body parts not affected are linked, rather than being copied.

An example queue file is shown in Figure 7.2.

7.11.1 Delivery Reports

The basic operation of a channel is discussed in Section 3.9 on page 48. Delivery Reports are text encoded, using a format similar to that used for text delivery. Reports are stored as files in the message directory, with name `report.<n>`, where n is the PP number for the lowest recipient to which the DR refers. Where a DR refers to multiple recipients, all except the lowest are marked to done in the queue, and the first is marked to dliv.

An example text-encoded delivery report is shown in Figure 7.3.

7.11.2 Receipt Notifications

These are handled by the UA, and are thus beyond the scope of PP. Mappings to RFC 822 are done according to RFC 1148.

```
Start-of-MsgEnvPrm
Message-type User-Mpdu
Message-size 84
Number-warnings 000
Warning-interval 24
Return-interval 72
Encoded-info EncTypes="hdr.822,ia5"
alt-recip-allowed true
cont-return true
Ua-id Test
pp-cont-corr "Subject: Test, To: pac"
content-corr 04165375626a6563743a20546573742c20546f3a20706163 none
Inbound-channel smtp
Inbound-mta "xtel.co.uk"
MsgId string="lancaster..123:25.08.91.15.53.50" Country=GB Admin=" "
Prmd="X-Tel Services"
Trace Country=GB Admin=" " Prmd="X-Tel Services"
arrival="Wed, 25 Sep 1991 16:53:49 +0100" action=0 End-of-block
Trace mta="lancaster.xtel.co.uk" Country=GB Admin=" "
Prmd="X-Tel Services"
arrival="Wed, 25 Sep 1991 15:53:50 +0000" action=0 End-of-block
Queued-time "Wed, 25 Sep 1991 15:53:51 +0000"
Start-of-MsgEnvAddr
Origs rno=000 status=done reform-done=000 reform-list="822tous|822tous.1"
outchan=dr2rfc outmta="lancaster.xtel.co.uk" ext-id=0 resp=false
mta-rreq=basic usr-rreq=basic type=822 eits="ia5,hdr.822-us"
content=822 orig="pp@xtel.co.uk"
x400="/I=P/S=Pat/O=XTEL/PRMD=X-Tel Services/ADMD= /C=GB/"
rfc="P.Pat@xtel.co.uk" end-of-addr
Recip rno=001 status=pend reform-done=001 reform-list="822tous|822tous.1"
outchan=822-local outmta="lancaster.xtel.co.uk" ext-id=1
resp=true mta-rreq=basic usr-rreq=basic type=822
eits="ia5,hdr.822-us" content=822 orig=pac
x400="/I=p/S=cowen/OU=x400/O=XTEL/PRMD=X-Tel Services/ADMD= /C=GB/"
rfc="p.cowen@x400.xtel.co.uk" end-of-addr
```

Figure 7.2: Example of Address File

```
Msgid string=DR020905-900816141523Z Country=gb Admin=" " Prmd=interspan
Trace mta="bells.cs.ucl.ac.uk" Country=gb Admin="gold 400" Prmd="uk.ac"
        arrival="Thu, 16 Aug 1990 14:15:32 +0000" action=0 End-of-block
Trace Country=gb Admin=" " Prmd=interspan
        arrival="Thu, 16 Aug 1990 14:15:23 +0000" action=0 End-of-block
Report-Hdr-End
Recip-No 1
DR-Failure Unable-To-Transfer Unrecognised-ORName
Arrival-Time "Thu, 16 Aug 1990 14:14:59 +0000"
Reported-Recipient-End
Report-End
```

Figure 7.3: Example Delivery Report Format

Appendix A

Example of the Tailor File

What follows is a suggested layout for the pp runtime tailor file. As the file is read in by most programs it is best if the number of long comments is kept to a minimum.

```
##############################################################
#
# Tailor file for PP
#
#       This version for sites with local smtp
#
##############################################################

##############################################################
#
# The following variables will need tailoring
#
##############################################################
#
# The loc_dom_mta is the host that the MTA will reside on and execute from
# It is a fully qualified domain name and represent the physical machine
#           Examples might be
#                       cheetah.nyser.net
#                       violet.berkeley.edu
#                       john.cs.nott.ac.uk
#
loc_dom_mta             lancaster.xtel.co.uk
#
# The Sitename is what the rest of the world sees you as. This is typically
# a domain shorter than the full domain name. It is the name used to represent
# your site or group as a whole. This name normally hides the individual
# machine names
#           Examples might be
#                       berkeley.edu
#                       nyser.net
#                       cs.nott.ac.uk
#
loc_dom_site            xtel.co.uk
#
# The postmaster should include the local site name
# e.g. postmaster@berkeley.edu or postmaster@cs.nott.ac.uk
#
postmaster              postmaster@xtel.co.uk
```

```
#
# The pplogin should be the username of the owner of pp.
#
pplogin             pp
#
# The qmgrhost should be the same as the name found in isoentities file.
# This can be the same as loc_dom_mta name
#
pptsapd_addr        LOCAL-ETHER=lancaster+20001
qmgrhost            lancaster
#
# The local or name should be the name of your site in X.400 terms.
# This is attribute value based, and starts with organisational units and
# works up to country. If you are unsure, leave it with just a country
# component, e.g. "/C=GB/"
#
loc_or              "/o=xtel/prmd=X-Tel Services/admd= /c=GB/"
###################################################################
#
# That is the end of the things you must tailor - the rest are
# probably ok - but you should check.
#
###################################################################

delim1              "\1\1\1\1\12"
delim2              "\1\1\1\1\12"
#returntime 36
#queuestruct        100
dap_user            "c=GB@o=X-Tel Services Ltd@cn=lancaster@cn=pp qmgr"


authchannel         free
authloglevel        high
wrndfldir           warnings
#submit_addr        lancaster:pp-submit

# Body types

headertype          822 822-jnt 822-us p2 822-uk p22 ipn g3fax ps
bodypart            ia5 dmd odif g3fax ipm voice iso6937 ia5-dk ps ms
bodypart            oid.0.9.2342.60200172.99.1.4


#  **-- Mapping tables --**
#
# All of these table MUST have the names given here
tbl     aliases                 show="Aliases:  mapping -> local id",
                                flags=dbm
tbl     users                   show="Users:  mapping local id -> disposition",
                                flags=dbm
tbl     x400-developers-aliases         show="Aliases:  mapping -> local id",
                                flags=linear
tbl     x400-developers-users           show="Users:  mapping local id -> disposition",
                                flags=linear
tbl     domain                  show="Mapping domain key ->full domain/MTA",
                                flags=dbm
tbl     or                      show="Mapping O/R Address -> MTA",flags=dbm
tbl     channel                 show="Binding MTA -> Channels",flags=dbm
tbl     or2rfc                  show="RFC 987:  X.400 -> RFC 822",flags=dbm
```

```
tbl     rfc2or          show="RFC 987:  RFC 822 -> X.400",flags=dbm
tbl     rfc1148gate     show="Gateways that perform RFC 1148 conversions",
                        flags=dbm
tbl     auth.channel    show="Authorisation:  channel policy",flags=linear
tbl     auth.mta        show="Authorisation:  mta based",flags=linear
tbl     auth.user       show="Authorisation:  user based",flags=linear
tbl     auth.qmgr       show="Authorisation:  qmgr control",flags=dbm
#  **-- Channel tables --**
#
# Needed by individual channels, not submit
tbl     local           file="ch.local",
                        show="local id -> user id + home directory",
                        flags=dbm

tbl     x400in84        file="ch.x400in84",
                        show="MTA -> connect info",flags=linear
tbl     x400out84       file="ch.x400out84",
                        show="X.400(84) connect info -> MTA + other info",
                        flags=linear
tbl     x400in88        file="ch.x400in88",
                        show="MTA -> connect info",flags=linear
tbl     x400out88       file="ch.x400out88",
                        show="X.400(88) connect info -> MTA + other info",
                        flags=linear
tbl     shell           file="ch.shell",
                        show="info for shell chan"
                        flags=dbm
tbl     list            file="ch.list",
                        show="Lists:  distribution lists",
                        flags=dbm
tbl     uucp            file="ch.uucp"
                        show="UUCP: rfc822 -> uucp",
                        flags=dbm
tbl   fax               file="ch.fax",
                show="FAX: mappings and info",
                flags=linear
tbl     mslocal         file="ch.mslocal",
                        show="Message Store:local id -> user id,home directory",
                        flags=linear

chan    822-local           prog=local,
        show="Local Delivery channel",type=both,
        sort="user time",adr=822,adr-order=usapref,
        hdrin=822,hdrout=822-us,
        bptin=ia5,bptout=ia5,
        content-out=822,outtable=local,access=mts,drchan=dr2rfc

chan    bugs-local          prog=local,
        show="Local Delivery channel for Bug Reports",type=both,
        sort="user time",adr=822,adr-order=usapref,
        hdrin=822,hdrout=822-us,
        bptin=ia5,bptout=ia5,
        content-out=822,outtable=local,access=mts,drchan=dr2rfc

chan    xua             show="XUA Submission Channel",type=in,
                        access=mts,adr-order=ukpref,
                        adr=x400, access=mts,

chan    x400in84        prog=x400in84,show="X400 inbound",type=in,
```

```
                        content-in=p2,adr=x400,
                        adr-order=ukpref,intable=x400in84

chan    x400in88        prog=x400in88,show="X400 inbound (1988)",type=in,
                        content-in=p22,adr=x400,
                        adr-order=ukpref,intable=x400in88

chan    x400out84       prog=x400out84,show="X400 outbound",type=out,
                        adr=x400,adr-order=ukpref,
                        content-out=p2,
                        hdrout=p2,
                        bptout="ia5,g3fax,odif",
                        outtable=x400out84

chan    x400out88       prog=x400out88,show="X400 outbound",type=out,
                        adr=x400,adr-order=ukpref,
                        content-out=p22,
                        hdrout=p22,
                        bptout="ia5,g3fax,voice,odif,iso6937,oid.0.9.2342.60200172.99.1.4",
                        outtable=x400out88

chan    mslocal         prog=mslocal,show="MS local delivery",type=out,
                        hdrout="p22,p2,ipn",
                        bptout="ia5,g3fax,voice,iso6937",outtable=mslocal,
                        sort="user time",access=mts

chan    mslocal-beta    prog=mslocalbeta,show="MS local delivery",type=out,
                        hdrout="p22,p2,ipn",
                        bptout="ia5,g3fax,voice,iso6937,oid.0.9.2342.60200172.99.1.4",outtable=mslocal,
                        sort="user time",access=mts

chan    smtp            prog=smtp,show="with SMTP (PP)",type=both,
                        adr=822,adr-order=usapref,drchan=dr2rfc,
                        bptout=ia5,hdrout=822-us,content-out=822,

chan    pac-janet       prog="greyout -P",show="via JANET with NIFTP (PP)",
                        type=in, adr=822,adr-order=ukpref,
                        hdrin=822-us,
                        outinfo=janet

chan    gb-janet    prog="greyout -P",show="via JANET with NIFTP (PP)",
                        type=both, adr=822,adr-order=ukpref,
                        hdrout=822-jnt,
                        bptout=ia5,drchan=dr2rfc,
                        content-out=822,outinfo=janet
#                       mta=much.cs.nott.ac.uk,

chan    gb-pss          prog=smtp,key=ipss,
                        show="via PSS with NIFTP (PP)",type=both,
                        adr=822,adr-order=ukpref,drchan=dr2rfc,
                        bptout=ia5,hdrout=822-jnt,
                        content-out=822,mta=much.cs.nott.ac.uk

chan    fax           prog=G3Fax, show="Fax channel",type=both,
                  content-in=p22, hdrout=g3fax, bptout=g3fax,
                        drchan=dr2rfc,outtable=fax, sort="user time",
#                 outinfo="in=/dev/ttyb,out=/dev/faxout,confirm,debug,master=gal@xtel.co.uk"
                  outinfo="in=/dev/ttyb,out=/dev/faxout,confirm,softcar,debug"
                        ininfo="in=/dev/faxin,softcar,confirm,debug,master=gal@xtel.co.uk"
```

```
chan    faxyuch        prog=G3Fax, show="Fax channel",type=both,
                               content-in=p22,
                bptout="g3fax,ia5", drchan=dr2rfc, outtable=fax,
                outinfo="in=/dev/ttyb,out=/dev/ttyb,master=faxmaster"

chan    dirlist                prog=dirlist,show="Directory List channel",type=both,
                               drchan=dr2rfc,adr=x400,adr-order=usapref,
                               sort="user time", outinfo="dn=C=GB@O=X-Tel Services Ltd@CN=lancaster"

chan    list                   prog=list,show="List channel",type=both,
                               adr-order=usapref,outinfo="linked",
                               outtable=list,drchan=dr2rfc,sort="user time"

chan    uucp                   prog=uucp-out,key="uucp-out,rmail",
                               show="UUCP channel",type=both,
                               adr=822,adr-order=ukpref,outtable=uucp
                               outinfo="uux=/tmp/uux,host=nott-cs",drchan=dr2rfc

chan    dr2rfc                 prog=dr2rfc,show="Dr2rfc channel",type=both,
                               adr=822,adr-order=ukpref,
                               bptin=ia5,hdrin=822,
                               outinfo="return=all order=uk",
                               content-out=822

chan    shell                  prog=shell,type=out,access=mts
                               show="Shell channel",
                               outtable=shell,sort="user time",
                               bptout="ia5,ps",hdrout=822-uk

# QMGR special channels (load/manage/clean)

chan    qmgr-load              prog=qmgr-load,show="Loading the QMGR",
                               type=qmgrload,outinfo="5";

chan    msg-clean              prog=msg-clean,show="Removing finished message"
                               type=delete

chan    trash                  prog=trash,show="Removing trash"
                               type=debris,outinfo=5h

chan    timeout                prog=timeout,show="Timeout messages"
                               type=timeout

chan    warning                prog=warnings,show="Send warning messages",
                               type=warn
chan    splitter               prog=splitter,type=split,
                               show="Divide message into single"
recipient messages
#  **-- shaper channels --**

#  Flatteners

chan    p2flatten              prog=p2flatten,show="Structured message -> P2",
                               type=shaper,content-out=p2

chan    822flatten             prog=rfc934,type=shaper,content-out=822

chan    p22flatten             prog=p2flatten,show="Structured Message -> P22",
```