Fido-Point-HOWTO

Roland Rosenfeld, roland@spinnaker.rhein.de, 2:2450/42 translated into the english by Oliver Much, Armageddon@icemans.ndh.com, 2:2453/690.12 *Revision* : 1.8 *Date* : 1996/09/1416 : 17 : 22

This file should help to set up a Fido-point under Linux, where ifcico will work as the mailer and FidoGate as the gatewaysoftware.

Contents

1 Where to get the latest version of this HOWTO

If you have an old version of this HOWTO you can request the latest version here (2:2450/42 ZYX and ISDNC/X75) with the filename `FIDOPNTE.GZ` (No mailbox only file requests). It is also available in the WWW at `<http://www.rhein.de/ roland/FidoPnt-en/>` or via FTP at `<ftp://sunrise.cs.uni-bonn.de/pub/linux/fido/FidoPnt-en.*>`.

There is also a german version available as `FIDOPNT.GZ` at 2:2450/111 and 2:2450/112 and at `<http://www.rhein.de/ roland/FidoPnt/>` and `<ftp://sunrise.cs.uni-bonn.de/pub/linux/fido/FidoPnt.*>`.

2 Programs we'll need

2.1 FidoGate

This is the gatewaysoftware, which will convert the Fido-messages into usenet-news/mail and vice versa. The latest release is version 3.9.7.

FidoGate is developed by Martin Junius (mj@fido.de, 2:2452/110).

You can find the latest version at <ftp://ftp.comnets.rwth-aachen.de/pub/mail+news/> and usually some time later at <ftp://sunsite.unc.edu/pub/Linux/system/Fido/>.

2.2 ifcico

This is a mailer similiar to BinkleyTerm, but it is more unix-like as it doesn't wait actively (which doesn't bother with Dos) and is very similiar to uucico from UUCP. ifcico comes with the package ifmail. The latest version is 2.8f. The older 2.8d keeps an ugly bug so don't use this version.

Ifcico was developed by Eugene Crosser (crosser@average.org, 2:5020/230).

The package ifmail comes with its own gatewaysoftware, but I prefer FidoGate and this HOWTO will only describe the configuration with FidoGate.

The latest version is uploaded by Eugene to <ftp://tsx-11.mit.edu/pub/linux/sources/usr.bin/> and <ftp://sunsite.unc.edu/pub/Linux/system/Fido/>

2.3 News-Transport-Agent (NTA)

As News-Transport-Agent one can use cnews as well as INN. To help you to choose one of them, here some of the pros and cons of both programs:

- cnews consists mostly of shell-scripts, which can't be debugged very good and work rather slow.
- cnews can't use NNTP (online-news-protocol). If you want to use NNTP (e.g. for various newsreaders, which can only communicate with the newssystem via NNTP), you needs a nntpd, whose configuration is quite hard and which doesn't support all the possibilities of NNTP.
- INN supports NNTP natively. That's why there is always a daemon (innd) running who takes care that the local news-articles are immeadiately readable by all users. Using cnews the articles are imported only after a specified time (e.g. every 10 minutes) to the newssystem.
- cnews will cause problems, if there is less than 10 MB free space in /var/spool/news

I myself used cnews in former times, but now I'm using INN as INN seems to be more handy. Besides that the configuration of INN was a lot easier than the one of cnews.

2.3.1 cnews

As older slackware-distributions came with a cnews that was extremly crippled (e.g. all man-pages were missing), one should use at least the cnews package from Slackware 2.1. As an alternative one can get hand on the complete source-distribution and compile it oneself. I used the performance-release from 93/02/20 from <ftp://ftp.uu.net/news/>. By now there's a newer one, but I haven't tried it yet. I will refer in this HOWTO to the older version and won't update this certain part in future.

2.3.2 INN

The latest Slackware-Distribution contains cnews as well as INN, so that one has the free choice. This INN should work, but I prefer to compile and configure my software myself, that's why I used the sources of INN-1.4sec, which can be found at `<ftp://sunsite.unc.edu/pub/Linux/system/News/inn1.4sec-linux-src.tgz>`.

2.4 Mail-Transport-Agent (MTA)

Here one has the choice between smail and Sendmail V8. smail is the smaller and simpler package, which has some small bugs, while Sendmail V8 seems to be not very easy to be configured. However there is a configuration with the macro-processor M4, which simplifies the configuration enormously so that it will be as simple as the configuration of smail. So, I myself changed from smail to Sendmail.

2.4.1 smail

As the old version 3.1.28.1 of smail had a bug in the uuname-driver (which is only important when using UUCP) I used version 3.1.29.1 from `<ftp://sunsite.unc.edu/pub/Linux/system/Mail/delivery/smail-linuxbin-3.1.29.1.tar.gz>`.
I don't use the configuration of smail in this HOWTO anymore as I upgraded to Sendmail. As Slackware contains now Sendmail instead of smail it shouldn't be too difficult to get rid off smail.

2.4.2 Sendmail V8

The contemporary version of Sendmail V8 is 8.7.4, but there shouldn't be any problems with version ¿8.6.9. Version 8.6.9 has a big security-gap, but this is only important if the computer is connected to the Internet via TCP/IP. One can find the latest version of Sendmail at `<ftp://FTP.CS.Berkeley.EDU/ucb/src/sendmail/>`, but as I already mentioned it's also a part of the Slackware-Distribution. It is very important to install the complete configurationfiles, which are hidden in a package with the confusing name `smailcfg.tgz`.

2.5 newspack

If you install cnews, INN or smail as sources, you should get the latest version of newspak. This package contains the adjustments of various mail- and newsprograms (cnews, INN, smail, sendmail-IDA, tin, trn, nn, elm, uucp,...) to Linux. Till now I haven't found a newspak, which contains also adjustments to Sendmail V8. However you find them in the Slackwaresources in the directory `n/sendmail`.

2.6 Newsreader

One can use any newsreader, e.g. tin, nn, trn, xvnews, xrn, slrn, knews,.. if one uses cnews, it should have a UUCP-mode, i.e. not only NNTP, as then one has to install the nntpd, which is quite difficult without any documentation. I suggest to use tin, as tin is easy to use and doesn't make much trouble when configuring it.

2.7 Mailreader

Here you can use what you want: elm, pine, mail, xmail, xmailtool, xfmail,...

2.8 Packers

Some packers come with the distributions. If special packers are missing one can get them from various servers like `<ftp://sunsite.unc.edu/pub/Linux/utils/compress/>` which have a packer-directory:

ZIP

unzip51 can unpack the new ZIP-format, too. There is now a zip, which can pack the new format.

ARJ

unarj241 unpacks ARJ-archives, but there's no packer for Linux.

LZH

lha existst as a Linux-port, too.

ARC

Old, but very important packer, as nodelists are packed with arc. There is at least one Linux-port.

RAR

unrar101 unpacks RAR-archives, I haven't found a RAR-packer, yet. One can find the sources to unrar at: `<ftp://ftp.kiae.su/.2/unix/arcers/unrar101.tgz>`.

2.9 TIC-processor

There are some small TIC-processors, which were written in PERL. Especially `tic010b.tgz` (also known as `lt010b.tgz`) by Cees de Groot (`cg@bofh.lake.de`, 241:10000/1512) can be mentioned. This program is part of the ifmail-package since version 2.8a and can be found in the directory `ifmail/misc/contrib/tic`.

There is also a TIC-processor included in FidoGate called ftntick. Last I have to mention the packet FidoTools-0.9 which can be found at `<ftp://ns.aanet.ru/vol1/nick/Linux/system/Fido/FidoTools-0.9.tar.gz>` and which mainly contains a tic processor.

3 Further documentation

News-HOWTO

for the configuration of cnews

Mail-HOWTO

for the configuration of mail

NET-2-HOWTO

for the configuration of the loopback-net

Network-Admin-Guide

for the configuration of all network-components, and UUCP.

Furthermore one should read the FidoGate-Documentation in the directory `fidogate/doc`

For INN the INN-FAQ is indispensable, as it not only answers most questions, but also accompanies the installation like a HOWTO.

4 My example configuration

As an example I will use my old configuration for the three FTN-nets Fido (zone 2), Gernet (zone 21) and Fido.de (zone 242). I have the following addresses:

- 2:2450/111.13 (Fido)
- 21:100/64.13 (Gernet)
- 242:5000/4.13 (fido.de)

I poll all three nets at the same uplink, but with different uplinks there wouldn't be many changes. Fido.de is an Internet-connection via the gate of Martin Junius in Aachen, Germany. From the Internet one can reach me at `roland@p13.flokiste.fido.de`. I send my Internet mails to `UUCP@242:4900/99`.

Please do not send me any mail to theses adresses. I changed my connections. But for this documentation I use them, as they're quite handy to explain the technique.

If you don't know an Internet gate yet, you should search for one, because FidoGate supports this execellently and assumes that you have an UUCP uplink or at least a connection to a gate.

Meanwhile I have found a direct Internet access at rhein.de for my computer, so that my computer can be reached as spinnaker.rhein.de via Internet. The computer rhein (it's located in the UUCP-worldmap so it doesn't need a domain, but nowadays this is an exception) supplies me with mail and news. I'll try to describe the configuration with Internet via a gate (fido.de) and via UUCP (rhein.de) parallely in this HOWTO, so that everybody should find his configuration.

Please note: Please do not use my adresses when you test your configuration. Take your own ones, else I will get the answers to your questions.

5 Permissions

The permissions of the files are a tricky topic in combination with FidoGate, ifcico and news. That's why you should have a look, if the users news and uucp exist in your configuration. My `/etc/passwd` looks like this:

```
uucp:sadfasdfasdf:5:5::/home/uucp:/bin/tcsh
news:aasdfasdfsda:24:24::/home/news:/bin/tcsh
```

Furthermore the groups uucp and news must exist. It seemed to be very helpful to insert the user news also in the group uucp and the user uucp also in the group news. From my `/etc/group`:

```
uucp::5:uucp,news
news::24:news,uucp
```

Also almost all spool-files should be group writeable, but we'll come back to this later on.

6 Installation of FidoGate

I refer to version 3.9.7, older versions don't support some configuration options.

At first we have to edit `fidogate/config.h`. I made the following changes in the original-file: (this is not the complete file, only the passages I changed!)

```
/* #define HOSTS_RESTRICTED */
#define SECURE
/* #define PASSTHRU_NETMAIL */
/* #define PASSTHRU_ECHOMAIL */
#define OUTDIR          "outbound"      /* Output of rfc2fido */

#ifdef SECURE           /* Secure permissions */
# define PACKET_MODE    0600            /* Mode for outbound packets */
# define BSY_MODE       0664            /* Mode for BSY files */
# define FLO_MODE       0664            /* Mode for FLO files */
# define DATA_MODE      0660            /* Mode for ffx data files */
# define DIR_MODE       0775            /* Mode for directories */
# define CONF_MODE      0664            /* Mode for written config files */
#else                   /* Open permissions */
# define PACKET_MODE    0666            /* Mode for outbound packets */
# define BSY_MODE       0666            /* Mode for BSY files */
# define FLO_MODE       0666            /* Mode for FLO files */
# define DATA_MODE      0666            /* Mode for ffx data files */
# define DIR_MODE       0777            /* Mode for directories */
# define CONF_MODE      0666            /* Mode for written config files */
#endif
```

You should see from the comments in `config.h` what this all means. There is no restriction to listed hosts. I've made it a habit of mine to call the outbound-directory `outbound` and not just `out`. You can change that if you do it consistently everywhere. Also I fitted the permissions to my flavour.

Next we have to make some changes in `fidogate/config.make`:

```
LIBDIR          = /usr/local/lib/fidogate
SPOOLDIR        = /var/spool/fnet
LOGDIR          = /var/log/fnet
OUTBOUND        = /var/spool/fnet/outbound
INBOUND         = /var/spool/fnet/inbound
PINBOUND        = /var/spool/fnet/pinbound
UUINBOUND       = /var/spool/fnet/uuin

OWNER           = uucp
GROUP           = uucp
DEBUG           = -O2 -s -fomit-frame-pointer
```

In `fidogate/src/Makefile` you should change `$(SPOOLDIR)/out` in `$(SPOOLDIR)/outbound`.

By this the directories are adjusted to my directory-structure and `uucp.uucp` becomes user of FidoGate. Later `ifcico` will work under the user `uucp.uucp`, too.

Now you've got to make some changes in `fidogate/src/ftninpost.pl`:

```
$RELAY    = "p13.flokiste.fido.de";
$PROTO    = "FIDOGATE";
$SENDMAIL = "/usr/lib/sendmail -oi -odq -oee -f%s -oMs$RELAY -oMr$PROTO -t";
$RNEWS    = "/usr/bin/rnews";
\&do_cmd("$LIBDIR/ftninrecomb $options");
```

Here the name of the gate is adjusted (it's not that important, but the name will appear later in the Received:-header of the mails). If you don't want to call Sendmail/smail every 15 minutes and don't start them as daemons, you should delete the queue-option (`-odq`) of sendmail, as else the mail will stay in `/var/spool/mail/input` (smail) or `/var/spool/mqueue` (sendmail). By setting the option `-oi` mails who have a line with only a fullstop in it are not terminated. Since FidoGate 3.9.5 this feature is built-in already. Then I adjusted the path of rnews to my system and finally activated ftninrecomb (in the original it's commented out by '##'). This option causes that mails which were splitted following FTS-0047 are glued together again.

Now you should execute the following commands in `fidogate/src` as `root`:

```
make depend
make all
make install-dirs
make install
```

Then you change the directory to `fidogate/lib` and run:

```
make install
```

By this all files you need are in `/usr/local/lib/fidogate`. Now you have to make some adjustments to your system. Here is a complete `/usr/local/lib/fidogate/config.common`, which contains the global configuration:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.common
#
# FIDOGATE config file common stuff,
# included by config.gate, config.main, config.ffx
#
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."
#

# Directories: lib, spool, BinkleyTerm-style outbound base dir (without
# the .../out.xxx), BinkleyTerm-style inbound dir
#
# lib, spool defaults are defined in config.h
#
LibDir          /usr/local/lib/fidogate
SpoolDir        /var/spool/fnet
LogDir          /var/log/fnet
Outbound        /var/spool/fnet
Inbound         /var/spool/fnet/inbound

#
# Internet address
#
#Hostname       p13
#Domain         .flokiste.fido.de
Hostname        spinnaker
Domain          .rhein.de
# Optional domain name for entries in HOSTS file
HostsDomain     .fido.de


#
# Zones and domains, the outbound directory is relative to the one
# specified with `Outbound'.
```

```
#
#               zone    Inet domain     FTN domain      Outbound
#               ----    -----------     ----------      --------
Zone            default .fidonet.org    fidonet         -
Zone            1       .fido.sub.de    fidonet         outbound.001
Zone            2       .fido.sub.de    fidonet         outbound
Zone            3       .fido.sub.de    fidonet         outbound.003
Zone            4       .fido.sub.de    fidonet         outbound.004
Zone            5       .fido.sub.de    fidonet         outbound.005
Zone            6       .fido.sub.de    fidonet         outbound.006
Zone            21      .ger.sub.de     gernet          gernet
Zone            242     .fido.de        fidode          fidode
#
# OPTIONAL:
#
# MSDOS client drive to UNIX server directory translation. This allows
# FIDOGATE running on a UNIX system and BinkleyTerm on an MSDOS PC.
#
#DosDrive       H:      /home
#DosDrive       I:      /var/spool
#DosDrive       I:      /usr/spool
#DosDrive       P:      /u1
#DosDrive       Q:      /u2
```

Have a close look to the following lines: This is the name of my computer in the Internet:

```
Hostname        spinnaker
```

This is my domain in the Internet:

```
Domain          .rhein.de
```

When I had only Internet-access via fido.de, I used `p13` as hostname and `.flokiste.fido.de` as domain.

If you have no Internet-address you can take any name as hostname and `.uucp` as domain (don't forget the leading point!). It is very important that you then define `NoFromLine` in `config.gate`, otherwise Internet addresses are generated on which nobody can answer. You should do this, too, if you have an Internet-address via a gate. fido.de e.g. bounces news-articles which contain such a FromLine, because it detects them as potential dupes.

Following changes are to be made in `config.main`:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.main
#
```

```
# FIDOGATE config file main AKA
#
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."
#

#
# Include common stuff (%L = LIBDIR)
#
include         %L/config.common

#
# FTN addresses
#
Address         2:2450/111.13
Address         21:100/64.13
Address         242:5000/4.13


#
# AREAS.BBS EchoMail distribution list
#
AreasBBS        /usr/local/lib/fidogate/areas.bbs

#
# FAREAS.BBS file distribution list
#
FAreasBBS       /usr/local/lib/fidogate/fareas.bbs

#
# ftnaf: CC of reply mails
#
CCMail          postmaster@spinnaker.rhein.de

#
# OPTIONAL for tosser
#
# Kill empty NetMails addressed to one of our AKAs
#
KillEmpty

#
```

```
# Kill EchoMail for unknown areas
#
#KillUnknown

#
# Kill routed EchoMail
#
#KillRouted

#
# Log sender/recipient of all NetMail messages
#
LogNetMail

#
# Check ^APATH for our own address
#
CheckPath

#
# Zonegate configuration for EchoMail
#
#               Address to      Add to SEEN-BYs
#               ----------      ---------------
#ZoneGate       2:123/456       123/456 2452/110
#ZoneGate       2:789/999       789/999 2452/110
#ZoneGate       1:105/42        105/42 2452/110

#
# Add extra nodes to SEEN-BY
#
#               Area            Nodes
#               ----            -----
#AddToSeenBy    ABLED           2:24/24
#AddToSeenBy    ABLENEWS        2:24/24
```

Last but not least the file `config.gate` has to be adjusted:

```
#:ts=8
#
# /usr/local/lib/fidogate/config.gate
#
# FIDOGATE config file gateway programs
#
```

```
# spinnaker.rhein.de
#
# Format:  keyword arg ...
#          keyword and args may be put in double quotes "..."

#
# Include common stuff (%L = LIBDIR)
#
include         %L/config.common

#
# FTN addresses - there must be a corresponding `uplink' statement
# for each `address' statement
#
#               real
#               ----
Address         2:2450/111.13
Address         21:100/64.13
Address         242:5000/4.13

Uplink          2:2450/111
Uplink          21:100/64
Uplink          242:5000/4

#
# Default origin line for EchoMail messages
#
Origin          " Spinnaker - Linux-Point der FloKiste Koeln "

#
# Organization header for News
#
Organization    "Spinnaker-FTN-UseNet-Gate"

#
# Don't generate From: line, FSC-0035 kludges
# MUST be set for FIDOGATE points (no real gateway)
#
NoFromLine

#
# FTN - Internet gateway. If set, Internet mail will be routed via FTN
# and this gateway.
#
```

```
#Gateway                242:4900/99

#
# Generate `User.Name@do.main' instead of `User_Name@do.main'
#
#DotNames

#
# X-FTN header:
#
#     f    X-FTN-From
#     t    X-FTN-To
#     T    X-FTN-Tearline
#     O    X-FTN-Origin
#     V    X-FTN-Via
#     D    X-FTN-Domain
#     S    X-FTN-Seen-By
#     P    X-FTN-Path
#
X-FTN           ftTOVDP

#
# Maximum size of FTN messages (default value)
#
MaxMsgSize              15000
#MaxMsgSize             32000

#
# MAUS gateway support:
#
# If set, convert addresses for `User_Name@XY.MAUSDomain' to
# `User Name % XY @ MAUSGate' for the Fido-MAUS gateway.
#
MAUSDomain              .maus.de
MAUSGate                2:2452/101.6

#
# CC of bounced messages
#
BounceCCMail            postmaster@spinnaker.rhein.de

#
# Send mail from message trackers to
#
```

```
TrackerMail             postmaster@spinnaker.rhein.de
```

Now we get on with the file `hosts`. The given example is specifically adjusted to the configuration of Martin Junius and can't be used by anybody else. You can delete it (create a new empty one with `touch hosts`), so that no names are used for the computer. But it should at least contain your own adresses:

```
# /usr/local/lib/fidogate/hosts
#
# Fields:
#
# node          FTN address
#
# hostname      Host name in the local domain (config: HostsDomain, Domain)
#               or fully qualified domain name (trailing `.') or `-' for
#               listed node without host name.
#
# options       -p  Generate addresses with point, e.g. `p99.hippo.fido.de'
#               -d  Node is currently down (can't receive mail)
#
# node            hostname                        options
#-----            --------                        -------
#
2:2450/111        flokiste                        -p
242:5000/4        flokiste                        -p
21:100/64         flokiste                        -p
```

Alternatively you can delete all lines not beginning with a `2:`, so only Fido-addresses remain and you should have no problems anymore. If you want you can use computers with other domains, as the example `tweety.dfv.rwth-aachen.de` shows. You shouldn't forget a dot a the end of a domain-name. This way you could use for instance the computer of `schiele-ct.de`, which also all have a Fido and an Internet address. If you want to insert only the sites of `schiele-ct.de` then you can turn `HostsDomain` to `schiele-ct.de` in the `config`-file and then insert all `schiele-ct.de` sites without the domain into `hosts`. I hope this became clear, as for all sites, which are not mentioned in `hosts`, the address is converted to e.g. `p13.f111.n2450.z2.fido.sub.de`, for the sites in `hosts` e.g. in `p13.flokiste.fido.de`. Don't forget to set the flag `-p` for almost all sites in `hosts`, to generate addresses with point. Otherwise the point number is deleted and this means work for the message-tracker of your node.

Next you can adjust the `bounce.*`-files to your system, but I can't remember that they were ever used. You can adjust `aliases`, too, but you don't need.

Now have a look at `areas`, which should at least contain all echomail areas, you use. If you enter more, this doesn't bother. Please note, that the german Fido-echos (e.g. `LINUX.GER`) are converted in the `fido.ger`-hierarchy (e.g. `fido.ger.linux`). International Fido-Echos are renamed into `fido.int.*`. The names of the Usenet-echos are converted to lower-case. Furthermore one should create an own hierarchy for any other Domain (zone). For all areas which do not belong to zone 2, you should add the zone-number (of course you can do this for all areas of zone 2, too). There are some other flags, too, which you can set. Here an excerpt from my `areas`-file:

```
# /usr/local/lib/fidogate/areas
#
# Echomail area <-> News newsgroup conversion
#
# Options:
#     -z ZONE            Alternate zone AKA for this area
#     -d DISTRIBUTION    Distribution header for this newsgroups
#     -o ORIGIN          * Origin line for this area
#     -g                 No gated messages
#     -l                 Only local crosspostings
#     -x                 No crosspostings
#     -8                 Messages with 8bit ISO-8859-1 charset
#     -H                 Names match entire hierarchy, names are translated
#     -!                 Don't gate area/newsgroup [hierarchy]
#     -R LVL             ^ARFC header level
#     -m MAXSIZE         MaxMsgSize for this area (0 = infinity)
#
# All fields may be quoted in "...", order is import, first area/newsgroup
# found matches!
#
# Format:
#
# area                        newsgroup                     [-option]
# ------------------------    --------------------------    ---------
FLOKISTE.INTERN               fido.flokiste.intern
FLOKISTE.STAT                 fido.flokiste.stat
LINUX.GER                     fido.ger.linux
SAILING                       fido.int.sailing
CT.GER                        ger.ct                        -z 21
CT.PROJEKTE                   ger.ct.projekte               -z 21
CT.PRUEFSTAND                 ger.ct.pruefstand             -z 21
CT.SUPPORT                    ger.ct.support                -z 21
FIDO.DE                       fido.de                       -z 242
5000.INTERNET                 fido.5000.intern              -z 242
ALT.RELIGION.EMACS            alt.religion.emacs            -z 242
COMP.OS.                      comp.os.                      -z 242 -H
REC.MUSIC.                    rec.music.                    -z 242 -H
DE.ALT.BINARIES.              de.alt.binaries.              -H -!
DE.                           de.                           -z 242 -H
```

As you can see it is possible to include whole hierarchies (`-H`) or to exclude sub-hirarchies (`-!`) for the usenet-areas, which have the same name under RFC (lower-case) and FTN (upper-case).

In the end you should create an empty file with `touch /usr/local/lib/fidogate/passwd` otherwise `rfc2ftn` won't work.

7 The installation of ifcico

I refer here to the latest version of ifcico, which can be found in the package `ifmail-2.8f`. Version 2.8d includes an ugly bug which did not appear in other versions, so don't use this one.

Next one has to edit the global compiler-configuration-file `ifmail/CONFIG`. Here all changes I made:

```
CONFIGFILE  = "/usr/local/lib/fnet/config"
LOCKDIR     = "/var/lock"
PUBDIR      = "/var/spool/uucppublic"
BINDIR = /usr/local/lib/fnet
OWNER = uucp
```

Maybe one has to change the paths, too. By this the debug-file is created in `/tmp` and the logfile in `/var/log/fnet`, which doesn't bother as the logfile since ifcico 2.3 is handled by `syslogd` (we'll come back to this later).

Slackware 3.0 continues to create this lockfile in `/var/spool/uucp`, which will cause problems with a already running getty on this port. You have to change `LOCKDIR` to `/var/spoool/uucp`) in this case.

Now one should create the directory `/use/local/lib/fnet` (user uucp, group uucp) and install the binaries in `/usr/local/lib/fnet` by executing

```
make depend
make all
make install
```

Now you have to install ifcico's config file. You will find an example in `ifmail/misc/config`. Copy it to `/usr/local/lib/fnet` and adjust it to your system-configuration. Here an example:

```
# /usr/local/lib/fnet/config
#
# IFmail - Configurations-File
# by Roland Rosenfeld 07.08.94

# Log file name. Overrides compile-time default.
logfile         /var/log/fnet/iflog

# Debug file name. Overrides compile-time default.
debugfile       /var/log/fnet/ifdebug

# Debugging verbosity level (is overidden by -x key). Default is 0.
```

```
# Following combination generates a debug-file with a length one can
# still read
verbose                 bcdefghijklmnqrt

# Main address:
address         2:2450/111.13@fidonet

# AKAs:
address         21:100/64.13@gernet
address         242:5000/4.13@fidode

# Passwords: Either you write the passwords directly in this file or
# you put them in an external one, which belongs to the user uucp and has
# the permissions 600, thus nobody can read them.

#include                /usr/local/lib/fnet/passwords
password        2:2450/111      SECRET
password        21:100/64       SECRET
password        242:5000/4      SECRET

# Directory for incoming packets/files:
inbound         /var/spool/fnet/inbound
# Directories for "listed" and "protected" sessions
listinbound     /var/spool/fnet/inbound
protinbound     /var/spool/fnet/inbound

# Directory for outgoing packets (default domain and zone):
# other zones will be like "/usr/spool/fnet/outb.003",
# other domains will be like "/usr/spool/fnet/<domain>.<zone>"
outbound        /var/spool/fnet/outbound

# Directory from which the file requests are satisfied
public          /var/spool/uucppublic

# Directory with executables to satisfy "magic" file requests
# if requested a file present in this directory, it will be
# executed and stdout sent to the remote system.  If the file
# is not executable, it is read line by line and the lines are
# processed as if they were received file requests (recusively).
# Execution of commands may compromize security!  You are warned.
magic           /usr/local/lib/fnet/magic

# First nodelist:
# The program searches for nl_short.xxx and the latest one will be used.
```

```
# I generate nl_short manually out of the complete nodelist, as the
# compilation of the complete list takes too long.
nodelist        /var/spool/fnet/nodelist/nl_short

# Further nodelists of other zones:
# (Attention: ifindex has a bug which leads to a Core, if one Nodenumber
# appears twice, e.g. using the nodelist and r24classic)
#               filename        originating address
nodelist        gernet          21:100/64.13@gernet
nodelist        242_list        242:5000/4.13@fidode

# Sequencer file (used to generate unique IDs)
sequencer       /usr/local/lib/fnet/seq

# At this place you find in the exaple-config-file (ifmail/misc/config)
# various explanations and options, which I mostly do not use and so
# won't explain here. But you should have a look at them.

# The modem is at /dev/ttyS1, the FIFO is locked at 38400 (without FIFO
# one should lock it a 19200).
# I use ttyS and not cua, as I use mgetty. Why I'm doing this is explained
# in the docs to mgetty+sendfax by Gert Doering.
ModemPort       ttyS1:L38400

# Now for the conversion of the telephonnumbers.I didn't really understand
# it, but the following should be correct (with your own local-access
# code) for whole germany. For special cases one should have a look at
# ifmail/misc/config.

PhoneTrans      49-2236-        /
PhoneTrans      49-     /       0
PhoneTrans      /       00

# In ModemReset I build in the correct AT-commands for my modem:
# AT Z0 = Initialize with profile 0
# AT M3 = Loudspeaker off after dialing
# If you use isdn4linux, don't forget to insert AT\&B512 to your
# initstring otherwise the connect will be lost after some bytes.
ModemReset ATZ0\r\n\dATM3\r\n\d

# Pulsedial:
ModemDial       ATDP\T\r
ModemHangup     ATZ\r
ModemOK         OK
```

```
# Instead of a simple "CONNECT" I take many Connect-strings, thus there is
# the correct speed of the connect in the logfile.
ModemConnect    CONNECT\s76800
ModemConnect    CONNECT\s57600
ModemConnect    CONNECT\s38400
ModemConnect    CONNECT\s19200
ModemConnect    CONNECT\s16800
ModemConnect    CONNECT\s14400
ModemConnect    CONNECT\s1200
ModemConnect    CONNECT\s9600
ModemConnect    CONNECT\s7200
ModemConnect    CONNECT\s4800
ModemConnect    CONNECT\s2400
ModemConnect    CONNECT\s12000
ModemConnect    CONNECT\r
ModemError      BUSY
ModemError      NO\sCARRIER
ModemError      NO\sDIALTONE
ModemError      NO\sANSWER
ModemError      RING\r
ModemError      ERROR

# Timeouts to wait for "OK" and "CONNECT", cannot be prefixed by logical
# expression.
TimeoutReset    3
TimeoutConnect  90

# Here you find further configuration-options for inbound-calls.
# As I do not admit any inbound-call I skipped this part, but in
# ifmail/misc/config you find infos for this.

# EMSI data for this node
# From this line on values CANNOT be prefixed with logical expression
# For now, escaping of '}' and ']' unimplemented, try to avoid these
# characters please!
Name            *** Spinnaker ***
Location        Bornheim-Widdig
SysOp           Roland Rosenfeld
Phone           - Unpublished -
Speed           9600
Flags           XA,V32B,V42B
```

To get the log-messages of `ifcico` you should configure syslogd correctly. Therefore you have to edit

/etc/syslog.config. The messages of ifcico appear as local0.*. For this I built in the following line:

```
local0.*                /var/log/fnet/ifmail
```

(Attention: Use **NO** spaces, but only TABS!)

By this all log-messages from icico go to /var/log/fnet/ifmail. If using an contemporary version of syslogd a

```
killall -1 syslogd
```

should inform the daemon that there is a new file, older syslogds need a complete restart.

Now we have to create the outbound-directories in /var/spool/fnet. Besides the standard outbound for Fido you have to create fidode and gernet for the other nets. The owner of these directories should be uucp.uucp and should have the permissions 775. ifcico can create directories for missing nets, but you have to adjust the permissions yourself.

8 Nodelist-compiler

All nodelists in the config-file should be in the directory /var/spool/fnet/nodelist. Then one calls ifindex, which then creates index.dir and index.pag, which contain the compiled nodelist.

As a matter of fact ifindex is not very quick, thus I shorten the nodelist to what I need (in germany) before the compilation. I use the following script for that:

```
#!/bin/sh
#
# /var/spool/fnet/nodelist/strip.nodelist
#
if [ ! $1 ]
then
    echo Syntax Error!
    echo Start with $0 xxx, whereas xxx is the number of the latest
    echo nodelist.
    exit
fi

NODELIST=nodelist.$1

if [ ! -s $NODELIST ]
then
    echo $NODELIST is not a correct nodelist
    exit
fi
```

```
NLSHORT=nl_short.$1

if [ -s $NLSHORT ]
then
    echo $NLSHORT already exists, moving to $NLSHORT.bak
    mv -f $NLSHORT $NLSHORT.bak
fi

head -n 2 $NODELIST >> $NLSHORT
grep ',49-\|^Zone,' $NODELIST >> $NLSHORT

echo Done.
```

You have to start this script with `strip.nodelist` *xxx*, where *xxx* stands for the number of the latest nodelist. The script generates `nl_short.`*xxx* from `nodelist.`*xxx*.

And what about the diffs? Since ifmail-2.6 there comes the program `nlpatch` with the package, which you call with the old nodelist and the latest diff, e.g.:

```
nlpatch nodelist.123 NODEDIFF.130
```

By this the nodelist of day 130 is created.

9 Poll-script and Unpacking

For polling I use the script `ifmail/misc/contrib/ifpoll` with small changes:

```
#!/bin/sh
# ver 0.7r
# ifpoll, poll my boss node or the node given as argument 1
#
# i start this shell script every day by crond, but you can
# start it also by hand :) start it as the owner of ifcico.
# rasca, berlin 1993 (Rasca Gmelch, 2:2410/305.4)
#
# Erweiterte Fassung von Roland Rosenfeld

# where "ifcico" and "ifpack" reside
FIDOPATH=/usr/local/lib/fnet

# logfile of ifcico
IFLOG=/var/log/fnet/ifmail
IFDEBUG=/var/log/fnet/ifdebug
```

```
# owner of "ifcico"
IFCICO_OWNER=uucp

# sysop of fido stuff
IFCICO_SYSOP=postmaster

# my boss node (default address to poll)
NODE="f111.n2450.z2"

# how often should i try to call NODE?
MaxTry=20

# delay between outgoing calls in seconds
DELAY=60

# where to log processing - file or tty/console
INFO_TTY=/dev/tty11

function GetConnectMessage () {
   grep 'chat got "CONNECT.*", continue' $IFLOG \
      | tail -n1 \
      | sed 's/.*chat got //;s/, continue//'
}

function GetNoConnectMessage () {
   grep 'chat got .*, aborting' $IFLOG \
      | tail -n1 \
      | sed 's/.*chat got //;s/, aborting//'
}

mv -f $IFDEBUG $IFDEBUG.old
touch $IFDEBUG

echo "`date \"+%b %d %T\"` ifpoll[$$]: starting" >> $INFO_TTY

# remember me, not to run as root..
#
if [ `whoami` != "$IFCICO_OWNER" ]; then
        echo "*** run $0 as the owner of ifcico ***"
        echo "`date \"+%b %d %T\"` ifpoll[$$]: wrong uid (rc 2)" >> $INFO_TTY
        exit 2
fi
```

```
# argv[1] is the optional node to call
#
if [ "$1" != "" ]; then
        if [ "$1" = "-?" ] || [ "$1" = "-h" ]; then
                echo "usage: ifpoll [node]"
                exit 3
        else
                NODE=$1
        fi
fi

# let's pack the fido stuff..
#

$FIDOPATH/fgpack

# loop until ifcico could connect the node or MaxTry is encountered
#
i=1; errlv=1
while let 'i <= MaxTry' && let 'errlv != 0'
do
        echo -n "`date \"+%b %d %T\"` ifpoll[$$]: $i. try ($NODE) " \
                >> $INFO_TTY
        #
        # start ifcico in master mode ..
        #
        $FIDOPATH/ifcico -r 1 $NODE
        errlv=$?
        if [ $errlv != "0" ]; then
                GetNoConnectMessage >> $INFO_TTY
                if [ $i != $MaxTry ]; then
                        sleep $DELAY
                fi
                let i=i+1
        else
                GetConnectMessage >> $INFO_TTY
        fi
done

# if the poll was fine, unpacking..
#
if [ $errlv = "0" ]; then
        echo "`date \"+%b %d %T\"` ifpoll[$$]: unpacking.. " >> $INFO_TTY
        $FIDOPATH/fgunpack $INFO_TTY
```

```
        # add here some additional lines for processing tic files or
        # incoming file-lists or simular..
        grep 'chat got .*, aborting' $IFLOG | \
                tail -n20 | \
                elm -s "ifpoll: failed" $IFCICO_SYSOP >/dev/null
fi

echo "`date \"+%b %d %T\"` ifpoll[$$]: finished (rc $errlv)" >> $INFO_TTY

# return the errorlevel of ifcico
exit $errlv
```

This script can be started without any parameters and then automatically calls my boss, or you start it with the node you want to call (in *pfnz*-format, e.g. `ifpoll f111.n2450.z2`). Of course one has to adjust this script to oneselfs needs (maybe not everybody has 12 consoles etc.)

This script is meant to call the packing-script:

```
#!/bin/sh
# /usr/local/lib/fnet/fgpack
# pack-script for FidoGate (together with ifcico)
#
# written by Roland Rosenfeld   19.08.94
#       roland@p13.flokiste.fido.de     (2:2450/111.13)
#

FNET=/var/spool/fnet
FTNPACK=/usr/local/lib/fidogate/ftnpack

PATH=/bin:/usr/bin:/usr/local/bin

topack=

for pkt in $FNET/*/????????.out $FNET/*/????????.pnt/????????.out
do
        if [ -s $pkt ]
        then
                topack="$topack $pkt"
        fi
done
$FTNPACK $topack
```

By this the `.out`-files are packed together in ARCmail-files, thus they can be transmitted more efficiently.

You have to define the used packers in `/usr/local/lib/fidogate/packing`, so `ftnpack` can find them:

```
# /usr/local/lib/fidogate/packing
#
# FIDOGATE ftnpack config file
#
# Commands:
#
#     arc    NAME  "/PATH/PROG %s %s"
#     prog   NAME  "/PATH/PROG %s"
#
#     pack   NAME  NODES
#     rpack  NAME  TARGET NODES
#     fpack  NAME  TARGET NODES
#

arc     zip     "/usr/bin/zip -gkjq %s %s"
arc     arc     "/usr/local/bin/arc an %s %s"

prog    gate    "/usr/local/lib/fidogate/ftn2rfc %s"

######## P A C K I N G ########################################################

pack    zip     *
```

Of course the paths of the packers have to be adjusted to your own needs. After a successful call with `ifcico` the following unpack-script is started:

```
#!/bin/bash
# usr/local/lib/fnet/fgunpack
#
# unpack-script for FidoGate (together with ifcico)
#
# written by Roland Rosenfeld   12.06.94
#       roland@p13.flokiste.fido.de     (2:2450/111.13)
#
# corrected paths, usage of syslog, more packers, clean up, restructured
#       martin@erde.GUN.de (Martin Seine) <2:2448/413.100>
#

FNET=/usr/local/lib/fnet
FGATE=/usr/local/lib/fidogate
IFCFG=$FNET/config
IN=/var/spool/fnet/in

# Facility is the log-file facility, where syslog stores the messages
```

```
# if you're not using syslog, no need to change it
FACILITY=local0

# System-manager who receives notices, if there are unpacking errors
MANAGER=postmaster

PATH=/bin:/usr/bin:/usr/local/bin:$FGATE

if [ $1 ]
then
        INFO_TTY=$1
else
        INFO_TTY=/dev/console
fi

WEARE=`basename $0`

#
# get directory-names from $IFCFG
#
INBOUND=`grep -i "^[    ]*inbound" $IFCFG | awk '{ print $2 }'`
LOGFILE=`grep -i "^[    ]*logfile" $IFCFG | awk '{ print $2 }'`

if [ ! -d $INBOUND/bad ] ; then
   if [ -e $INBOUND/bad ] ; then
        rm -Rf $INBOUND/bad
   fi
   mkdir $INBOUND/bad
fi

CORRECT=true

function iflog() {
   if [ -S /dev/log ] ; then
        logger -i -p $FACILITY.info -t $WEARE $@
   else
        echo "`date \"+%y/%m/%d %T\"` $$ $WEARE:" $@ >> $LOGFILE
   fi
   echo "`date \"+%b %d %T\"` $WEARE [$$]:" $@ >> $INFO_TTY
}


function unpack_mail() {
   pushd $INBOUND >/dev/null
```

```
EMPTY=true
for f in *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.su? *.pkt \
         *.MO? *.TU? *.WE? *.TH? *.FR? *.SA? *.SU? *.PKT
do
     if [ -f $f ] ; then
             if [ $EMPTY = true ] ; then
                     rm -rf bak.oo
                     mv -f  bak.o  bak.oo >/dev/null
                     mv -f  bak    bak.o  >/dev/null
                     mkdir  bak
             fi
             mv -f  $f bak/
             ln -s -f $INBOUND/bak/$f $IN/$f
             EMPTY=false
     fi
done

# now all new packets lie in $INBOUND/bak

if [ $EMPTY = true ]
then
     iflog No new mail found in $INBOUND
     exit
fi

popd >/dev/null

pushd $IN >/dev/null
#
# Analyze packer with file(1) and unpack them if possible
#
for f in *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.su?
do
    if [ -f $f ] ; then
       arc=`file -L $f | awk '{ print $2 }'`
       case $arc in
         ARJ) unarj e $f ;;
         ARC) arc e $f ;;
         ZIP) unzip -x $f < /dev/null ;;
         LHA) lha e $f ;;
         ZOO) zoo eq $f ;;
         RAR) unrar e $f ;;
           *) iflog unknown packer \'$arc\' for $f
              false ;;
```

```
          esac
          if [ $? -gt 0 ] ; then
              iflog couldn\'t unpack $f \(moved to $INBOUND/bad\)
              cp -f $f $INBOUND/bad
              CORRECT=false
          else
              iflog unpacked $f \($arc\)-archive
              rm -f $f
          fi
       fi
   done
   popd >/dev/null
}

#
#
# main unpack-program
#
#

export FNET
cd $FNET
unpack_mail

iflog starting ftn2rfc
ftn2rfc -x ftninpost -l

if [ $CORRECT = false ] ; then
/usr/lib/sendmail $MANAGER <<END
Subject: Fido-packet errors
There occured errors while processing Fido-Packets.

Please check the logfiles

Your Gateway
END
fi

#iflog starting Linux-TIC-Processor
#/usr/local/lib/tic/process_tics.pl < /dev/null
#/usr/local/lib/tic/poster.pl Daily < /dev/null
#/usr/local/lib/tic/lister.pl < /dev/null
```

This script moves all mails arrived in `/var/spool/fnet/inbound` to `/var/spool/fnet/inbound/bak` whereas these directories are moved to `bak.o` and `bak.oo`, thus one has always the last three poll-results, to fix any problems afterwards. Next all `*.pkt`-files are copied to `/var/spool/fnet/in` and the archives are unpacked (the script automatically recognizes ZIP, ARJ, ARC and LHA). For this the following lines must be in `/etc/magic` (put it at the top of the file as the file is scanned from top to bottom):

```
# Some entries for the ifcico-shell-script
#
0       byte            0x1a            ARC Archive (maybe)
0       string          PK              ZIP Archive
2       string          -lh             LHA Archive
0       string          ZOO             ZOO Archive
0       short           0xea60          ARJ Archive
```

If this is done, `ftn2rfc` is called, which will convert the packets to mail/news and after this starts `sendmail` and `rnews` via `ftninpost`.

After this other tools can be started, e.g. a TIC-processor or a program that scans the news for user of the own system.

10 The configuration of smail

If you chose Sendmail, you may skip this chapter.

In my system smail resides in the directory `/usr/local/lib/smail` and a link on it is in `/usr/lib/smail>`. The configuration-files are normally in `/etc/smail`, but some distributions have them in `/var/lib/smail`. All files described by me should be in this directory. The systemwide alias-file can be found follwoing the FSS (file system standard) in `/etc/aliases`, in former times in `/usr/lib/aliases`.

Let's begin with a basic configuration for the use w/o any UUCP-connection to the Internet:

```
# /etc/smail/config
#
#       smail configuration for p13.flokiste.fido.de
# (see smail(5) man page for details and other options)
#
-smtp_debug
hostname=p13.flokiste.fido.de
# more_hostnames defines other adresses, which smail should accept
#   for this site.
#   flokiste.fido.de has to be added, becauses there could be mails sent
#   to Roland Rosenfeld 2:2450/111, which are routed to me by the ITrack
#   of my uplink to me.
#   Hostnames with pxxx.fxxx.nxxx.zxxx-Namen are not needed, because
#   they all are mapped to p13.flokiste.fido.de in
```

```
#   /usr/local/lib/fidogate/hosts auf p13.flokiste.fido.de oder
more_hostnames=flokiste.fido.de:spinnaker:spi:p13
-visible_name
-smart_path
-uucp_name
error_copy_postmaster
postmaster=postmaster
```

If you have a direct UUCP-access to the Internet then you have to add the following lines

```
smart_path=rhein
smart_transport=uux
uucp_name=spinnaker.rhein.de
```

and to remove thelin `-smart_path`. In this case rhein is the UUCP-name of my uplink.

Now you have to define what is routed to where:

```
# /etc/smail/routers
#
# smail routers for p13.flokiste.fido.de

# See smail(5) for a complete description of the contents of this
# file.

fido:
        driver = pathalias,
        transport = fido;
        file = /etc/smail/paths.ftn,
        proto = lsearch

smart_host:
        transport=fido,
        driver=smarthost;
        path=p0.f4.n5000.z242.fido.de
```

Here everything is routed via Fido, where `paths.ftn` (l.d.) defines, where which FTN-address will be inserted. Everything that can't be processed based on `paths.ftn`, will be send to the fido.de-Internet-gate.

For the configuration with UUCP another Smarthost (where everything will be send that wasn't defined before) will be used:

```
smart_host:
        driver=smarthost,
        transport=uux
```

`paths.ftn` looks like this:

```
# /etc/smail/paths.ftn
#
.fido.sub.de   p0.f111.n2450.z2.fido.sub.de!%s
.z2            p0.f111.n2450.z2.fido.sub.de!%s
.fido.de       p0.f111.n2450.z2.fido.sub.de!%s
.maus.de       p0.f111.n2450.z2.fido.sub.de!%s
#
.ger.sub.de    f64.n100.z21.ger.sub.de!%s
#
.z242.fido.de  p0.f4.n5000.z242.fido.de!%s
```

Last but not least you have to define who mails will be transported:

```
# /etc/smail/transports
#
fido:   driver = pipe;
        group = uucp,
        cmd = "/usr/local/lib/fidogate/rfc2ftn -w Normal ${strip:user}",
        pipe_as_sender
```

As messages from the Fido-net should be addressed to the realname, it should exist (with an underscore between first and last name) on the system. Therefore you define it in `/etc/aliases`, thus the mail will be send to the user himself. Here an excerpt of my alias-file, where most of the aliases are defined, so that the mails always should reach me:

```
# /etc/aliases
#
# This is used by smail for sendmail compatibility.
# See also the /usr/lib/smail/lists directory for another way
# to make mail lists.
#
# Note: if your /etc/smail/directors says that the aliases
# director uses proto=dbm, you must use mkaliases to make
# /etc/aliases.pag and /etc/aliases.dir, which are the
# latest files used by smail.
#
# If you use proto=lsearch, smail will read this file.  (That's
# much slower but it's OK if this file is small.)
#
support: roland
```

```
admin: roland
Roland_Rosenfeld: roland
Roland.Rosenfeld: roland
postmaster: roland
faxadmin: roland
usenet: roland
sysop: roland
rosenfeld: roland
roro: roland
uucp: roland
news: roland
```

One should test the functionality of the file at first with a local mail as some versions of smail (if they were compiled with unfavorable parameters) seem to detect the file only as a dbm even if one defines `lsearch` (look at comment in `aliases`).So it's recommended to run `mkaliases` after each change in this file, which updates `aliases.dir` and `aliases.pag`.

As many error-messages of the newssystem will be send to `usenet`, `news` or `uucp`, one should route these messages to oneself.

11 Configuration of Sendmail V8

If you have installed Sendmail, you should install the whole set of configuration files which in the Slackware distribution is hidden in the package `smailcnf.tgz`. You'll find them in your directory-tree in `/usr/src/sendmail/cf`. In this directory there should be a approx.50KB big README which describes the new configuration-method.

Though Sendmail is controlled via an almost 20KB big configurationfile (`/etc/sendmail.cf`) you don't have to write it on your own (which most of the people think, who are scared of sendmail). You only need the macro processor M4, and a little configurationfile (mc-File) in `/usr/src/sendmail/cf/cf`. M4 will create your `sendmail.cf` from this file and a lot of other config files, which you do not need to change.

Next FidoGate has to be made known. Therefore create the file `/usr/src/sendmail/cf/mailer/ftn.m4` with the following content:

```
PUSHDIVERT(-1)
#
# /usr/src/sendmail/cf/mailer/ftn.m4
#
# FIDOGATE FTN mailer for sendmail V8
#
# MAILER(smtp) and MAILER(uucp) must be included!
#

ifdef(`FTN_MAILER_PATH',, `define(`FTN_MAILER_PATH',
```

```
                                   /usr/local/lib/fidogate/rfc2ftn)')
ifdef(`FTN_MAILER_ARGS',, `define(`FTN_MAILER_ARGS', `rfc2ftn -w normal $u')')
ifdef(`FTN_MAILER_FLAGS',, `define(`FTN_MAILER_FLAGS', `')')
ifdef(`FTN_MAX_SIZE',, `define(`FTN_MAX_SIZE', 100000)')
POPDIVERT
#####################################
###    FTN Mailer specification   ###
#####################################

VERSIONID(`ftn.m4 V1.5')

ifdef(`_MAILER_smtp_',
`# FIDOGATE mailer
Mftn,   P=FTN_MAILER_PATH, F=CONCAT(mDFMhu, FTN_MAILER_FLAGS),
        S=52/31, R=ifdef(`_ALL_MASQUERADE_', `11/31', `21'),
        A=FTN_MAILER_ARGS
 ')
```

As one can see the configuration for using rfc2ftn isn't that complex, and we don't have to care about what is exactly happening, we simply use the defined mailer "ftn" like the predefined mailers "smtp", "uucp" etc.

Now let's go on with the configuration of the mc-file. I called it `spinnaker.mc`, but the name isn't that relevant, it is only necessary that it resides in `/usr/src/sendmail/cf/cf`. `spinnaker.mc` looks like this:

```
# /usr/src/sendmail/cf/cf/spinnaker.mc
#
# sendmail V8 configuration for spinnaker.rhein.de
# using UUCP and FidoGate
#
include(`../m4/cf.m4')
VERSIONID(`spinnaker.mc V1.16')

OSTYPE(linux)dnl

define(`confUSE_ERRORS_TO', `True')dnl
define(`confCOPY_ERRORS_TO', `postmaster')dnl
define(`confMIME_FORMAT_ERRORS', `False')dnl

FEATURE(notsticky)dnl
FEATURE(mailertable,hash -o /etc/sendmail/mailertable)dnl
FEATURE(always_add_domain)dnl
FEATURE(nodns)dnl
FEATURE(nocanonify)dnl

MAILER(local)dnl
```

```
MAILER(ftn)dnl
MAILER(smtp)dnl
MAILER(uucp)dnl

Cwspi spi.rhein.de
Cwp13 p13.flokiste.fido.de
Cwp13.f111.n2450.z2.fido.sub.de
Cwp13.f64.n100.z21.ger.sub.de

# define(`SMART_HOST', uucp-dom:rhein)
define(`SMART_HOST', ftn:f4.n5000.z2.fido.de)

LOCAL_RULE_3
# Route fidonet.org and fido.sub.org via FTN!
R$* < $* z2.fidonet.org > $*            $1 < $2 z2.fido.sub.de > $3
R$* < $* fido.sub.org > $*              $1 < $2 fido.sub.de > $3
```

At the beginning the operating system (linux) is defined, then we define that all MAILERDAEMON-mails are also sent to postmaster, thus one can find errors faster and fix them. Then some options for my configuration are given, but you can overtake them for your needs. Furthermore the mailers "local" (for mail to user of the same computer, which are delivered by `deliver`, `mail.local` or `procmail`), "ftn" (Fido-compatible nets via FidoGate), "smtp" (transport via the online-mail-protocol SMTP) and "uucp" (transport via UUCP) are defined. The latter two ones aren't necessary for a system that is only connected with FidoGate to the Internet, but the mailer "ftn" is based on them (that's why `ftn.m4` is so short). In the lines beginning with `Cw` various names of my computer are defined.

By

```
define(`SMART_HOST', ftn:f4.n5000.z2.fido.de)
```

all mail that wasn't transported otherwise will be send via FidoGate to my fido.de-uplink. If you have an UUCP-uplink you can replace these lines with

```
define(`SMART_HOST', uucp-dom:rhein)
```

which sends all other mails to my UUCP-uplink rhein.

At the end of `spinnaker.mc` I've built in a special gimmick:

```
LOCAL_RULE_3
# Route fidonet.org and fido.sub.org via FTN!
R$* < $* z2.fidonet.org > $*            $1 < $2 z2.fido.sub.de > $3
R$* < $* fido.sub.org > $*              $1 < $2 fido.sub.de > $3
```

Here `z2.fidonet.org` is mapped into `z2.fido.sub.de` and `fido.sub.org` into `fido.sub.de`, thus all Fido-mails addressd to the various Fido-domains will be send to `fido.sub.de`, which is then send via FTN (instead of UUCP). Please note: There should be at least one TAB between `$*` and `$1` (not just blanks).

Therefore we need the above defined file `/etc/sendmail/mailertable`. In this file "exceptions" of the default routing are defined, e.g. the routing via Fido.

```
.fido.sub.de    ftn:f111.n2450.z2.fido.sub.de
.fido.de        ftn:f111.n2450.z2.fido.sub.de
.ger.sub.de     ftn:f64.n100.z21.ger.sub.de
.z242.fido.de   ftn:f4.n5000.z242.fido.de
```

As one can see all Fido-mails are transported via ftn. This mailertable has to be compiled to a database after any change. Use the following command in the directory `/etc/sendmail`:

```
makemap hash mailertable.db < mailertable
```

If you get the error "makemap: Type hash not supported in this version", you seem to have installed Slackware 3.0, which doesn't support the hash-type (don't ask me why). Then you should use the alternative command

```
makemap dbm mailertable < mailertable
```

in the same directory. But now you have to change the mailertable line in `spinnaker.mc` from `hash` to `dbm`. Change this line to

```
FEATURE(mailertable,dbm -o /etc/sendmail/mailertable)dnl
```

So your problems should go away.

After `spinnaker.mc` is ready edited, we can create `sendmail.cf` from it. Go to the directory `/usr/src/sendmail/cf/cf` and execute `pmake spinnaker.cf` there. By this `/usr/src/sendmail/cf/cf/obj/spinnaker.cf` is created. Move this file to `/etc/sendmail.cf` and the configuration is ready. In the case pmake isn't installed run this instead:

```
m4 spinnaker.mc > obj/spinnaker.cf
```

After this you have to create alias-file:

```
#
# /etc/aliases
#
# compile with
```

```
#       newaliases
#
nobody: /dev/null
support: roland
admin: roland
operator: roland
Roland_Rosenfeld: roland
Roland.Rosenfeld: roland
postmaster: roland
faxadmin: roland
usenet: roland
sysop: roland
rosenfel: roland
rosenfeld: roland
roro: roland
uucp: roland
news: roland
```

This file has to be compiled with the command `newaliases` to a database after any change.

Now start sendmail as a daemon by entering:

```
/usr/sbin/sendmail -bd -q20m
```

in `/etc/rc.d/rc.local`. On many systems it is already defined in `/etc/rc.d/rc.M` which is possible, too. Alternatively one can start sendmail by cron, and let inetd listen to the port. For this enter the following in the crontab of the user root:

```
*/20  * * * *   /usr/sbin/sendmail -q
```

and insert the folloging into your `/etc/inetd.conf`:

```
smtp  stream  tcp   nowait  root  /usr/sbin/tcpd  /usr/sbin/sendmail -bs
```

Don't forget to a `killall -1 inetd` after changing `inetd.conf`, to inform inetd about your changes.

Sendmail will write its logmessages via syslog. I recommend to enter the following entry in `/etc/syslog.conf`:

```
mail.*                      /var/log/mail
```

Do not forget to restart syslogd after this change with a killall -1 as already described in 7.

12 cnews-configuration

If you decided to use INN instead of cnews you can skip this chapter.

I'll refer here to the path-conventions which are used by the cnews of Slackware 2.1: `/var/lib/news` for configurationfile, logfiles and statusfiles and `/usr/lib/newsbin` for the cnews-binaries. In other distributions everything will be placed either in `/usr/lib/news` and `/usr/lib/news/bin` or in `/usr/local/lib/news` and `/usr/local/lib/news/bin`.

I'll list up here the most important configuration-files:

First the `sys`-file, which contains where the newsgroups are exported to:

```
# /var/lib/news/sys
#
# Attention: do not insert non-necessary blanks, this could cause
#            problems.
#
# ME: defines, which newsgroups you have on your own system.
#     You could write "all", but then you risk to automatically create
#     hierarchies (when having also an uucp-feed) you dont want to
#     have.
ME:alt,comp,news,gnu,de,rec,fido,ger,spinnaker,junk

# FidoGate (all FTN-nets, which are served via FidoGate):
# fido.*, ger.*, de.*, comp.*, rec.*, alt.*, gnu.*
fidogate/flokiste.fido.de:\
fido,de,comp,rec,alt,gnu/all:Lf:
```

We have to create an outbound-directory to get cnews to work. cnews expects this directory as `/var/spool/news/out.going/fidogate`. The owner of this should be `news.news` and it should have the permissions `775`.

Now we have to define how the news via FidoGate will be processed:

```
# /var/lib/news/batchparms
#
# site          size    queue   builder muncher sender
# ----          ----    -----   ------- ------- ------
/default/       100000  20      batcher compcun viauux
#
fidogate        250000  200     batcher nocomp  viafido
```

This means that the script `viafido` is used for the transport via FidoGate. Furthermore the maximal packet-size is set to 250K. If you have not much free space on your HD you may decrease the "`size`" (for the above you need at least 10MB of free space on `/tmp`), but this results in smaller `.pkt`-files.

Newer versions of cnews have a different format for this file:

```
fidogate    u     250000  200     batcher | nocomp | viafido
```

The file `/usr/lib/newsbin/batch/nocomp` seems to be missing in Slackware 3.0, but you can create with:

```
#!/bin/sh
# /usr/lib/newsbin/batch/nocomp
#
exec cat $*
```

The file `viafido` doesn't exist, you have to create it:

```
#! /bin/sh
# /usr/lib/newsbin/batch/viafido
#
# Submit news batch to FIDOGATE's rfc2ftn -w.
#
# The 'exec' cuts down the number of processes active for this simple case.

exec /usr/local/lib/fidogate/rfc2ftn -w Normal -b -n
```

After this the expire-list has to be configured, which defines, which articles are how fast to be deleted:

```
# /var/lib/news/explist
#
# The history sould be kept 30 days. Articles older than 30 days,
# reaching us are not inserted to the newssystem but simply deleted.
/expired/                       x       30      -

# Articels are kept minimal 3 days and maximal 90 days
/bounds/                        x       3-5-90  -

# The following list ist searched from top to bottom.
junk                            x       7       -
fido.junk                       x       7       -
control                         x       7       -
de.newusers.questions           x       8       -
de.newusers                     x       90      -
de                              x       14      -
comp.os.linux.announce          x       90      -
comp.os.linux                   x       8       -
fido.2450                       x       27      -
```

```
fido.r24                        x       10      -
fido.ger.linux                  x       18      -
fido.ger                        x       14      -

# default:  9 days (everything not matched above should be kept 9 days)
all                             x       9       -
```

By this the news-system should work, but some things have to be excecuted periodically, thus it works properly.

For this I use some cron-jobs. This is the crontab for the user `news`:

```
# crontab.news
#
SHELL=/bin/sh
MAILTO=news
#
# Crontab entries for news system
#
15 * * * * /bin/date >  /tmp/news_cron_ok

# Before both polls the news are collected and exported to fido.
# Additionaly the news are colleced and exported every hour.
27      8       * * *   /usr/local/lib/fidogate/run-batch
27      20      * * *   /usr/local/lib/fidogate/run-batch
13      *       * * *   /usr/local/lib/fidogate/run-batch

# New articles should be inserted into the newssystem every 5 minutes
*/5     *       * * *   /usr/lib/newsbin/input/newsrun

# Inspect the newssystem every evening
18      21      * * *   /usr/lib/newsbin/maint/newsdaily

# expire old articles every 12 hours.
24       8      * * *   /usr/lib/newsbin/expire/doexpire
24       20     * * *   /usr/lib/newsbin/expire/doexpire
```

`run-batch` is a little shell-script, which at first calls `newsrun` and then `sendbatches`:

```
#!/bin/sh
# /usr/local/lib/fidogate/run-batch
#
# Call newsrun, sendbatches
#
```

```
/usr/lib/newsbin/input/newsrun
/usr/lib/newsbin/batch/sendbatches
```

In Slackware 3.0 in every call of `sendbatches` the UUCP-queue is checked, where a error-message is given, as the system `fidogate` doesn't exist as a UUCP-system. To avoid this message you have to edit the script `queuelen` like this:

```
#!/bin/sh
# Find size of current queue of news outbound to $1.  Taylor version.
# =()<. ${NEWSCONFIG-@<NEWSCONFIG>@}>()=
. ${NEWSCONFIG-/var/lib/news/config}

PATH=$NEWSCTL/bin:$NEWSBIN:$NEWSPATH ; export PATH
umask $NEWSUMASK

if test "$1" = "fidogate"
then
   echo -e "\t0"
   exit 0
else
uustat -s $1 -c rnews | wc -l
fi
```

Now we have to enter the newsgroups into the newssystem. To do this job we use the program `/usr/lib/newsbin/maint/addgroup`, which takes the name of the newsgroup and the character `y` as parameters, e.g.

```
/usr/lib/newsbin/maint/addgroup fido.ger.linux y
```

By this the newsgroup `fido.ger.linux` (equals the fido-area `LINUX.GER`) is added. Do this for all your newsgroups. Then add the newsgroups `fido.junk` and `junk`, which will contain articles, which couldn't be sorted into another newsgroup. If Fido-echomail belongs to an area, which isn't defined in `/usr/local/lib/fidogate/areas`, it will be stored in fido.junk. If the area is defined in `fidogate/areas`, but you forgot to create the newsgroup with `addgroup`, then the article is stored to `junk`. As the newsadmin should subscribe this group to find out what groups are missing and to create them. To cancel articles you need the newsgroup `control`, which should be created with `addgroup`, too.

If there are problems with permissions a bug in `relaynews` could be the reason, as after an `su` the old user-ID and group-ID are used. The following patch should fix this problem:

```
--- org/relaynews.c     Tue Mar 17 08:42:42 1992
+++ relaynews.c Sun Jan 30 13:48:34 1994
@@ -201,7 +201,7 @@
        else
```

```
                newsuid = geteuid(), newsgid = getegid();
        if (setgid(newsgid) < 0 || setuid(newsuid) < 0 ||
-           getgid() != newsgid || getuid() != newsuid) {
+           getegid() != newsgid || geteuid() != newsuid) {
                if (getenv("NEWSPERMS") != 0)
                        error("recursive loop setting ids", "");
                /*
```

13 INN-configuration

First one has to install INN of course. You may use either a preconfigured binary or a self-compiled one, which shouldn't be too hard using the example-`config.date` from the newspak. I made the following changes (this is NOT the whole `config.data`, but only the differences between `config.data.newspak` and my configuration):

```
_EXITVAL                void
INEWS_PATH              DONT
INND_NICE_KIDS          DO
INND_NICE_VALUE         10
DEFAULT_CUTOFF          30
HAVE_UUSTAT             DO
_PATH_LOGFILE           /var/log/inn/log
_PATH_MOST_LOGS         /var/log/inn
```

I changed the paths so that for `NEWSPAK_NEWS_LIB_DIR /usr/local/lib/news` is used and for `NEWSPAK_SPOOL_DIR /var/spool/news`. As far as I know Slackware uses `/usr/lib/news`, so one has to make either a link or to make the apt changes.

After the installation you have to make some changes in `/usr/local/lib/news` and its subdirectories. I'll discuss the files to be configured one after another:

First some smaller files: `/usr/local/lib/news/passwd.nntp` and `/usr/local/lib/news/nntpsend.ctl` should only contain comments and nothing else. `/usr/local/lib/news/hosts.nntp` contains only the following line:

```
##  hosts.nntp - names and addresses that feed us news
##  Format
##      <host>:
##      <host>:<password>
##  <host> can be a name or IP address; no wildcards.  Any hosts not
##  listed here are handed off to nnrpd.
spinnaker.rhein.de:
```

So only my own computer is allowed to access my NNTP-server.

/usr/local/lib/news/nnrp.access should look like this:

```
##  nnrp.access - access file for on-campus NNTP sites
##  Format:
##      <host>:<perm>:<user>:<pass>:<groups>
##  Connecting host must be found in this file; the last match found is
##  used, so put defaults first.
##      <host>          Wildcard name or IP address
##      <perm>          R to read; P to post
##      <user>          Username for authentication before posting
##      <pass>          Password, for same reason
##      <groups>        Newsgroup patterns that can be read or not read
##  To disable posting put a space in the <user> and <pass> fields, since
##  there is no way for client to enter one.
##
## Default is no access, no way to authentication, and no groups.
*:: -no- : -no- :!*
##  Foo, Incorporated, hosts have no password, can read anything.
*:Read Post:::spinnaker*
localhost:Read Post:::*
spinnaker.rhein.de:Read Post:::*
```

So everybody has read and write access via NNRP (that's the protocol with that newsreader communicate with the newsserver) to the newsgroups, whose name begins with `spinnaker`. They are not allowed to read any other newsgroups. On my own computer everybody has the permission to read and write in every newsgrup.

Next you have to configure /usr/local/lib/news/inn.conf like this:

```
##  inn.conf -- inn configuration data
##  Format:
##      <parameter>:<whitespace><value>
##  Used by various programs and libinn.  The following parameters are defined:
##      domain          Local domain, without leading period.
##      fromhost        What to put in the From line; default is FQDN
##                      of the local host.
##      moderatormailer Where to mail moderated postings, if not found
##                      in the moderators file; see moderators(5).
##      pathhost        What to put in the Path and Xref headers; default
##                      is FQDN of the local host.
##      organization    If $ORGANIZATION doesn't exist.  What to put in
##                      the Organization header if blank.
##      server          If $NNTPSERVER doesn't exist.  Local NNTP server
##                      host to connect to.
##
```

```
organization:   private site, Widdig, Germany
#server:        spinnaker.rhein.de
server:         localhost
```

The most important file of the INN-configuration is `/usr/local/lib/news/newsfeeds` which defines, which newsgroups are exported to whom.

```
##  newsfeeds - determine where Usenet articles get sent
##  Format:
##      site[/exclude,exclude...]\
##              :pattern,pattern...[/distrib,distrib...]\
##              :flag,flag...\
##              :param
##  Summary of flags:
##      <size           Article must be less then size bytes.
##      Aitems          Article checks -- d (must have Distribution header)
##                      p (don't check for site in Path header).
##      Bhigh/low       Internal buffer size before writing to output.
##      H[count]        Article must have less then count hops; default is 1.
##      Isize           Internal buffer size (if a file feed)
##      Nm              Only moderated groups that match the patterns.
##      Nu              Only unmoderated groups that match the patterns.
##      Ssize           Start spooling if more than size bytes get queued.
##      Ttype           Feed types -- f (file) m (funnel; param names the
##                      real entry) p (pipe to program) c (send to stdin
##                      channel of param's sub-process); x (like c, but
##                      handles commands on stdin).
##      Witems          What to write -- b (article bytesize) f (full path)
##                      g (first newsgroup) m (Message-ID) n (relative
##                      path) s (site that fed article) t (time received)
##                      * (names of funnel feed-in's or all sites that get
##                      the article) N (Newsgroups header) D (Distribution
##                      header) H (all headers) O (overview data) R
##                      (replication data).
##  Param field depends on T flag.  For Tf, relative paths are from the
##  out.going directory.  For Tp and Tc, it is a shell command to execute.
##  If a Tm refers to this entry (which will have its own T param) then "*"
##  is expanded to all the funnel sites that triggered this one.  Useful
##  for spawning one mail process, e.g.
##
##  This file is complicated -- see newsfeeds.5!

# ME shows, which newsgroups should be accepted
ME\
```

```
        :!*\
        ::

# Everything, which should be sent via FidoGate:
# - fido.* except fido.junk, because this is a pseudo newsgroup where
#   only those articles are stored which are not entered correctly
#   into FidoGate's areas file,
# - ger.*, the gernet-newsgroups
# If Internet is received via fido (e.g. via fido.de), the Internet
# newsgwroups (rec.*, comp.*, de.*, alt.*, gnu.*,...) have to be
# inserted here, too.
#
# All my ftn-uplinks have the same name (configured in
# /usr/local/lib/fidogate/hosts), here it is flokiste.fido.de.  So
# this name has to be added here, otherwise articles coming here via
# flokiste.fido.de would be sent back es dupes to flokiste.fido.de.
#
# To keep Fido and Internet seperatly (I don't want that any posting
# received via Fido is exported via Internet and vice versa), I
# inserted rhein as an additional alias for this system.
fidogate/flokiste.fido.de,rhein\
        :fido.*,!fido.junk,ger.*\
        :Tf,Wnb:

# This is my Internet (usenet)-feed.
# Here we export the areas comp.*, de.*, rec.*, alt*, gnu.*,...
# To suppress exporting articles which originally come from Fido, I
# added flokiste.fido.de as an alias for rhein.
rhein/flokiste.fido.de\
        :comp.*,de.*,rec.*,alt.*,gnu.*\
        :Tf,Wfb:
```

Then you have to configure `/usr/local/lib/news/expire.ctl`, which defines when which newsgroups expire.

```
##  expire.ctl - expire control file
##  Format:
##      /remember/:<keep>
##      <patterns>:<modflag>:<keep>:<default>:<purge>
##  First line gives history retention; other lines specify expiration
##  for newsgroups.  Must have a "*:A:..." line which is the default.
##      <patterns>      wildmat-style patterns for the newsgroups
##      <modflag>       Pick one of M U A -- modifies pattern to be only
##                      moderated, unmoderated, or all groups
```

```
##      <keep>          Mininum number of days to keep article
##      <default>       Default number of days to keep the article
##      <purge>         Flush article after this many days
##  <keep>, <default>, and <purge> can be floating-point numbers or the
##  word "never."  Times are based on when received unless -p is used;
##  see expire.8

##  If article expires before 14 days, we still remember it for 14 days in
##  case we get offered it again.  Depending on what you use for the innd
##  -c flag and how paranoid you are about old news, you might want to
##  make this 28, 30, etc.
/remember/:30

# Keep all newsgroups minimal 3, maximal 90 and normaly 5 days (from
# arrival)
*:A:3:5:90
*:U:3:5:90
# Keep moderated newsgroups a bit longer
*:M:5:9:90

# Keep local newsgroup 3 months.
spinnaker.*:A:3:90:90

# Keep german newsgroups 7 days
de.*:A:3:7:90
# de.comp.os.linux.* a bit longer.
de.comp.os.linux.*:A:3:14:90

# Keep the moderated c.o.l.* 3 months.
comp.os.linux.*:M:5:90:90

# Keep fido some longer as default.
fido*:A:3:9:90
ger*:A:3:8:90

# Remove the junk very quick.
fido.junk:A:3:4:90

# And another exception:
fido.ger.linux:A:3:24:90
```

Of course you may define this on your own needs. But you shouldn't decrease the default-times (especially for Fido) too much, as the messages take a long time to your system and so might expire immeadiately.

Now create the file `/usr/local/lib/news/send-fidogate` that will periodically pack the Fido-echomail. You can find a prototype of this file in `fidogate/config/inn/send-fidogate`, but you have to make some little changes in the beginning of the file:

```
#! /bin/sh
#:ts=8
##  SH script to send batches to FIDOGATE

##  =()<. @<_PATH_SHELLVARS>@>()=
. /usr/local/lib/news/innshellvars

#
# FIDOGATE rfc2ftn
#
RFC2FTN="/usr/local/lib/fidogate/rfc2ftn -w Normal -b -n "
```

To get the news exported, there must be a batch-directory for them, which is called `/var/spool/news/out.going`, with owner `news.news` and permission `775`. In contrast to cnews there mustn't be any subdirectories in this directory for the newsfeeds, as INN uses only one file per feed which it creates itself. If you "upgrade" from cnews to INN, you should delete the existing directories.

As INN works with a daemon, which always runs, it must be started in the boot-process. Add the following at the end of `/etc/rc.d/rc.M` before the call of `rc.local`:

```
/usr/local/lib/news/etc/rc.news \&
```

For the first test of INN this script can be started manually (as user root). But be careful, that only one innd is running at the same time.

In some distributions another NNTP-daemon (nntpd) is defined for cnews in `/etc/inetd.conf`. Then innd can't be started, as the TCP-port is already busy. That's why you have to comment this line out and restart inetd with a kill -1.

To make INN work properly you have to add the following to the crontab of the user news:

```
SHELL=/bin/sh
MAILTO=roland
#
# INN:
#
13  *   * * *   /usr/local/lib/news/send-fidogate
17  *   * * *   /usr/local/lib/news/send-uucp rhein
13  21  * * *   /usr/bin/nice /usr/local/lib/news/bin/news.daily delayrm
48  21  * * *   /usr/local/lib/news/rnews -U
```

Every hour the batches for all Fido-systems and the UUCP-system rhein are packed (without a UUCP-connection you can skip the last part of course). Furthermore there's a daily clean-up, which is reported via mail. Then all accidentally not-delivered articles will be processed.

The next thing we've got to do is creating the newsgroups. Therefore and for almost all other changes in the newssystem only the command `ctlinnd` is used. It should be called only by the user news and you can find it in the directory `/usr/local/lib/news/bin`.

New newsgroups are created with `ctlinnd newgroup fido.ger.linux` (where `fido.ger.linux` is only an example). You can use various options, which are explained in the manpage of `ctlinnd(8)`. You should create the newsgroups `control`, `junk` and `fido.junk`, which contain the control-messages (e.g. cancel), articles for unknown newsgroups and articles for not in `/usr/local/lib/fidogate/areas` defined areas.

To find the descriptions of the newsgroups in the newsreader, you may want to create the file `/usr/local/lib/news/newsgroups`, which contains the name of a newsgroup and its description (seperated by a TAB) in every line. The order of the groups is irrelevant, from a duplicate line only the first one is shown.

As INN reports his log-messages via syslog, one should make these changes in `/etc/syslog.conf`:

```
news.crit                               /var/log/inn/news.crit
news.err                                /var/log/inn/news.err
news.notice                             /var/log/inn/news.notice
```

Furthermore create the directory `/var/log/inn`.

For further information on INN, an upgrade from a cnews-system to INN, problems etc. please read the very good INN-FAQ and the manpages.

14 News/Echomail reading and writing

I myself prefer `tin` as the newsreader. As tin, like all other newsreaders wasn't constructed for Fido, there's no standard-method to adress echomail to someone, which is possible and usual in Fido. FidoGate allows this this via a additional headerline `X-Comment-To:` You can create it manually but do not forget the username in brackets. You also may use a complete Internet-adress including a realname.

Examples:

```
X-Comment-To: (Roland Rosenfeld)
X-Comment-To: rosenfel@uran.informatik.uni-bonn.de (Roland Rosenfeld)
```

In the directory `fidogate/contrib/news` you find patches for `tin`, `xrn` and `nn`, which generate this line automatically, if you answer to an article via echomail. If you don't want to patch `tin`, you can insert the following line in `~/.tin/tinrc`:

```
news_quote_format=X-Comment-To: (%N)\n\n%F wrote in msg %M:
```

but then don't forget to delete the emptyline before the `X-Comment-To:` in every follow-up.

If you already use one of the beta-versions of tin (from `tin-1.3beta-950824`) you can the following in `~/.tin/attributes`:

```
scope=*
auto_save_msg=OFF

scope=fido.*
x_comment_to=ON

scope=ger.*
x_comment_to=ON
```

by this you'll find in the Fido- and Gernet-echos a `X-Comment-To:`-Header without patching tin.

As an alternative you can configure the newsreader since FidoGate 3.9.5 so that one of the following lines stands in the message. The realname will then retrieved from this:

```
user@do.main (User Name) writes:
User Name <user@do.main> writes:
user@do.main.writes:
user@do.main (User Name) wrote:
User Name <user@do.main> wrote:
user@do.main wrote:
In article <id@do.main> user@do.main writes:
In article <id@do.main>, user@do.main writes:
```

15 Mail/Netmail reading and writing

I prefer `elm` for doing this. You don't have to do something special, you only have to use the pfnz-notation for adressing. So

```
Roland Rosenfeld 2:2450/111.13
```

becomes

```
Roland_Rosenfeld@p13.f111.n2450.z2.fido.sub.de
```

or

```
Roland_Rosenfeld@p13.f111.n2450.z2.fido.sub.de (Roland Rosenfeld)
```

And

`Roland Rosenfeld 21:100/64.13`

becomes

`Roland_Rosenfeld@p13.f64.n100.z21.ger.sub.de`

Internet adresses can be used directly and for MAUSnet systems you use the apt Internet-address, e.g. `Firstname_Lastname@su.maus.de`. Despite that the mail is send via the Fido-Maus-gate (if you haven't defined something else in `/fidogate/src/config.h`).

For Maus-messages the file `/usr/local/lib/fidogate/maus` should always be up to date, thus the different Maus-domains will recognized and converted correctly (not all Mice are accessable via `.maus.de`). FidoGate comes along with the latest version of the file you can find in `fidogate/lib/maus`.

Another important point is sending mails to tools (Areafix, FileMgr,...). These often have problems with mails created by FidoGate. As the mails often have a "`* To: address`" in the first line, which some of the tools interpret as a command to order all Areas ("`*`").If you don't want such a line, you can enter in `elm` in the menu **H**)eader with **U**)ser-Defined-Header (before sending the message) an additional header-line:

`X-Flags: N`

Another problem is that FileScan and some other tools have the bad habbit to take the address not from the header, but from the message-ID. But this doesn't contain the FTN-address (Z:N/F.P), but the Internet-address. You can fix this with:

`X-Flags: M`

by this the usual Fido-message-ID is generated (for different nets with the apt address).

Of course you can combine the X-flags. I use for instance for all messages to robots:

`X-Flags: MN`

If you want to send crashmails, you have to add an additional "`X-Flags: C`". If you address to a point, the mail is crashed to his boss.

FileAttaches are possible, too. Add a "`X-Flags: F`" and a filename (with path!) as subject.

If you want to set flags for all mails, you have to create a file like `~/.elm/elmheaders`, which only contains the headerline, e.g.

`X-Flags: MN`

This line will be added to all messages.

If the X-Flags show now effect, have a look into the FidoGate-logfile, as FidoGate will probably remark that `Hostname.Domain` in `/usr/local/lib/fidogate/config.common` is not identical with the name of the computer FidoGate runs on. But this is a must for the acceptance of the X-Flags, else everybody could create crashmails and filerequests and send them from your site.

16 Requesting files

For filerequests one can use the perl-script `ifreq` which is located in `/ifmail/misc/contrib`. Some DOS-request-processors have a problem with a missing linefeed at the line's end. This patch should fix this:

```
--- ifreq~      Sat Aug  6 14:41:28 1994
+++ ifreq       Sat Oct  8 12:05:09 1994
@@ -73,7 +73,7 @@
# open the flofile for appending
open(FLOFILE, ">>" . $reqfile) || die "can't open $reqfile";
while (@files) {
-                       print (FLOFILE shift(@files), "\n");
+                       print (FLOFILE shift(@files), "\r\n");
}
close(FLOFILE);
```

17 Troubleshooting

If this all doesn't work right (despite my explanations), you have different possibilities to find the mistakes and fix them. An important tool for doing this is `ifstat`, which reports which packets are on hold for which addresses. Don't get confused, if these packets shrink after starting `ifpoll`, because this packs the mails into `ZIP`-Files.

17.1 Mail is not processed

It would be a good idea to raise the debug-level of rfc2ftn in `/etc/smail/transports` resp. in `/etc/sendmail.cf`. To do this just add the parameter `-v` in ytt/rfc2ftn/ (you can use more than one `v` to raise the debug-level, e.g. I use something like `-vvvvvvvvvvvvvvvvvvvvvv`). To really get hold of the debug-messages I send the mail with a copy to a non-existent address so that I get the returned message with all the debug output. If such a returned message doesn't arrive it's quite possible that some of the permissions aren't correct which causes the system to wait (the still running processes `sendmail` and `rfc2ftn` indicate this). Then you find the latest debug-file in `/var/spool/smail/msglog`. In the directory `/var/spool/smail` you find the "stranded" messages. Using sendmail the log-messages are written via syslog into the syslogfile and the "stranded" messages lie in `/var/spool/mqueue`.

17.2 News are not processed

The fix to this problem is similiar to the one we discussed already for the mail. I build in `/usr/lib/newsbin/batch/viafido` (cnews) resp. in `/usr/local/lib/news/send-fidogate` (INN) some `-v`'s and append a `2>/tmp/fgate.out` in the apt line, which causes that the output of stderr is written to `/tmp/fgate.out`, which one can look at and search for the problem.

Another good idea is to look at `/var/spool/news/out.going/*/togo` between calling newsrun and run-batch if the article was listed and deleted afterwards. Sometimes (depending on the version of cnews) the articles are saved to `/var/spool/news/in.coming`.

INN should take the article immediately into the newssystem thus they should appear in the file `/var/spool/news/out.going/*`. After running `send-fidogate` they should disappear there and you should find them in the Fido-outbound.

Furthermore one should have a look at all the logiles:

smail

`/var/spool/smail/log/logfile`

Sendmail

`/var/log/mail`

cnews

`/var/lib/news/log.*` `/var/lib/news/batchlog.*` `/var/lib/news/errlog.*`

INN

`/usr/local/lib/news/log` `/usr/local/lib/news/errlog` `/var/log/inn/*`

fidogate

`/var/log/fnet/log`

ifcico

`/var/log/fnet/ifmail` `/var/log/fnet/iflog` `/var/log/fnet/ifdebug`

To verify the created packets or fix problems in already existing packets FidoGate comes along with the program `pktdebug`. If you need more information you should get hold of the DOS-program Inspect which runs perfectly in the DOSEmu. When you allow dosemu to access the apt directories and start DOSEmu as user `uucp` you can even change the `.pkt`-files and archives.

If cnews completly hangs up, you have to debug the various cnews-shell-scripts. To do this just add a "`set -xv`" in the beginning of the script which then reports which line of the script is called with which parameters. For further information have a look at the manpages of the `bash`.

In the case only the posting of news-articles doesn't work try to post with `Pnews` from the `trn`-package directly. In contrast to tin this is only a simple shell-script which alway waits until you've read the error-messages of cnews (`tin` overwrites them with a menu).

18 Credits

The following people helped me with their comments and improvements to write this HOWTO:

- `andij@andi.tricbbs.fn.sub.org` (Andreas Jellinghaus)

- `uli@corno.cologne.de` (Ulrich Villers) 2:2450/115.3
- `martin@erde.GUN.de` (Martin Seine) 2:2448/413.100
- `piet@smallchange.rhein.de` (Peter Theisohn) 2:2450/30
- `mg@genyosha.in-chemnitz.de` (Mike Gaertner) 2:249/5060.17

If you have any additions or error-reports then send them to me!