**MBSE BBS System Guide 1.0.6**

Document History.
Version 1.27 – Taken from HTML files August 2015 with spelling and other typo's fixed.
Date of last revision : 17/10/2015 17:35

0X12456789

# Table of Contents

# Introduction to MBSE BBS.

**Distribution.**

ThIs is the only official distribution sites for the mbse bbs package. Which is:

1. https://sourceforge.net/projects/mbsebbs/

New versions of mbse bbs are announced in the fidonet area MBSE & sometimes LINUX_BBS support echos. On the official fidonet nodes you can request the latest version with the magic mbse and/or MBSEBBS. You will then get a zip file, in this zip file is the original tar.gz file. This is to let systems who only support 8.3 file names to pick up the distribution package.

**History**.

At the end of 1997 I was looking for several BBS systems that could run on GNU/Linux and it must be capable to run Fidonet mail. After reviewing almost all packages that were available at that time I found that there were no packages that suited my needs. Some had the plain user interfaces that my bbs users were used to but no Fidonet capabilities, others looked awful or were difficult to use by normal bbs users without Unix experience. I also didn't want to run shareware any more, one day you pay for some program, and the next day support is over because the writer of that program decided to stop development or simply disappears from the Fidonet stage. With all Y2K problems ahead the solution should be Open Software so that you have the sources in case something goes wrong. One package was very interesting and had the look and feel of RemoteAccess, that package was RapidBBS. There was only one problem, it had no Fidonet capabilities. I rewrote the data structures and created a deamon that should control all bbs activities. In March 1998 I started writing the mbfido program that should handle all Fidonet mail and .tic files. In June 1998 the final message base format became JAM using the LoraBBS sources as a guide to create the JAM libraries. The original JAMapi was not stable enough to do all the work that needed to be done.

In July 1998 the first version of MBSE BBS was installed on the bbs I have, on the second modem line. The first line was running McMail, GEcho and RA on a Novell client while on the GNU/Linux box the mars_nwe emulator from Martin Stower was running. In November 1998 mbcico was created from ifcico from Eugene M. Crosser. In January 1999 it also compiled and run on a Sun Sparcstation 2 system.

In April 1999 the motherboard of the GNU/Linux server died, I replaced it with the Mobo of one of the client machines. From that day on, MBSE BBS became the only bbs running on my system, because I was short on serial port boards at that time. McMail and RA became history and MBSE BBS was on its own. From that day on, updates were almost daily, all users and up and downlinks showed that there were plenty of bugs to solve.

One month later most problems were solved.

In July 1999 Jan van de Werken started beta testing MBSE BBS on his system. In September 1999 MBSE BBS was public released for the first time.

**Is it Y2K ready?**

There have been no problems since 1 January 2000 with MBSE BBS. I do run pktdate by Tobias Ernst in the tosser, this solves problems with incoming mail. Due to the internal date format, this program should run until 2038, just as long as Unix/Linux and the internet will function without changing the date format - Micheal.

As a follow on here – I have been running mbse for over ten years without using extra date processing (e.g., using pktdate) with no problems so this can be ignored - Vince.

# Known bugs.

There are always bugs, but these ones are known....

- Reading of function keys in mbsebbs doesn't always work well, especially on slow links and over PPP. This is not a MBSE BBS problem.
- Problems with older D'Bridge [1a] mailers.
- If you have regular sessions with a node which only supports FTS-0001 sessions and you use a session password you **must** also set a mail password and these passwords must be the same. This is a side effect of the way FTS-0001 handshake works, by sending a small mail packet which contains the password.
- Some Linux distributions have their glibc libraries compiled wrong, that will cause the **mbtask** program to do nothing useful. However these days this problem seems to have gone away.
- You cannot share the JAM message bases between little Endian and big Endian machines using networked file systems because the storage format doesn't specify the internal byte order for different CPU's. This is an error in the JAM spec.

Note:  This is a very old list and many if not all no longer apply – unless you know different...

# Linux Distributions.

**Which distribution**
GNU/Linux is available in several distributions, they all have advantages and disadvantages for bbs use. Which distribution to pick is very personal. Since late 2001 a port to FreeBSD is available, from version 0.33.19 this port is ready to use. Since January 2002 a port to NetBSD is available, later other *BSD versions were added.

**Slackware**
I am using MBSE BBS on several Slackware distributions. You can make a very small setup for MBSE BBS like Zipslack. Not included is the mgetty package.

**Redhat and Mageia**
I write this as if these are the same which isn't true of course. From MBSE BBS's point of view they are almost the same, so that's why I treat them as the same distributions. The E-Smith server and gateway based on RedHat is also supported. For people with little GNU/Linux experience these distributions are a good choice if you can spare the disk space. I haven't found a simple dedicated setup for the bbs, so the safest way is to install almost everything, which is quite simple. This will cost you about 1200 Megs. Maybe that someone more experienced with these distributions can give more details on how to build a small server.

**SuSE**
Since SuSE 7.1 the setup scripts are working and tested. Older distro's might work.

**Debian**
The installation works on a Debian 2.1, 2.2 and 3.0 distribution without any problems. How to build an optimized Debian system is not tested by me.

**Ubuntu**
Since 0.91.10 the installation on Ubuntu 7.04 works fine. Read the file README.Ubuntu in the source for more information.

**GenToo**
Installation and startup scripts are tested on GenToo. Read the file README.Gentoo for more information.

**FreeBSD**
I tested on FreeBSD 6.2 stable release. Older releases have been used and work. The setup is quite simple, do a small setup (average user), and add some needed packages from the ports collection such as gcc, mgetty, infozip etc. You may need to build mgetty yourself if you have an older FreeBSD distribution.

**NetBSD**
Currently NetBSD 3.1 is used for the latest tests. NetBSD before 3.0 doesn't work any more.

**OpenBSD**
Currently OpenBSD 3.7 is used for testing.

Famous last words...
I don't have the disk space and time for all kinds of GNU/Linux distributions to install for testing. Most distributions work without major changes. For questions about "unsupported" distributions, the best place to be is in the MBSE support echo.

# Running a BBS under Linux.

**Introduction**

Everyone who has been running a (single line) BBS under DOS until now will need to understand that running a BBS under Unix or Linux or any other multitasking o/s (operating system) is completely different of what you are used to. Any references to Unix should be considered equally the same as for Linux unless otherwise stated. Under DOS things were quite simple, from AUTOEXEC.BAT you started a new .BAT file that would run forever and started all needed programs after each other. The programs that were started depended on the error level of the previous program. Only one program could run at the same time.

People who had previous run a BBS on another multitasking o/s, or were running a BBS on a small Lan with a file server and workstations for each line, are already more used to the idea of running more programs at the same time, and to "signal" what to do next with semaphore files.

The Unix approach is more or less the same, but there are more differences. The main difference is that there is no mailer connected with the modem waiting for a call, instead there is a getty process watching your modem(s). For internet calls, these are started via (x)inetd using standard *nix methods. Another big difference is that you don't see what's happening, there is no screen with the mailer or bbs picture on it. All programs run in the background. If you don't like that, stop now and go back to your old DOS bbs. It's just the way everything is done.

Programs that must start at specific times (events in DOS), are started from cron using crontab, this is the event scheduler for Linux & Unix. With this program, maintenance can be started, polls created etc. For starting programs when they are needed there is a task manager loaded at system boot up. This task manager "watches" the semaphore directory of the bbs and will start what is needed. It will also watch the mailer outbound to see if there are nodes to call.

**Waiting for a call.**

Under Linux this is done with the mgetty program, this is the process that is connected with each modem (or ISDN adapter) and waits for a call. The mgetty program (written by Gert Doering, gert@greenie.muc.de) will detect the call, and find out what or who did make the call. It can detect incoming humans who want a login prompt, PPP calls from users who want to make a PPP connection (browsing your BBS with Firefox or Netscape for example), A fax machine trying to deliver a fax and finally a mailer trying to establish an EMSI, FSC-0006 or FSC-0001 session. The mgetty program is responsible for starting the right client programs. How to do this is explained in the installation manuals, but be sure to compile it with Fido and PPP support.

For internet calls, via (x)inetd mbsebbs can accept calls from connections on ports. For example if a connection comes in on port 24554 the mailer is started in binkp mode.

**A Human is calling.**

This could be a bbs user. For each user to login to your bbs there is a unix account. They automatic create such an account the first time they login with the **bbs** account. During the creation of their account the shell that is installed for their account is the mbsebbs binary, so that's the only thing that they get if they call in. When they logout the bbs, or drop carrier etc, the session is ended and mgetty takes over the line again. Note that they never can get a Unix shell unless you install a **door** in the bbs that calls a shell for them.

Because for dialin with mgetty the mblogin program is used and not the normal login program, only bbs users can login. All other users including **root** are denied access. You might consider installing SSH on your system for remote maintenance.

It is also possible to install the mblogin program as the login program for Telnet users if you are directly connected to the internet.

**A PPP call is detected.**

Installing a PPP server on your system is beyond the scope of this project. However if you did install it, users can login to your bbs with their favourite browser and use your bbs. Note that the necessary tools to automatic create newsgroups don't exist at this time. With the proper setup you can automatic create and maintain html pages for the file areas.

**A mailer call is detected.**

If a mailer is detected by mgetty, the **mbcico** program is started and will take over from mgetty. It will establish a mail session with the caller and the mail and or files will be exchanged just like any DOS mailer would do. After the call, mbcico will hangup and mgetty will take control of your modem again. If there is any mail received, mbcico will place the semaphore **mailin** so that another process can take care of the received mail.

**There is mail in the inbound.**

As said before, if the **mailin** semaphore is present, the task manager will then start the **mbfido** program that will toss the mail, process any files received and if necessary it will create other semaphore's for example to link the message bases, start the nodelist compiler etc. Note that this can be done while there may be a new mail session going on, a bbs user is online, it doesn't matter. Processing mail and files can be done using real multitasking without any damage to other processes.

**It's time to poll a node.**

At the time that you wish to poll a node, let cron create "poll" requests. When a poll is created, the semaphore **scanout** is also created. The task manager will then start mbcico at regular intervals so that mail will get out. If there is no more mail to send, the **scanout** semaphore is removed. If a time slot ends, you can just remove the "poll" requests that didn't succeed.

**It's Zone Mail Hour, so now what.**

Relax, if you have netmail ready for nodes the mailer script will try to send these mails to those nodes. If it was crash mail, and the destination was a non CM node, the mailer will try to send those mails too. Note that other crashmails are send any time. Also note that packed mail and files are not send during ZMH. If a node calls you during ZMH he will get everything that's waiting, including packed mail and files. The task manager (more on that later) calculates the Zone Mail Hour from UTC time, you don't have to change anything for summer- and wintertime. The task manager knows about all zone mail hours of all Fidonet zones (1..4). For othernets, zone mail hour is taken from the setup as default value. The task manager will also handle the mail for nodes with the Txx flag (FSC-0062) on the right times.

**Daily maintenance.**

This is started by cron jobs. There is no need to take your bbs lines down during maintenance, you can do it any time of the day. There are several scripts for this, daily, weekly and monthly.

**How about system load.**

Because GNU/Linux is a 32 or 64 bit OS, and not bothered with a graphical user interface (unless you install it), it has all the time in the world to serve your bbs programs. Background programs are build to release time to the Unix os, they don't need to run fast because it's background processing. The bbs and the mailer, have a low server load although there is no time release build in. Only the bbs has some short moments when it needs a lot of your system, for example when a user logs in and scans for new mail. The bbs I run is a 486-DX4 100 MHz, 20 MB ram, with 2 analogue lines, this seems to work fine. When this system's Mobo died, I used a 386DX33 for several months with 20 MB ram, and the only thing users ever noticed was that scanning for new mail was slower. I think this is the slowest hardware that will work. However, you must always use 16550A Uarts for the COM ports. For best performance use SCSI or Sata II or higher disks. I noticed that old 5"FH SCSI disks perform better for bbs usage then modern EIDE disks. This is probably caused by the fact that the kernel needs more time for the cheap IDE bus. If you want to use X11 on your bbs, you need more ram and a faster CPU or a separate machine via a Lan and export the display to that machine. Currently the smallest amount of memory you need is 36 MB, and that amount is only needed if you use a virus scanner which needs more memory then mbse.

# Basic Installation

**Introduction.**

Before you compile and install MBSE BBS you must first setup the basic environment. If you don't do this, things will fail.

To compile and install MBSE BBS most distributions have installed all needed packages. If important packages are missing then the configure script will tell you. There are also less important packages which if missing still let you compile MBSE BBS, but you will miss some features. Here is a short list of these packages:

1. **Zlib**. On some distributions you also need **zlib development**. When you have zlib installed, then in **mbcico** extra code will be compiled in the Hydra and Binkp protocol drivers that will allow the PLZ extension. In the binkp protocol GZ compression will be enabled as well.
2. **Bzlib**. On some distributions you need to install **bzlib2 development** to add support for binkp BZ2 compression.
3. **GeoIP**. Allows logging of the origin of the network connections by country and continent. You need at least GeoIP 1.4.3 to compile without errors. See www.maxmind.com.

If you use Gentoo or Ubuntu, read the files README.Gentoo or README.Ubuntu. The last one may be useful for Debian users too.


**Step 1: planning the file systems.**

MBSE BBS is default installed in **/opt/mbse**. It is possible to install the bbs in a different location using the options in configure (see the option "--prefix=PREFIX", e.g., "--prefix=/home" . See file system Hierarchy Standard for more info. The default file system layout looks like this:

```
/opt/mbse                       0775  Default MBSE_ROOT
/opt/mbse/bin                   0770  Binaries
/opt/mbse/etc                   0770  System configuration files
/opt/mbse/etc/dosemu            0750  DOSemu configuration files
/opt/mbse/ftp/pub               0755  Default FTP root for download areas.
/opt/mbse/home                  0770  Users homedirectories
/opt/mbse/home/bbs              0770  Newuser account
/opt/mbse/html                  0755  HTML documentation
/opt/mbse/log                   0770  MBSE BBS logfiles
/opt/mbse/share/doc             0750  Generated sitedocs
/opt/mbse/share/doc/html        0750  Generated html sitedocs
/opt/mbse/share/doc/tags        0750  Generated area tags
/opt/mbse/share/int             0750  Base for language files
/opt/mbse/share/int/macro/de    0750  German template macro's
/opt/mbse/share/int/macro/en    0750  English template macro's
/opt/mbse/share/int/macro/es    0750  Spanish template macro's
/opt/mbse/share/int/macro/fr    0750  French template macro's
/opt/mbse/share/int/macro/gl    0750  Galego template macro's
/opt/mbse/share/int/macro/nl    0750  Dutch template macro's
/opt/mbse/share/int/macro/zh    0750  Chinese template macro's
/opt/mbse/share/int/menus/de    0750  German menu files
/opt/mbse/share/int/menus/en    0750  English menu files
/opt/mbse/share/int/menus/es    0750  Spanish menu files
/opt/mbse/share/int/menus/fr    0750  French menu files
/opt/mbse/share/int/menus/gl    0750  Galego menu files
/opt/mbse/share/int/menus/nl    0750  Dutch menu files
/opt/mbse/share/int/menus/zh    0750  Chinese menu files
/opt/mbse/share/int/txtfiles/de 0750  German ansi screens
/opt/mbse/share/int/txtfiles/en 0750  English ansi screens
/opt/mbse/share/int/txtfiles/es 0750  Spanish ansi screens
/opt/mbse/share/int/txtfiles/fr 0750  French ansi screens
/opt/mbse/share/int/txtfiles/gl 0750  Galego ansi screens
/opt/mbse/share/int/txtfiles/nl 0750  Dutch ansi screens
/opt/mbse/share/int/txtfiles/zh 0750  Chinese ansi screens
/opt/mbse/tmp                   0770  Temp directory
** /opt/mbse/tmp/arc              0770  Temp archiver directory
/opt/mbse/var                   0770  Var root
/opt/mbse/var/arealists         0750  Areamgr arealist files
/opt/mbse/var/badtic            0750  Bad TIC files
/opt/mbse/var/boxes             0770  Base for nodes fileboxes
```

```
/opt/mbse/var/bso              0770  Binkley Style Outbound directory
/opt/mbse/var/bso/outbound     0770  Default outbound for main aka
/opt/mbse/var/dosemu           0770  Base for DOS drives (doors)
/opt/mbse/var/dosemu/c         0770  DOS drive C:
/opt/mbse/var/fdb              0770  Files database
/opt/mbse/var/hatch............0750..File hatch work area
/opt/mbse/var/inbound          0750  Protected inbound directory
/opt/mbse/var/magic            0750  Magic filerequest names
/opt/mbse/var/mail             0770  JAM messagebase root
/opt/mbse/var/msgs             0770  *.msgs netmail directory
/opt/mbse/var/nodelist         0750  Nodelists
/opt/mbse/var/queue            0750  Queue for before outbound
/opt/mbse/var/rules            0770  Echomail area rules files
/opt/mbse/var/run              0770  Pid files of running programs
/opt/mbse/var/sema             0777  semaphore files
/opt/mbse/var/ticqueue         0750  Queue for TIC files
/opt/mbse/var/unknown          0750  Unprotected inbound directory
```

Don't use UMSDOS or SAMBA file systems for the bbs, stick by the standard GNU/Linux file systems (ext3, ext4 or reiserfs) or ufs if you use FreeBSD. If you intent to make your bbs also accessible by FTP and WWW you must create the directory structure under the ftp user behind the pub directory. Read the ftp server doc for details. If you don't follow these guide lines, you will run into trouble later and have to spend a lot of time in correcting this error.

**Step 2: Running the installation script.**
First, if you use FreeBSD, install the psmisc package. This will make sure that MBSE BBS is stopped properly if you shut down your computer. Also if you use FreeBSD 5.0 or later make sure that you mount procfs, see **man procfs** for details. If you don't mount procfs, you cannot add new users to the bbs and existing bbs users cannot change their passwords. As sysop, you can't change passwords too.

If you are upgrading, proceed with step 4. If not, follow the next steps very carefully!

The installation script must be run by root. It checks if there is a previous or failed installation on your system. If that's so the script will not run. In other words, you can only run this script once. The script makes backup copies of the system files it changes, these files will get the extension **.mbse** To run the installation script you need the archive **mbbsebbs-1.0.6.tar.bz2** – there will no doubt be a later version so substitute '-1.0.6' with the current version. Unpack the archive on to your system, in /tmp:

```
cd /tmp
tar xfvj /path/to/the/mbsebbs-1.0.6.tar.bz2
To start the script type:
cd mbsebbs-1.0.6
bash ./SETUP.sh
Ubuntu users should do:
cd mbsebbs-1.0.6
sudo bash ./SETUP.sh
```

Yes, use **bash** as shell here. On some systems root doesn't use bash as login shell, calling the script with bash forces the use of bash. The script does the following:

1. Create the group **bbs**
2. Create the user **mbse** On Ubuntu this user is added to the admin group.
3. Create a **.profile** for user **mbse**
4. Create and set owner of directory tree under /opt/mbse

Then the script will ask you to give a password for user **mbse** This password is for system maintenance and for you to make changes to the bbs. You will need that frequently but you should not make that password easy to guess of course. The script will then continue again:

5. The user **bbs** is added.
6. The password will be removed from user **bbs** This action will make changes in /etc/shadow (if you have that) otherwise in /etc/passwd. On FreeBSD it uses other tools to modify the master database. On NetBSD you have to do that manually, there are no tools to do that.
7. If they don't exist in the file /etc/services the services fido, tfido and binkp will be added.
8. If they don't exist in the file /etc/inetd.conf the internet protocols for the mailer will be added. The **inetd** is restarted to activate the changes. If your distribution uses xinetd instead of inetd, an include file for xinetd is added instead.

**Step 3: Check the basic installation**

The last screen of the script is about sanity checks. Perform those checks! If something is wrong, now is the time to fix it. Don't panic and remember the backups of the system files that are changed are in /etc with the extension **.mbse** i.e: those were the original files. The installation logfile is copied to /opt/mbse. If everything is all right, then remove the directory /tmp/mbsebbs-1.0.6:

```
cd /tmp
rm -Rf mbsebbs-1.0.6
```

**Step 4: Install the basic packages.**

Login as user **mbse**. Yes, very important, **login as user mbse**. While in mbse's home directory (/opt/mbse) unpack the distribution archives:

tar xfvj /path/to/mbsebbs-1.0.6.tar.bz2

*Note that the ./configure options shown below may not work for you "as is" in which case try just using "./configure".*
You now have the subdirectory with sources in the right place. Next build the binaries and install them using the following commands:

```
cd ~/mbsebbs-1.0.6
./configure [--enable-optimize] [--enable-newsgate] [--enable-gdkdel]
make
su          important, do not use "su -"
password: enter root password here
make install
exit
```

Ubuntu users should do:

```
cd ~/mbsebbs-1.0.6
./configure [--enable-optimize] [--enable-newsgate] [--enable-gdkdel]
make
sudo make install
```

**Important:** it seems logical to use the `--enable-newsgate` option but it isn't. When you do, the mbnntp program is disabled and you cannot serve echomail as news to your users via internet. But you can gate echomail to the internet. Independent of your choice, you can always make internet news available for your bbs users. Only use --enable-newsgate if you really need to gate echomail to the internet.

To support Chinese the DELete key needs to be changed because it conflicts with one of the characters. If you want to use Chinese, use `--enable-gbkdel` to configure.

The last part of the installation procedure shows you the location of the bbs startup script that is added to your system. Remember this one for a moment. Because this is your first time installation, example menus and example textfiles are installed. If they already exist on your systems (when you do an upgrade) they will not be installed again.

Now start the bbs for the first time (still as root) by executing the startup script you just saw on the screen followed by a space and the word **start**. For example:

```
/etc/rc.d/init.d/mbsed start
```

Ubuntu users do:

```
sudo /etc/init.d/mbsebbs start
For Mageia do:
/etc/rc.d/init.d/mbseb start
or
sudo systemctl start mbsed.service
```

This will start the **mbtask** daemon. After that the bbs will be opened for use. Check the file **/opt/mbse/log/mbtask.log** for startup problems. The first time **mbtask** is started on your system it will create a lot of new configuration files with default settings.

**Step 5: Ready.**

Now you have shell scripts in ~/etc, most of them are called by cron, some are called during system startup and shutdown. You also have some default configuration files, these are ttyinfo, modems, fidonet networks. In the default (English) directory you now have default menu data files and ansi screens. These are copies of the test system so you have to edit them to build your own bbs.

Editing ansi screens can be done on a GNU/Linux system with one of the following packages:

    1. **duhdraw**, this is available from many bbs systems as **duhdraw.tgz** (68 Kbytes). The binaries are included

in this archive, if you compile it yourself it may give trouble so if the binaries work, use these.

2. **TetraDraw** This is a very nice Ansi editor, you can get this file as TETR~VC#.TGZ also from many bbs systems. The file is 157 Kbytes.

You may also want to edit ~/etc/header.txt and ~/etc/footer.txt, these files are the top and bottom of the newfiles/allfiles listings.

Now the basic environment is finished, the next thing to do is configure the bbs.

# Upgrading the BBS.

**Introduction.**
First of all, if you upgrade from a previous version of MBSE BBS, read all the ChangeLog entries from the version you are currently running up to the new version you are installing. Then make a backup of at least all files in /opt/mbse/bin and /opt/mbse/etc.

**Install the source.**
Login as user **mbse**. Yes, very important, **login as user mbse**. While in mbse's home directory (/opt/mbse) unpack the distribution archives:
```
tar xfvj /path/to/mbsebbs-1.0.6.tar.bz2 [– or later version]
```
You now have the subdirectory with sources in the right place. Next build the binaries and install them using the following commands:
```
cd ~/mbsebbs-1.0.6
make clean
./configure
make
```
**[Important, do not use "su -"]**

```
su password: enter root password here
make install
Ubuntu users should do:
cd ~/mbsebbs-1.0.6
make clean
./configure
make
sudo make install
```
The last part of the installation procedure shows you the location of the bbs start up script that is added to your system. Remember this one for a moment.
Now restart the bbs (still as root) by executing the start up script you just saw on the screen followed by a space and the word **reload** or **restart**, this depends on the distribution you use. There are also distributions that must use the startup script twice, once with **stop** and then **start** as parameter. For example:
```
/etc/rc.d/init.d/mbsed reload
exit
```
Ubuntu users do:
```
sudo /etc/init.d/mbsebbs restart
```
This will close the bbs, kill **mbtask**, start the new **mbtask** and open the bbs again for use. At this point you may need to update some configuration settings with **mbsetup**. Read the ChangeLog for the details. The ChangeLog may also describe a different restart procedure if this is needed.

**Other updates.**
Not everything is updated during the upgrade, only the binaries. Normally this is enough. However it may be that the example menus, macro templates are updated in the distribution. Because these files may have some very personal changes, these files are not overwritten. You can force that, in the subdirectory ~/mbsebbs-1.0.6/examples type
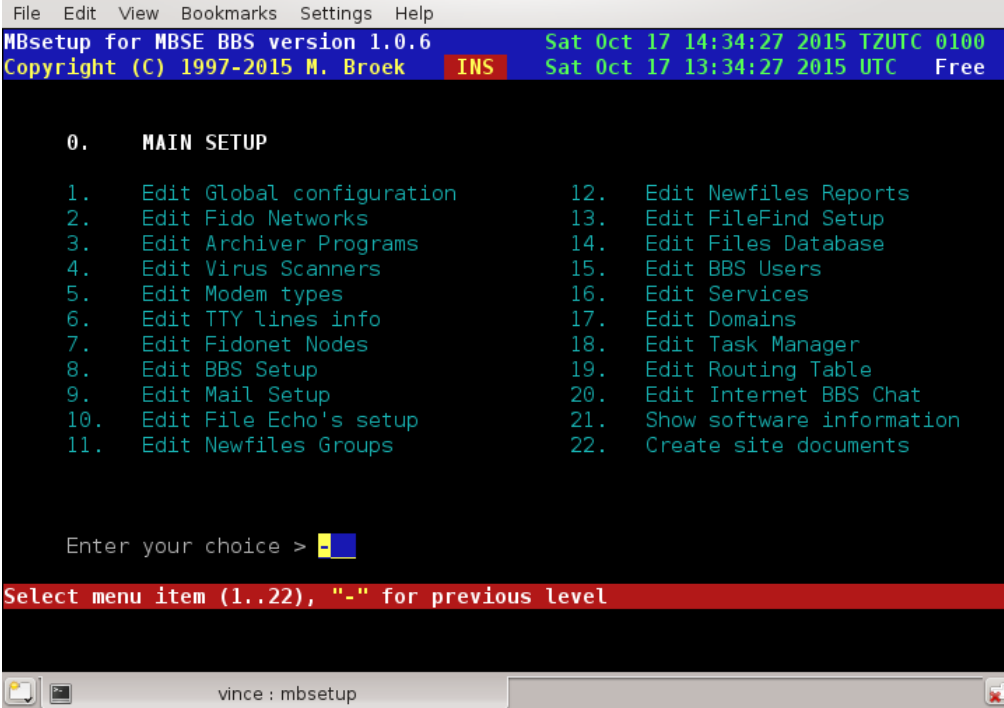
```
make help
```
to see what you can force to update. But carefully taking over the changes may be better.
In the directory /opt/mbse/bin and /opt/mbse/etc are new scripts installed. They have the extension **.new** and you need to examine these to see if you can simply copy these over your current ones.

# Setup Guide.

**Invoking mbsetup**

As user **mbse** type **mbsetup** to start the setup program. When you start **mbsetup** you will see the following screen:

# Mbsetup Main Options

1. [Edit Global configuration](#)
2. [Edit Fido networks](#)
3. [Edit Archiver programs](#)
4. [Edit Virus scanners](#)
5. [Edit Modem types](#)
6. [Edit TTY lines info](#)
7. [Edit Fidonet nodes](#)
8. [Edit BBS setup](#)
   1. [Edit Security limits](#)
   2. [Edit Language setup](#)
   3. [Edit BBS menus](#)
   4. [Edit File areas](#)
   5. [Edit Transfer protocols](#)
   6. [Edit Oneliners](#)
9. [Edit Mail setup](#)
   1. [Echo mail groups](#)
   2. [Echo mail areas](#)
10. [Edit File echo's setup](#)
    1. [Edit Fileecho groups](#)
    2. [Edit Fileecho areas](#)
    3. [Edit Hatch manager](#)
    4. [Edit Magic files](#)
11. [Edit Newfiles groups](#)
12. [Edit Newfiles reports](#)
13. [Edit Filefind setup](#)
14. [Edit Files database](#)
15. [Edit BBS users](#)
16. [Edit Services](#)
17. [Edit Domains](#)
18. [Edit Task Manager](#)
19. [Edit Routing Table](#)
20. [Edit Internet BBS Chat](#)
21. [Show software information](#)
22. [Create site documents](#)

# Setup - Global Setup.

In this setup you can edit all global settings for MBSE BBS. All sections will be discussed below.
Note that all path quoted below are off the default mbse home directory of /opt/mbse/

### 1.1. Fidonet Aka's.
Here you can enter 40 fidonet addresses. These are 5d addresses. Make sure you group the different zone's together, strange things will happen if you don't group them.

### 1.2. Edit Registration Info.
**BBS Name**     The name of your BBS
**Maildomain**   Your internet mail domain name (or system host.domain.com if you don't have a maildomain).
**Sysop uid**    The Unix name of your Sysop account
**Sysop Fido**   The Fidonet name of your Sysop account
**Location**     The Location of your BBS
**OLR id**       The packet name for Offline mail downloads.
**Comment**      A comment line for your BBS
**Origin**       The default origin line for echomail
**Newuser**      The default user name "bbs" for new users.
**My FQDN**      My real internet Full Qualified Domain Name.

A note about the last item, My FQDN. This **MUST** be the name that is returned with `nslookup 1.2.3.4` (use your internet IP address there) if you have a fixed IP address. Systems that are on a dynamic IP address and use some sort of dyndns service **may** put that dyndns FQDN in there. Using this will cause the other system to do twice as much dns resolve actions, so use this only when needed. Another note, what you put in here has nothing to do with mail or mail domains, that needs to be in the second option of this screen. The My FQDN field is used by the Internet BBS Chatserver (IBC) to allow bbs systems to build a chat network.

### 1.3. Edit Global file names.
**System logfile**   The name of the main logfile
**Error logfile**    The name of the errors logfile
**Debug logfile**    The name of the debug logfile
**Mgr logfile**      The name of the area-/filemgr logfile
**Default menu**     The name of the default main menu
**Chat logfile**     The name of the logfile for chatting
**Welcome logo**     The name of the BBS logo ANSI file

### 1.4. Edit Global Paths
**Home dirs**    The path to the users home directories
**FTP base**     The FTP base path, ie. /opt/mbse/ftp/pub
**Arealists**    The path where area lists and filebone lists are stored
**Ext. edit**    The full path and file name to the external editor
**Rules dir**    The path to the directory to store area rules
**Magic's**      Where the magic file requests are kept
**DOS path**     The DOS drive and path
**Unix path**    The Translated DOS path in real
**LeaveCase**    Leave outbound .flo file names as is, No forces to uppercase
**Nodelists**    The path to the nodelist directory
**Inbound**      The unprotected fidonet inbound
**Prot inb.**    The (password) protected fidonet inbound
**Outbound**     The outbound for the main aka
**Out queue**    The outbound queue for temporary files
**\*.msgs**      A netmail directory from where \*.msg mails are tossed
                 These are \*.msg files can be created by several doors. (May be not in use yet)

| **Bad TIC's** | Where bad TIC files are saved |
| **TIC queue** | Where TIC files for downlinks are kept |
| **TMail DOS** | The T-Mail 8.3 (short) base path (empty = disable) |
| **TMail Win** | The T-Mail long file name base path (empty = disable) |

If you fill in the DOS path then the DOS path and Unix path are translated to DOS paths in the .flo files for outbound mail sessions. You only need this if another mailer uses the outbound that doesn't understand Unix paths. The DOS path is a fake, but it must match the other mailer. Note that the TIC queue must be some were in the Unix path, otherwise it is impossible to create a DOS path from the path to the TIC files in that directory. To set this up correctly is also important if you use other mailers as well, for example binkd, or even a DOS binkley style mailer running from a network. All those mailer must "see" the same file attaches in the .flo files.

### 1.5. Edit BBS Configuration

| **Exclude Sysop** | True is Sysop will be invisible |
| **Show connect** | Show connection info at logon |
| **Ask protocols** | Ask protocol before each up/download |
| **Sysop level** | The Sysop security level |
| **Password Length** | The minimum password length, should be 6. |
| **Password Char.** | The password hiding character |
| **Idle timeout** | The idle timeout in minutes |
| **Login Enters** | Maximum times for only enter <Enter> |
| **Login Attempts** | Maximum login attempts |
| **Homedir Quota** | Maximum size in MBytes for each user |
| **Location length** | The minimum length of the location (3 in Holland!) |
| **Show new msgarea** | Show new available message areas at logon (for OLR users) |
| **OLR Max. msgs.** | Maximum messages to download, 0 is no limit. |
| **OLR Newfile days** | The maximum age for newfiles in the OLR packet |
| **OLR Max Filereq** | Maximum file requests allowed for OLR users |
| **BBS Log Level** | What will be logged or not the BBS program |
| **Utils Log Level** | What will be logged or not for utilities |
| **Utils slowly** | Should utilities release time slices |
| **CrashMail level** | Minimum level to allow sending netmail crash |
| **FileAttach level** | Minimum level to allow attach files to netmail |
| **Min diskspace MB** | At which low disk space level utilities should stop working. |
| **Simult. Logins** | Maximum simultaneous logins allowed, 0 is unlimited, 1 is advised. |
| **Child priority** | Sub-processes nice priority, 0=high, 15=low CPU load. |
| **file system sync** | Call sync before and after execute. |
| **Default language** | Default language (English). |

The minimum disk space setting is to prevent that files get corrupted if your file system is full. All drives are checked except CD-roms and floppies and the /boot directory if that one is on a separate file system. Ext3, ext4, reiserfs, msdos and vfat partitions are checked. The lowest free disk space found counts. Default is 10 MB.

The Child priority sets the nice value for example zip/unzip, virus scanners etc. Modern fast hardware will do fine with a low setting, older (pre PII) hardware may need 15 to prevent a too heavy CPU load.

The file system sync setting is to call sync before and after the execute call, such as when unzip is called. Almost all GNU/Linux systems need this because GNU/Linux uses asynchronous directory changes. Since Linux 2.5.19 it should be possible to mount file systems synchronous. If this setting is No on asynchronous file systems mail and files can get corrupted. XxxxBSD systems use synchronous directory updates and don't need this switch. If you don't know what all this is about, leave this to the default setting.

Choice of default language. English is the best choice because that is the only language that is complete in the default installation. In the user set up another language may be used, the language set here is then used as a backup.

### 1.6. User flag Descriptions.
In this menu you can give the 32 users flags a meaningful description.

## 1.7. New users defaults.

| | |
|---|---|
| **Access level** | The access level and flags after registration |
| **Cap. user name** | Capitalize the user name |
| **Ask Sex** | Ask for Male/Female |
| **Ask Voicephone** | Ask for voice phone number |
| **Ask Dataphone** | Ask for data phone number |
| **Telephone scan** | Scan for duplicate numbers |
| **Ask Handle** | Ask for handle (nickname) |
| **Ask Birth date** | Ask for birth date (needed for checks) |
| **Ask Location** | Ask for users location |
| **Ask Hot-Keys** | Ask for hot-keyed menus (default is yes) |
| **One word names** | Allow one word names |
| **Ask address** | Ask user for his/her home address |
| **Give email** | Give new users email access (default is yes) |
| **Ask Screenlen** | Ask new users to set their screen length |
| **Do newmail check** | Yes/No or Ask new user if for newmail check at logon |
| **Do newfiles check** | Yes/No or Ask new user if for newfiles check at logon |

## 1.8. Text Colors.

Several prompts use different colours. They can be changed with the following menu.
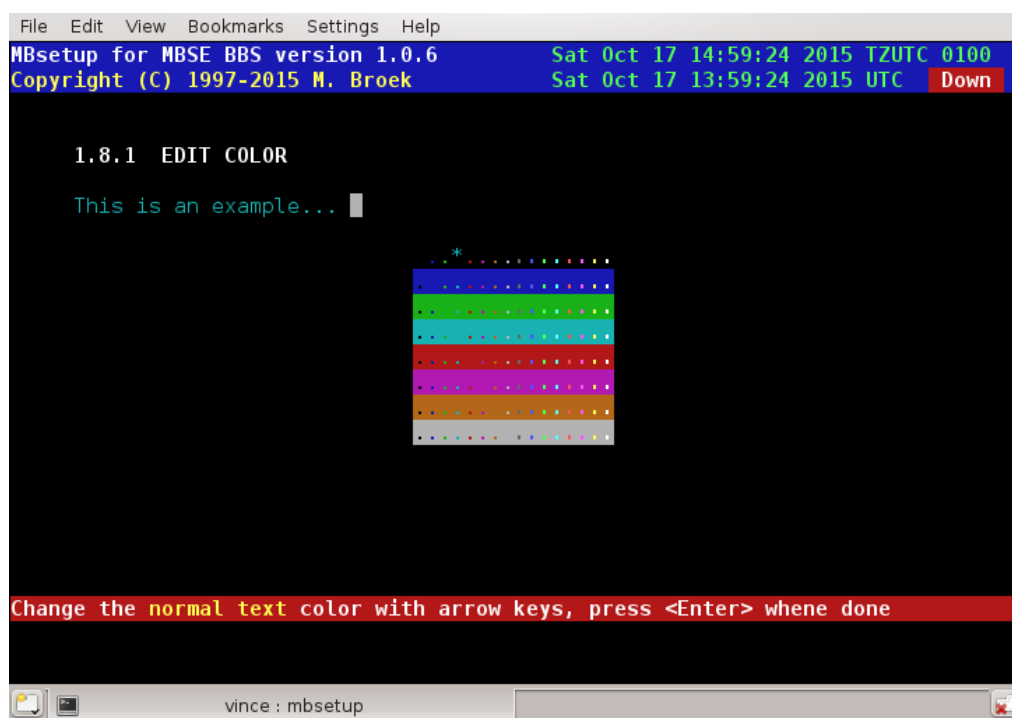
## 1.9. Sysop paging

For sysop chat a protocol is used to communicate with the mbtask daemon which has a very simple IRC like chatserver. For sysop/user chat a forced channel with the name #sysop is used.

**Page length**     The length of a page in seconds
**Page times**     Maximum number of times a user may page the sysop
**Sysop area**     Message from user to Sysop area number
**Ask reason**     Ask reason for chat, this will be logged
**Log Chat**     Log the chat conversation
**Prompt Chk.**     Check at menu prompts for Sysop breaking in
**Freeze Time**     Freeze users time during chat

## 1.10. Fileecho Processing.

When you change the number of Systems or Groups, the databases affected will be updated automatic.

**Keep days**     How long TIC files should be kept on hold
**Hatch pwd**     The internal hatch password. Make this weird.
**Drv space**     The minimal free space on your disk in kilobytes
**Systems**     The maximum number of connected nodes
**Groups**     The maximum number of fileecho groups
**Max. dupes**     The maximum number of entries in dupe database
**Keep date**     Keep original filedate
**Keep netm**     Keep sent netmails
**Loc resp**     Respond to local created filefind messages
**Plus all**     Allow filemgr +all command
**Notify**     Allow filemgr notify=on/off command
**Passwd**     Allow filemgr/areamgr passwd command
**Message**     Allow filemgr message=on/off command
**Tic on/off**     Allow filemgr tic=on/off command
**Pause**     Allow filemgr pause/resume commands

When you change one of the **Allow filemgr** settings, you also need to edit the file ~/share/int/macro/en/filemgr.help to reflect the new settings. Default all these switches are set to Yes.

## 1.11. Edit Fidonet mail and echomail processing.

Note that the first 2 mail boards must also exist in the normal mail areas if you want to see what is in them. Here

they are defined for quick access of the tosser. For the Max. systems and groups see 1.12. If you use MBSE BBS together with a DOS based BBS (using DOSEMU or mars netware emulator), you can set the behaviour of the outbound to 4d. addressing instead of 5d. This option may disappear in the future.

Another option is present, this is the **pktdate** option. This is the full path and file name to an external program that can inspect and correct the mail .pkt files. Originally I put this in to run pktname of Tobias Ernst of 2:2476/418 to fix y2k problems in the incoming mail. At this time most y2k fixes are build in, but in case you need it it's there. To make it clear; the y2ktools written by Tobias are static compiled for GNU/Linux and they should run on all GNU/Linux i386 versions. This should no longer be required – Vince.

| | |
|---|---|
| **Badboard** | The path and file name of the bad messages |
| **Dupeboard** | The path and file name of the duplicate messages |
| **Pktdate** | Full path and file name of a .pkt preprocessor |
| **Max pkts.** | Maximum Kb. of mail packets before a new one is created. |
| **Max arcs.** | Maximum size in Kilobytes of an arcmail file |
| **Keep days** | How many day should we keep ARCmail on hold |
| **Echo dupes** | Maximum number of entries in the echomail dupe database |
| **Reject old** | Reject echomail messages older then n days |
| **Max msgs** | Default maximum number of messages in each area |
| **Days old** | Default number of days old to keep messages |
| **Max systems** | Maximum number of nodes to connect to echomail |
| **Max groups** | Maximum number of echomail groups |
| **4d address** | Use 4d. addressing (not needed you only use MBSE BBS) |
| **Split at** | Gently split messages after n KBytes (12..60) |
| **Force at** | Force split of messages after n KBytes (16..63) |
| **Plus all** | Allow areamgr +all command |
| **Notify** | Allow areamgr notify=on/off command |
| **Passwd** | Allow areamgr/filemgr passwd command |
| **Pause** | Allow areamgr pause/resume commands |

When you change one of the **Allow areamgr** settings, you also need to edit the file ~/share/int/macro/en/areamgr.help to reflect the new settings. The default of all these switches is set to Yes.

A note about the splitting of messages. Some tossers can't handle messages greater than 16 KBytes, these tossers are rare these days. Most tossers can handle messages of 32 KBytes. To set these values on the safe side set "Split at" to 27 and "Force at" to 31. This means that a long newfile report will be split after 27 KBytes when a new group of files should start in the report. If it can't find that point because a large number of files is in the group that is just being processed, the message split will be forced right after the file that passes the 31 KBytes limit. I use values of 1 KBytes below maximum for overhead such as SEEN-BY and PATH lines. Values larger then 32 KBytes is not a good idea, recent tests in May 1999 have shown that your messages will not reach all systems if they are larger then 32 KBytes. Splitting is used for newfiles reports and gated news articles to Fidonet.

**1.12. Edit Internet mail and news processing.**

Email and news is set up here. There are three possible configurations which you can set with 1.12.11:

- **No internet**. If you don't have any connection to the internet use this setting. Email will come from the default Fidonet UUCP gate and will be send out via the UUCP gate. Users have email addresses like **user@f2802.n280.z2.fidonet.org** Note, the user name is their Unix name when sending email.

- **No maildomain**. If have internet but don't have your own mail domain (most dial up systems) you should use this mode. Email will be sent via your local SMTP port, then through your own sendmail (or whatever you use) to your ISP. As soon as you are connected to the internet the mail will be sent to your ISP's mailer. In your sendmail you should define the mailer of your ISP as Smarthost. Incoming email will still come from the UUCP gate. Users have email addresses like **user@f2802.n280.z2.fidonet.org** Incoming email comes from the default Fidonet UUCP gate. Note, the user name is their Unix name. If you have your own mail domain, don't use this option, use the next option:

- **Own maildomain**. If you are permanent connected to the internet either with a static or dynamic IP address use this option. Use this option also if you have an UUCP domain and have a dial up UUCP connection. Also you can use this for dial up if you have your own mail domain without UUCP, you will then

fetch the mail from a multi drop mailbox (POP or IMAP) and send the mail using SMTP via the SMTP mailer of your provider.

If you have a dynamic IP address you can still get a DNS name from for example [dynip.com](dynip.com) Incoming email will come directly from the internet, but if someone sends email via the UUCP gate it is also accepted. Users have email addresses like **user@yourbbs.domain.org**. Note, the user name is their Unix name and yourbbs.domain.org must be an existing internet domain.

Configuration of the internet, ppp, sendmail etc. is not discussed here, see the HOWTO's and other documentation that exists at [www.linuxdoc.org](www.linuxdoc.org), it's all there.

| | |
|---|---|
| **POP3 node** | The POP3 node to use, should be localhost |
| **POP3 user@domain** | Use user@domain to login in the POP3 server |
| **SMTP node** | The SMTP node to use, should be localhost |
| **NNTP node** | The NNTP node to use, should be localhost |
| **NNTP m.r.** | If the NNTP server needs the Mode Reader command |
| **NNTP user** | The user name for the NNTP server if needed |
| **NNTP pass** | The password for the NNTP server if needed |
| **News dupes** | The number of entries for the news dupes database. |
| **Email aka** | The Fidonet aka to use for the fidonet.org UUCP gate |
| **UUCP aka** | The default Fidonet UUCP gate, e.g., 2:292/875@fidonet |
| **Emailmode** | The email mode, discussed above |
| **News mode** | News feed mode, INN, rnews or UUCP. |
| **Articles** | The default maximum news articles to fetch per group. |
| **Split at** | Gently split newfiles reports after n KBytes (12..60) |
| **Force at** | Force split of newfiles reports after n KBytes (16..63) |
| **Control ok** | Allow news control messages to be gated |
| **No regate** | Don't regate already gated messages |

In rnews mode the NNTP entries are replaced by:

| | |
|---|---|
| **Path rnews** | The full path and file name to the rnews binary. |

In UUCP mode the NNTP entries are replaced by:

| | |
|---|---|
| **UUCP path** | The full path to the uucp public directory. |
| **UUCP node** | The UUCP node name of your ISP. |

### 1.13. Allfiles and Newfiles lists.
These are the settings that affect the generation of newfiles and allfiles reports.

| | |
|---|---|
| **New days** | The number of days old files are "newfiles" |
| **Security** | The highest security level to include files in the reports |
| **Groups** | The number of newfile groups the newfiles database can hold |

### 1.14. Mailer Setup.
Note that you can't disable FTS-0001 sessions as that is a mandatory session protocol in Fidonet. There are 40 phone number translations present, this is for countries with lots of telephone operators with all kind of prefixes for carrier select functions. The setting of TCP/IP flags is important too, this gives the information what this system will do. If you only enter IBN as TCP/IP flags then the system will only call IBN nodes (binkp). The default is XX,CM and TCP/IP systems (internet) should use the XX,CM,IBN,IFC flags. If you cannot do TCP/IP connections, leave TCP/IP "phone" empty. The three TCP/IP fields are used to give your system information via EMSI or binkp to the other nodes.

| | |
|---|---|
| **Mailer logl.** | The logging level for mailer sessions |
| **TCP/IP "phone"** | The TCP/IP "phone" number (IP address or domain name) |
| **TCP/IP flags** | The TCP/IP capability flags for TCP/IP sessions |
| **TCP/IP speed** | The TCP/IP line speed |
| **Timeout reset** | The time out for normal modem commands |

| | |
|---|---|
| **Timeout connect** | The time out for waiting for connect |
| **Dial delay** | The maximum delay between calls, minimum is 10 seconds. |
| **No Filerquests** | Disable file requests |
| **No callout** | Disable call out |
| **No EMSI session** | Disable EMSI |
| **No YooHoo/2U2** | Disable FTS-0006 sessions |
| **No Zmodem** | Disable zmodem protocol |
| **No Zedzap** | Disable zedzap protocol |
| **No Hydra** | Disable Hydra protocol |
| **No MD5** | Disable binkp MD5 passwords |
| **Zero Locks OK** | Allow zero byte lock files from another OS |
| **Phonetrans 1..40** | Maximum 40 phone number translations |
| **Max. files** | Maximum files to request, 0 is unlimited |
| **Max. Mbytes** | Maximum MBytes to request, 0 is unlimited |

## 1.15. Edit HTML pages setup.

Here you set up the HTML pages that can be created with the **mbfile web** command. These are HTML pages of your download areas and indexes of all areas. If there are pictures in these areas thumbnails are created if you have the **convert** command available. The document root is the same as defined in your web server. The link to ftp must be created from that directory to your ftp base directory. To do that become root, cd to the document root and type **ln -s /var/spool/mbse/ftp files** In this case the link is called **files**. Note that all download areas are accessible, there is no user authentication yet available.

| | |
|---|---|
| **Docs root** | The path to the httpd documents root. |
| **Link to ftp** | The link to the ftp directory. |
| **URL name** | The URL of your web server. |
| **Charset** | The default character set, ISO-8859-1. |
| **Author name** | The author name you want in the HTTP headers. |
| **Convert command** | The graphics convert command. (ImageMagick needed). |
| **Files/page** | The number of files to display per web page. |
| **Hist. Limit** | Limit the number of mail history.html lines, 0 is unlimited. |

## 1.16. Manager flag Descriptions.

In this menu you can give the 32 area-/filemgr flags a meaningful description.

# Setup - Fidonet Networks.

Each fidonet network can have maximum 6 zones. The main zone (where you are) must be the first zone, the others will follow. You can add 6 additional nodelists to merge with the main nodelist. These additional nodelists are normally more recent that the main nodelist, so entries in the additional nodelists will replace entries from the main nodelist when you compile the nodelists. In the shown example you can see that I have a regional nodelist and a pointlist added for my region. For each additional list you must specify the RC address because that information is normally not present in these nodelists. **Watch out!** Nodelist names are case sensitive. If you receive a nodelist and automatic put them in place with the **mbfido** program, and the resulting file is upper case, you must use upper case names here also. You don't need to give the extension of the nodelist name, the **mbindex** will figure that out.

# Setup - Archiver programs.

To process mail, files and test new uploads you need archivers to process those files. For each (un)archiver you must set up the full path and file name and command line switches. Archivers and un-archiver's may be different programs such as **zip** and **unzip**.

There is a little difference in processing mail and files, mail will always work on the same directory, while for ticfile processing the archives can contain subdirectories. So it is obvious that for re-archiving a file you need the recursive switches to keep the directory structure within an archive as it was.
There is also a special command to replace a banner in an archive. This is when you receive files with the banner of your uplink in it and you want to replace it with the add of your own bbs and you don't want to mess with the files in the archive.

The last option is to extract the file FILE_ID.DIZ from the archive, this can be used for file description when the file is imported in your bbs. To make life a little more easy, during the first bbs set up the most common archivers already configured. You only need to make sure that they are really present on your system.
When you configured the sources and build mbse, the configure script searched for existing archivers. When mbsetup was run the first time, when mbtask was started, the archivers found on your system are already configured with the right paths and enabled. If you want the refresh the archivers configuration again after you added new archiver programs, then remove ~/etc/archiver.data. Compile the whole bbs again, start with `make clean` and `./configure`. After you reinstalled the binaries, run **mbsetup** and the new archivers are automatically added to the setup. Of course, you can also add new archivers manually. Archivers that were not originally found but are known, are present in the set up but are disabled.

# Setup - virus scanners.

Once upon a time there was no DOS and no computer viruses. But since DOS was invented as a small OS which was easily extensible, virus writers saw their chance to easy spread their hacks. Although running a GNU/Linux system is relative safe, most of the files that you have available on your bbs are DOS/Windows based programs. And before you put them available for download, they should be checked for viruses. Macro viruses are a relative new danger, this can also hurt Unix/Linux users.

There are several scanners for GNU/Linux available. Default only four of them are setup. You may consult http://www.openantivirus.org for more scanners mentioned in a mini-FAQ maintained by Rainer Link.

When you configured the sources and build mbse, the configure script searched for existing scanners. When mbsetup was run the first time, when mbtask was started, the scanners found on your system are already configured with the right paths and enabled.

The following scanners are default installed in the setup:

- **NAI Virus Scan** (uvscan) for Unix (GNU/Linux) made by Network Associates, USA. Not free for personal use. Uses the same DAT files as for Windows and DOS.
- **AntiVir/Linux** made by H+BEDV Datentechnik GmbH. Can also be installed in sendmail or Postfix to scan incoming and outgoing email. This may be a good idea if you run a email gateway. This version can be registered for personal use.
- **Clam AntiVirus** is a GNU licensed virus scanner for Unix. It is available from www.clamav.net. It has one slight disadvantage over other scanners (or just the opposite), when it tests a file with the Eicar testvirus signature it will report that and triggers the virus detection. This happens with NAI DAT files.

As soon as you have made one scanner available in the setup and you receive files in tic areas where the scan flag is set, then these files will be checked. As soon as one of the scanners detects a virus the received file will not be imported. Uploads from users will be checked with the installed virus scanners as well.

**Stream scanners**

One feature is stream scanning. In this setup you need a virus scanner loaded as a daemon and it must listen to a TCP/IP port to receive commands and data to scan. Currently this is only implemented for ClamAV. First you need a machine where **clamd** is running, this can be a remote machine but of course also the bbs machine itself. ClamAV needs to be configured so that it listens to a TCP/IP port, and depending on other things on the local socket too. Recent versions of ClamAV can do both together. Change your /etc/clamav/clamd.conf to contain the following lines:

```
# Path to a local socket file the daemon will listen on.
# Default: disabled (must be specified by a user)
LocalSocket /var/run/clamav/clamd

# Remove stale socket after unclean shutdown.
# Default: no
#FixStaleSocket yes

# TCP port address.
# Default: no
TCPSocket 3310

# TCP address.
# By default we bind to INADDR_ANY, probably not wise.
# Enable the following to provide some degree of protection
# from the outside world.
# Default: no
#TCPAddr 127.0.0.1
```

I left the comment for the TCPaddr, but it's up to you to protect the clamd server. After you restart **clamd** test the connection with telnet host.where.clamd.runs 3310, type VERSION followed by a return and you should see the ClamAV version. If that works, you can enable the ClamAV stream scanner in mbsetup and disable the old command line scanner.

So why would you use this. It's about 10 times faster then the command line scanner.

# Setup - Modem types.

In the setup screen you can define all kinds of modems you use. This includes ISDN modems. This is not the setup of individual lines, that is in the next section, so if you own a bbs with 5 analogue lines with only two brands and types of modems connected, you need only to define those two types of modems here. Some defaults are installed during initial bbs set up.

**Setup a modem.**

| | |
|---|---|
| **Type** | The description of this modem. |
| **Init 1** | The first modem init string. |
| **Init 2** | The second init string (if needed). |
| **Init 3** | The third init string (if needed). |
| **Reset** | Not in use |
| **Hangup** | Only needed if drop DTR doesn't work. |
| **Dial** | The dial command. |
| **Info** | Command to get caller-id (not tested). |
| **Ok** | The modem "OK" response. |
| **Offset** | The answer/connect time offset. |
| **Speed** | The maximum modem line speed, ie 28800. |
| **Available** | If this modem is available. |
| **Deleted** | If this modem must be deleted. |
| **Stripdash** | Strip dashes from the dial command. |
| **Connect strings** | Here you can define 20 connect strings. |
| **Error strings** | Here you can define 10 non-connect strings. |

Special characters

| | |
|---|---|
| **\\** | Send one backslash. |
| **\r** | Send the CR character. |
| **\n** | Send the LF character. |
| **\t** | Send the TAB character. |
| **\b** | Send the BS character. |
| **\s** | Send a space character. |
| **\d** | Wait one second. |
| **\p** | Wait 0,25 second. |
| **\D** | Send untranslated phone number. |
| **\T** | Send translated phone number. |

**The Hangup field.**
This is only needed if your modem doesn't hangup by dropping the DTR line for one second. Most modems do that if &D2 or &D3 is in the init string.

**The Offset field.**
The **Offset** field is to calculate the cost for outgoing calls. Analogue modems need time to establish the connection, 6 seconds is quite common. So when you see the CONNECT Blabla message, the phone connection is there already 6 seconds and you are already paying for 6 seconds. This offset is thus added to the total calculated connect time for cost calculations. For ISDN modems this can be 1 or 0.

MBsetup for MBSE BBS version 1.0.6        Sat Oct 17 15:10:05 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS   Sat Oct 17 14:10:05 2015 UTC   Free


```
5.  EDIT MODEM TYPE

1.  Type     USR Courier v.all
2.  Init 1   ATZ\r
3.  Init 2
4.  Init 3
5.  Reset    ATZ\r
6.  Hangup
7.  Dial     ATDT\T\r
8.  Info     ati6\r
9.  Ok       OK          12. Available No        15. Connect strings
10. Offset   6           13. Deleted   No        16. Error strings
11. Speed    33600       14. Stripdash No



       Enter your choice > -
```

Select menu item (1..16), "-" for previous level


vince : mbsetup

# Setup - TTY Lines.

**Introduction.**

For each modem or ISDN line your bbs has you must set up a tty line. This also includes console lines (for local login). Since version 0.81.0 you don't need to setup network lines any more, these are now built in. If a call comes in over a (modem or ISDN) tty you didn't define, that call is refused.
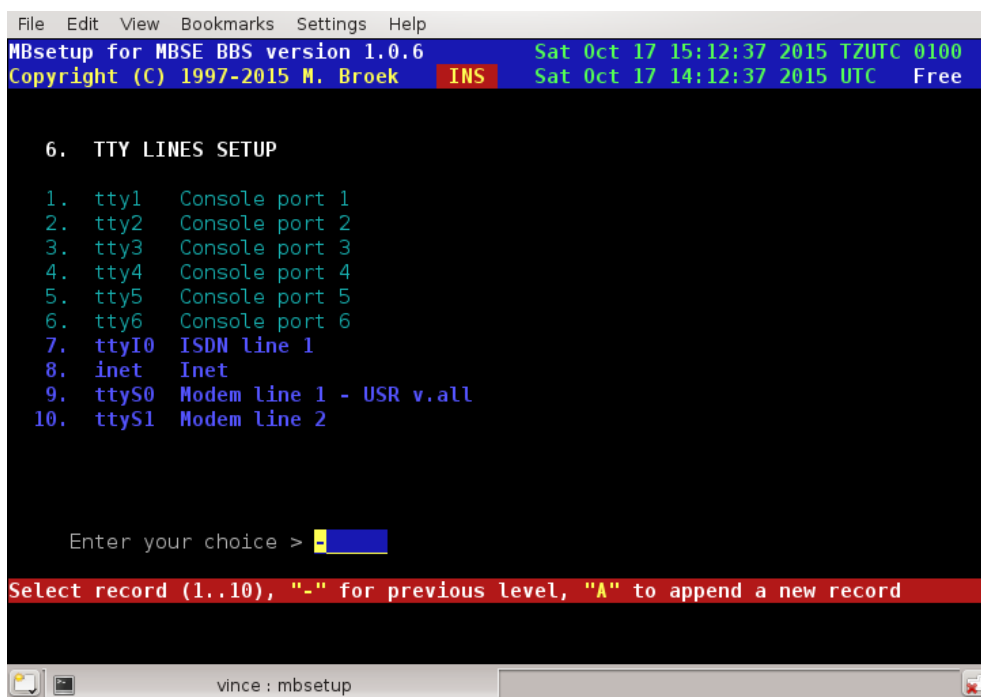
One thing about the port speed, this is only needed for devices connected to serial ports such as modems and external ISDN adapters. For internal ISDN cards this should be set to zero. If you set it to some other value, things still work but you will get error messages in the logs.

A note about the EMSI flags, this <u>must</u> match your modem capabilities, if not dial out will not work correct. It is used to see which line to use to call a certain node. If you add the X75 flag on an analogue line, your system will try to call ISDN nodes using an analogue modem. So these are not your nodelist flags as they may represent combined ISDN and analogue flags but the flags that belong to a modem.

Setup a line.

| | |
|---|---|
| **Comment** | A description for this line. |
| **TTY device** | The tty device name without /dev/ |
| **Phone nr.** | The phone number on this line. |
| **Line Speed** | The maximum line speed for this line. |
| **Fido Flags** | The EMSI flags for this line, include your modem flags here! |
| **Line Type** | Can be POTS, ISDN, Network and Local. |
| **Available** | Is this line available for use. |
| **Honor ZMH** | Deny users during ZMH on this line. |
| **Deleted** | If this line must be deleted. |
| **Callout** | Allow calling other systems from this line. |
| **Portspeed** | The "locked" modem speed, 0 to 4000000 baud, only for serial ports. |
| **Modemtype** | The modem connected to this line. |
| **EMSI name** | The EMSI name presented for this (modem) line. |

Some examples.

MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:13:58 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS   Sat Oct 17 14:13:58 2015 UTC    Free


    6.   EDIT TTY LINE

    1.   Comment     Modem line 1 - USR v.all
    2.   TTY Device  ttyS0
    3.   Phone nr.   01279-793009
    4.   Line Speed  57.6 kbits
    5.   Fido Flags  TCE,XA,V42B,V90C
    6.   Line Type   POTS
    7.   Available   No
    8.   Honor ZMH   No          11.  Portspeed 115200
    9.   Deleted     No          12.  Modemtype USR Courier v.all
    10.  Callout     Yes         13.  EMSI name Modem line #1 USR v.All




       Enter your choice > ▮

Select menu item (1..13), "-" for previous level


vince : mbsetup

# Setup - Fidonet nodes.

## Introduction

Unlike many other bbs packages, for each node you need only one record. For each node you can set up the aka's, mail, files and session handshake. There are ten screens for each node.

## 1. Edit general Setup

This screen has the general information for the node. The contact info fields are just for private use, these fields are only used in this screen. The Outbox dir is for files to sent to the node, all file you put in there will be sent to the remote if user mbse has read/write access to the file. Only real files are allowed, not symlinks etc.

| | |
|---|---|
| **Sysop name** | The name of the system operator. The name entered here is used in Areamgr and Filemgr messages etc. |
| **Outbox dir** | Private extra outbound directory for this node. Any file dropped into this directory will be sent to the node and removed after it is sent. |
| **Pvt. phone** | Contact info, node private phone. |
| **Pvt. fax** | Contact info, node private fax. |
| **Pvt. cellphone** | Contact info, node cellphone (GSM). |
| **Pvt. e-mail** | Contact info, node e-mail address. |
| **Pvt. remark** | Contact info, node remark. |
| **Route via** | A route via Fido address. All mail for this node will be send via this Aka, even mail and files for other networks. This can be useful if this node has internet access so you can send everything to this node over the internet. Note that to force calling this node via internet, ISDN or pots can also be done by using nodelist flags override. See section 3. This setting is more useful to force routing via another node or aka. |
| **Netmail direct** | Set "direct" flag in netmail to this node. |
| **Netmail crash** | Send netmail always "crash" to this node. |
| **Netmail hold** | Put mail on "hold" for this node. |
| **Pack mail** | Should net- and echo mail be packed in arcmail archives. |
| **Send notify** | Send automatic generated notify messages. |
| **Language** | The language to use for areamgr msgs. |
| **Deleted** | If this node must be deleted. |

## 2. Edit nodes aka's

In this screen you can set up 20 aka's for the node.

```
 File  Edit  View  Bookmarks  Settings  Help
 MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:25:33 2015 TZUTC 0100
 Copyright (C) 1997-2015 M. Broek     INS    Sat Oct 17 14:25:33 2015 UTC   Down


       7.2  EDIT NODES AKA'S

       1.    2:250/32@fidonet              11.
       2.    24:30/125@sportnet            12.
       3.                                  13.
       4.                                  14.
       5.                                  15.
       6.                                  16.
       7.                                  17.
       8.                                  18.
       9.                                  19.
      10.                                  20.



       Enter your choice > -

 Select menu item (1..20), "-" for previous level



               vince : mbsetup
```

## 3. Edit node session

In this screen you can set up the session defaults.

| | |
|---|---|
| Session passwd | This is the mailer session password. |
| Dial command | You can put an override here for the normal dial command. If you leave this empty the command from the modem setup is used. |
| Phone number 1 | An alternative phone number to dial. |
| Phone number 2 | An alternative phone number to dial. Use these above commands if the node has another phone number as mentioned in the nodelist. |
| Nodelist flags | Override for the nodelist flags, the nodelist flags are completely ignored if you enter something here so make sure all flags you need are here. All aka's of this node will use the same flags. This can be useful to force calling this node via the internet and to force a certain protocol. |
| Inet hostname | The FQDN host name or IP address of this node. This replaces the original system name from the nodelist and this is then used to make the internet TCP/IP connection. This can be useful if the FQDN is not present in the nodelist. |
| Outbound sess. | Outbound session method, choices are **Direct** and **Directory**. Direct means normal classic mailer connections. Directory means that everything is put in a directory from which the other node can fetch the mail and files using ftp protocol. |
| Inbound sess. | Inbound session method, choices are **Direct** and **Directory**. Direct means normal classic mailer connections. Directory means that inbound mail and files are expected in a directory were it is stored by the node using ftp protocol. |
| No EMSI | Disable EMSI handshake. |
| No YooHoo/2U2 | Disable FTSC-0006 handshake. |
| No Filerequest | Disable file request from this node. |
| Don't call | Do not call this node. |
| 8.3 names | Set this if the node only accepts 8.3 file names. |
| No PLZ | Disable PLZ compression on outgoing calls with the binkp protocol. |
| No GZ/BZ2 | Disable GZ and BZ2 compression on outgoing calls with the binkp protocol. |
| No Zmodem | Disable Zmodem protocol. |
| No Zedzap | Disable Zedzap protocol. |
| No Hydra | Disable Hydra protocol. |
| Binkp CRC32 | Enable CRC32 file transfer checks. |
| Binkp old esc | Some systems followed the wrong character escape method as published in FSP-1011 |

revision 3. For these systems turn this flag on or you won't be able to send long file names with spaces in the name or other special characters. Such known systems are Argus and Irex up to 2.29. Others may exist, even some older mbse versions.

No binkp/1.1   Irex from version 2.24 at least up to 2.29 have a buggy binkp/1.1 implementation. Set this to Yes for nodes that use one of these Irex versions and the binkp session will fall back to the binkp/1.0 protocol.

Ign. Hold      Ignore the nodelist flags Hold or Down. Set this only if you want to call a node that is Down or Hold and you know that you can connect it.

```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:28:26 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek     INS    Sat Oct 17 14:28:26 2015 UTC   Down


   7.3  EDIT NODE SESSION

   1.    Session passwd      ****************
   2.    Dial command
   3.    Phone number 1
   4.    Phone number 2
   5.    Nodelist flags
   6.    Inet hostname
   7.    Outbound sess.  Direct    14.  NR mode       No
   8.    Inbound sess.   Direct    15.  CRC mode      No
   9.    No EMSI         No        16.  No PLZ        No
   10.   No YooHoo/2U2   No        17.  No GZ/BZ2     No
   11.   No Filerequest  No        18.  No Zmodem     No   21.  Binkp old esc No
   12.   Don't call      No        19.  No Zedzap     No   22.  No binkp/1.1  No
   13.   8.3 names       No        20.  No Hydra      No   23.  Ign. Hold     No

         Enter your choice > -

 Select menu item (1..23), "-" for previous level


              vince : mbsetup
```

## 4. Edit node - mail processing

PKT password     The password to insert in .pkt files.

Check PKT pwd    Check password in received .pkt files. If not, errors or missing passwords are only logged. If set, errors or missing password are refused and the .pkt files are renamed to .bad

Mail forward     Not in use yet.

ARCmail comp.    Use ARCmail 0.60 file naming convention for out of zone mail.

ARCmail a..z     Allow a..z last character for ARCmail file names.

```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:29:34 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek     INS    Sat Oct 17 14:29:34 2015 UTC   Down


     7.4  EDIT NODE - MAIL PROCESSING

     1.    PKT password    ****************
     2.    Check PKT pwd No
     3.    Mail forward  No
     4.    ARCmail comp. Yes
     5.    ARCmail a..z  Yes
     6.    Archiver      ZIP












         Enter your choice > -

 Select menu item (1..6), "-" for previous level


              vince : mbsetup
```

## 5. Edit mail groups

Here you can tag which mail groups are available for this node. Note that all groups are visible here, even for networks this node has no aka's in. Be careful not to allow a node to connect areas from networks he has no aka in.

```
 File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:30:10 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 14:30:10 2015 UTC    Down


        7.5   MAIL GROUPS

        1.    AUTOCREADAS
        2.    FIDONET
        3.    LOCAL
        4.    NEWS GROUPS
        5.    NOGROUP
        6.    PRIV-2-SYSOP
        7.    Sportnet






        Enter your choice > -

Toggle item (1..7), "-" for previous level, "*" (de)select all



      vince : mbsetup
```

## 6. File setup

| | |
|---|---|
| Files password | The password for .tic files. |
| Incl. message | Send a netmail message for each file to send. |
| Send TIC file | Send .tic file to this node. |
| Advanced TIC | Send advanced or standard .tic files. |
| Advanced SB | Send advanced Seen-By lines, some systems can't handle this! |
| To line in TIC | Send the To line in the tic file. Some brain dead systems will forward this line instead of processing it and then the next system in the line will fail. |
| File forward | Forward TIC files for this node (not yet ? ). |

```
 File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:30:47 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 14:30:47 2015 UTC    Down


        7.6   EDIT NODE - FILES PROCESSING

        1.    Files password ****************
        2.    Incl. message  No
        3.    Send TIC file  Yes
        4.    Advanced TIC   Yes
        5.    Advanced SB    Yes
        6.    To line in TIC No
        7.    File forward   No
        8.    TIC 4d address No






        Enter your choice > -

Select menu item (1..8), "-" for previous level



      vince : mbsetup
```

## 7. File groups

The same story as for mail groups is true for the file groups.

```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:31:07 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 14:31:07 2015 UTC   Down


      7.7  FILE GROUPS

      1.    ALLFILES      11.   FIDOGAZETTE   21.   OREDSON S/W   31.   STN
      2.    APPLEMAC      12.   FIDONEWS      22.   OS2           32.   TABLETS
      3.    BBBSFDN       13.   FILEMGR       23.   PASCALFDN     33.   TAURUS FDN
      4.    BFDS          14.   GFD           24.   PDN           34.   UTILNET
      5.    CHEEPWARE     15.   LNX4          25.   PERLNET       35.   W32BINKD
      6.    COBOL-TOOLS   16.   MBSE          26.   R50           36.   WINDOWS
      7.    DBRIDGE FDN   17.   MYSTIC        27.   RAR
      8.    ECHOLIST      18.   NASA          28.   SDS
      9.    FERNWOOD      19.   NODEDIFFS     29.   SFDS
     10.    FIDO-ADMIN    20.   ODDBALL       30.   SPORTNET




      Enter your choice > -

Toggle item (1..36), "-" for previous level, "*" (de)select all



                    vince : mbsetup
```

## 8. Directory session

Set up in this screen is needed if you selected Inbound and or Outbound session as Directory session. The node is then getting and bringing mail using the ftp protocol and no normal mailer connects are being used any more. This may be a good solution for points, they only need a simple mail reader and a tosser. The mail is transferred by the **mbfido** program, during the toss or tic command after the normal inbound is processed **mbfido** will look for nodes inbound directories to process. When mail is added to the outbound, mail for nodes with directory sessions is not stored in the normal Binkley Style Outbound but in the session outbound directory.

You need to set up a ftp account for this node and the best thing is to set this up so that the other node is forced in a chroot environment, ie. make it so that the node only sees his own directories and cannot leave that. The ftp account needs to be a member of group bbs because mbfido needs access to the files as well as the node. The nodes account needs also to be a member of group bbsftp for the chroot environment. The node needs umask 002 so that files that he puts there will have group read/write access. During toss mbfido also stores the files with group read/write access so that the node can remove the files when he has fetched all files. Directory locking is supported and is compatible with other mailers that support ftp. Here is a small sample of protftp.conf for use with directory sessions:

```
# This is /etc/proftpd.conf for use with mbsebbs ftp mail
# sessions and anonymous ftp access.
# It assumes that you have a user/group "nobody" and "ftp"
# for normal/anonymous operation.
# It also assumes you have a group bbsftp were ftp mail users
# must be a member of.
#
ServerName                      "Seaport ProFTPD server"
ServerType                      inetd
DefaultServer                   on
#
# Port 21 is the standard FTP port.
Port                            21
# Umask 002 is to force new dirs and files to be group writable.
Umask                           002
#
# To prevent DoS attacks, set the maximum number of child processes
# to 30.
MaxInstances                    30
#
# Set the user and group that the server normally runs at.
User                            nobody
Group                           nogroup
#
SystemLog                       /var/log/proftpd
TransferLog                     /var/log/xferlog
#
# Anonymous ftp and members of group bbsftp have a chroot environment.
DefaultRoot     ~ bbsftp
#
# Normally, we want files to be overwriteable.
<Directory /*>
  AllowOverwrite                on
</Directory>
#
# A basic anonymous FTP server configuration.
# To enable this, remove the user ftp from /etc/ftpusers.
<Anonymous ~ftp>
  RequireValidShell             off
  User                          ftp
  Group                         ftp
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias                     anonymous ftp
  #
  # Limit the maximum number of anonymous logins
  MaxClients                    50
  #
  # We want 'welcome.msg' displayed at login, and '.message' displayed
  # in each newly chdired directory.
```

```
 DisplayLogin                  welcome.msg
 DisplayFirstChDir             .welcome
 DisplayReadme                 README
 #
 # Display .dotfiles and .dirs
 LsDefaultOptions              "-a"
 #
 # Limit WRITE everywhere in the anonymous chroot
 <Limit WRITE>
   DenyAll
 </Limit>
 #
 # An upload directory that allows storing files but not retrieving
 # or creating directories.
 <Directory incoming/*>
    <Limit READ>
      DenyAll
    </Limit>

    <Limit STOR>
      AllowAll
    </Limit>
 </Directory>
 #
</Anonymous>
```

Here are some examples to create an account, first create the extra group bbsftp:

groupadd bbsftp

Then create accounts like this:

```
useradd -g bbs -G bbsftp -d /opt/mbse/var/bbsftp/irex -s /bin/bash -c "FTP Mail account" irex
```

Then give the node a password for the ftp login. Note that this account has a shell, proftpd needs a valid shell to allow access. If the node tries to telnet to your bbs he can't login with this account if you use **mblogin**, this is because the node is not in the bbs user file. If you system allows it, the node can access the shell using rlogin or ssh.

Now the items to setup for the directory sessions:

Outbound settings

| | |
|---|---|
| Files path | The path were to store the outbound mail and files. |
| Check for lock | If we need to check for a lock file before adding files. |
| Wait clear lock | Wait for the lock to clear if present or not. If set to yes and there is a lock created by the node then mbfido will wait for maximum 10 minutes for the lock to disappear. If set to no, mbfido skips this node and will add mail and files at a later time. |
| Check lockfile | The full file name and path of the lock file to check for. |
| Create lock | If mbfido should create a lock during the time it will add mail and files. |
| Create lockfile | The full file name and path of the lock file to create. |

Inbound settings

| | |
|---|---|
| Files path | The path from were to get the inbound mail and files. |
| Check for lock | If we need to check for a lock file before getting files. |
| Wait clear lock | Wait for the lock to clear if present or not. If set to yes and there is a lock created by the node then mbfido will wait for maximum 10 minutes for the lock to disappear. If set to no, mbfido skips this node and will scan this directory at a later time. |
| Check lockfile | The full file name and path of the lock file to check for. |
| Create lock | If mbfido should create a lock during the time it will get mail and files. |
| Create lockfile | The full file name and path of the lock file to create. |

7.8  EDIT NODE DIRECTORY SESSION

         Outbound settings
1.   Files path        /opt/mbse/var/bbsftp/irex/outbound
2.   Check for lock  Yes              3.   Wait clear lock Yes
4.   Check lockfile  /opt/mbse/var/bbsftp/irex/lock.bsy
5.   Create lock       Yes
6.   Create lockfile /opt/mbse/var/bbsftp/irex/lock.bsy

         Inbound settings
7.   Files path        /opt/mbse/var/bbsftp/irex/inbound
8.   Check for lock  Yes              9.   Wait clear lock Yes
10.  Check lockfile  /opt/mbse/var/bbsftp/irex/lock.bsy
11.  Create lock       Yes
12.  Create lockfile /opt/mbse/var/bbsftp/irex/lock.bsy

         Enter your choice >

Select menu item (1..12), "-" for previous level

## 9. Security

This will set the security bits for the node. These bits can be used to give access or deny access to certain echomail areas or tic file areas. Default bit 1 is set for each node.

```
File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 15:34:56 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 14:34:56 2015 UTC    Down

   7.9   NODE SECURITY FLAGS

    1. Everyone        X    12. Flags 12      -    23. Flags 23        -
    2. Flags 2         X    13. Flags 13      -    24. Flags 24        -
    3. Flags 3         -    14. Flags 14      -    25. Flags 25        -
    4. Flags 4         -    15. Flags 15      -    26. Flags 26        -
    5. Flags 5         -    16. Flags 16      -    27. Flags 27        -
    6. Flags 6         -    17. Flags 17      -    28. Flags 28        -
    7. Flags 7         -    18. Flags 18      -    29. Flags 29        -
    8. Flags 8         -    19. Flags 19      -    30. Flags 30        -
    9. Flags 9         -    20. Flags 20      -    31. Flags 31        -
   10. Flags 10        -    21. Flags 21      -    32. Flags 32        -
   11. Flags 11        -    22. Flags 22      -



       Enter your choice > -

Select menu item (1..32), "-" for previous level



                    vince : mbsetup
```

## 10. Area and File managers

This screen sets names and passwords of the uplink managers, also our own password is set here.

| | |
|---|---|
| Uplink AreaMgr program | The name of the Areamgr program of this node. This could be AreaFix, AreaMgr etc. |
| Uplink AreaMgr password | The password for the Areamgr of this node. |
| Uplink AreaMgr is BBBS | If the uplink uses BBBS software set this to Yes. |
| Uplink FilMgr program | The name of the Filemgr program of this node. This could be FileMgr, Allfix, Raid etc. |
| Uplink FileMgr password | The password of the Filemgr if this node. |
| Uplink FileMgr is BBBS | If the link uses BBBS set this to Yes. |
| Our Area/Filemgr passwd | Our own Area/Filemgr password for this node to send to our node. |
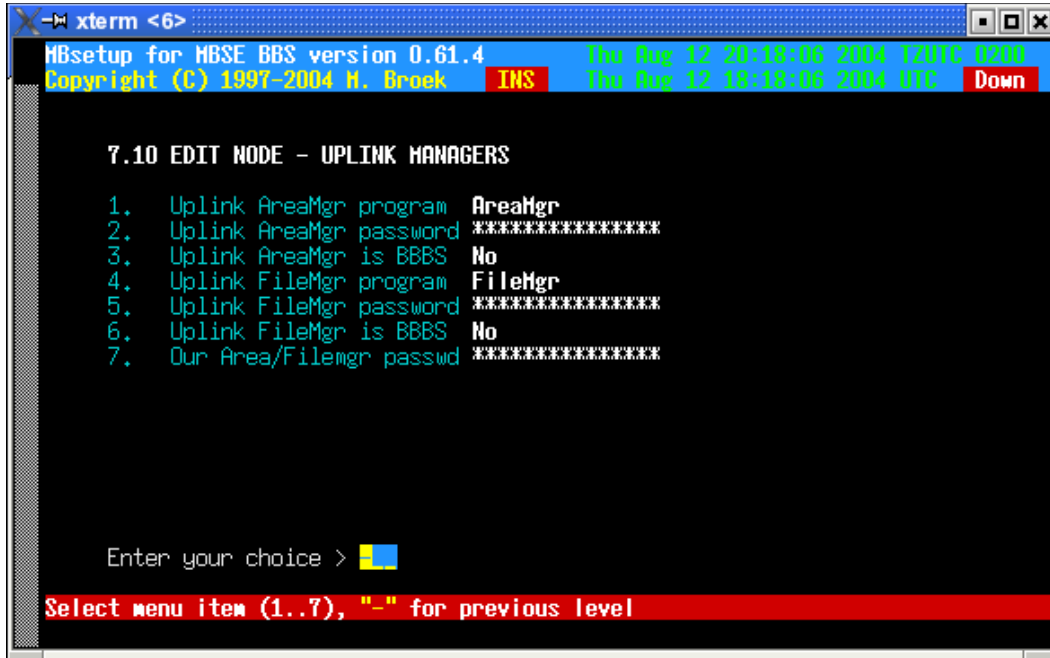


## 11. Statistics

In this statistics screen you can see the mail and files flow with this node. Values are stored for the current week, the previous week, the current month and previous month and the overall total since you defined this node. There are actual 12 months of statistics stored in the nodes record, only 2 are visible.
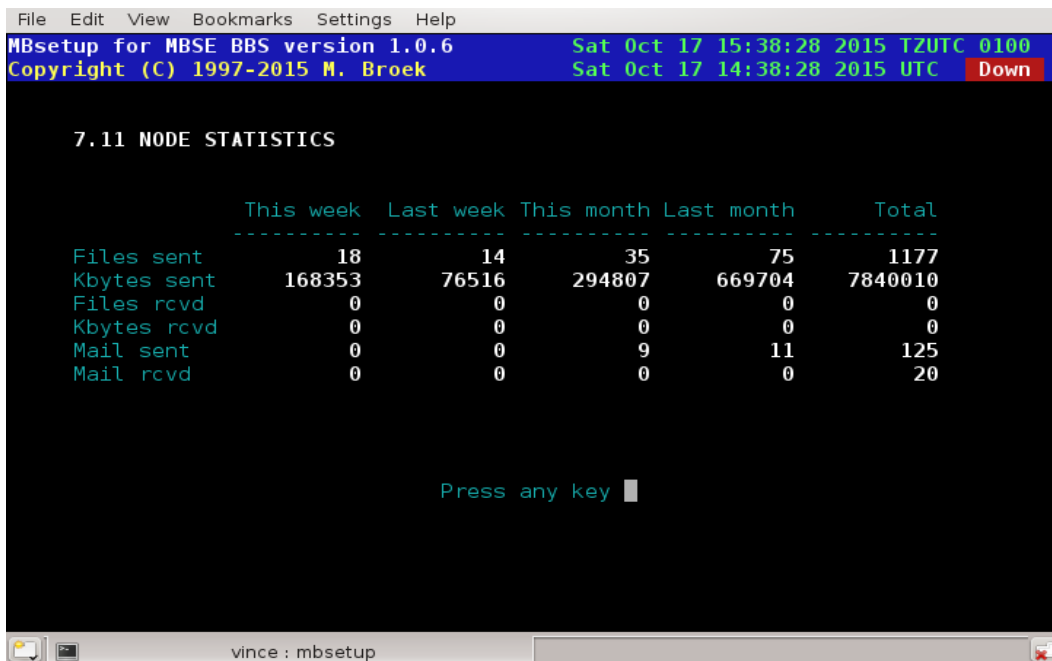
# Setup - Edit BBS Setup.

**Edit BBS Set up.**

The BBS setup is split in the following sections:

1. Security limits
2. Language setup
3. BBS menus
4. File areas
5. Transfer protocols
6. Oneliners

# Setup - BBS Setup - Security Limits.

**Security limits, introduction.**
Every BBS needs several security limits to make a difference between several user groups. These are the twits, regular users and (co-)sysops. If you have a donation system you will probably have more different levels. Every level a user can have must have a record in this file. To operate MBSE BBS you need at least 3 levels, twit with level 0, new users with the level as setup in 1.7.1, and the sysop level as setup in 1.5.4 As said, for special user groups you can add more levels as you need. If a user logs in and has a level in the user base you didn't define here, he won't be able to login. Even the twit level needs some access to be able to throw him out in a nice but friendly way, give him 5 minutes, 1 file to download and no more then 1 Kb so he will understand he is not wanted. Some defaults are installed during first bbs setup.

**Limits setup**

| | |
|---|---|
| **Access level** | The access level value. |
| **Maximum time** | The maximum time each day. If set to zero, the user gets 24 hours a day. |
| **Download Kb.** | Maximum Kilobytes download each day, zero is unlimited. |
| **Download Files** | Maximum files to download each day, zero is unlimited. |
| **Description** | The description for this level. |
| **Available** | If this level is available. |
| **Deleted** | If this level must be deleted. |

So if you completely want to disable a users ratio's, set Maximum time, Download Kb and Download files to zero. A users session is still limited to 24 hours.

# Setup - BBS Setup - Language Setup.

**Language introduction.**

You need to define at least one language, this is the default language. The ISO language code of the default language needs to be the same as in the global set up 1.5.24 and should be set to **en**. I did this so you can make your own local languages next to the default languages. If something is not present in your local language, the BBS will fall back to the default language. It is therefore wise to name all menus, macro templates and text files the same for all languages used. When you set up the languages for the first time, entries for English, Dutch, Spanish, Galego, German, French and Chinese languages are created.

And important note about the ISO code, don't change this. And if you do, then you *must* change the directory name under `/opt/mbse/share/int` to the same name as the code here. For example for the **en** code the system expects the following paths and files:

/opt/mbse/share/int/language.en

/opt/mbse/share/int/menus/en/*.mnu

/opt/mbse/share/int/txtfiles/en/*.ans

/opt/mbse/share/int/txtfiles/en/*.asc

/opt/mbse/share/int/macro/en/*

**Language setup.**

| Selec t | The letter to select this language. |
| **Name** | The name of this language. |
| **ISO code** | The 2 letter ISO code of the language. |
| **Available** | If this language is available. |
| **Security** | The minimum security level to select this language. |
| **Deleted** | If this language must be deleted. |

# Setup Menu System.

**Introduction.**
One of the most powerful features of the BBS is it's menu system. You have complete control over each individual menu item which can be restricted according to criteria such as security levels.

**ANSI Screens.**
For the menus to work properly you can draw ANSI screens, this is what the users will see. For GNU/Linux there is "Duh DRAW" written by Ben Fowler, see sunsite.unc.edu /pub/Lunux/docs. If you can't find it or have no internet access, you can also use THEDRAW. This utility can be found on many BBS'es around the world. Unfortunately it is a DOS program so you will need dosemu on your GNU/Linux box or a separate DOS computer. You can define main screens and include screens for each menu, the include screen may for example show the keys that you have available in every menu. See the list of control codes.

**Display lines.**
It is also possible to display menu lines with the built in display option. The used colours are selectable, a normal colour and a bright colour. The normal colour is the default, you can toggle bright on and of using the ^ in the display line. If you end a menu display line with a ; then no newline is send after that line. If you want to output the ^ or ; characters you need to escape them with a backslash like this: \; or \^. The order of menu entries is important.

**Automatic commands.**
A menu function is usually executed when a user presses the hot-key assigned to that particular menu item. But menu functions can also be executed automatically. Each menu item contains an AutoExec field. By default this field is set to No, but by toggling it to Yes, the menu item can be made to execute when it is played back (displayed) by the BBS.
As you read through the menu function types outlined in this chapter, you may come to realize that this is a very powerful feature. For example, when used with the menu function that displays a text file, you can design very elaborate, graphical text file menus that you wouldn't normally be able to display in a line-by-line menu.
Automatic menu execution can be used in many other instances as well. Just to give you some ideas, it might be used to display a text file to users who have a security level equal to or greater than a certain level. Yet another use is to execute multiple function menus which are used to execute several functions when a single command is entered.

**Multiple languages.**
For each language you can define a set of menus. Only for the default language all menus must exist. It makes sense to make the file names of your menus for each language the same and not to translate them. If a menu is missing for a non default language, the menu from the default language path is used instead.

**Editing a menu.**
The order of the menu lines in the setup is not important except for the autoexec menus, they must be placed in the right order from start, ie. begin with the menu specific screen display, then the global include display and finally show the prompt.

```
MBsetup for MBSE BBS version 1.0.6            Sat Oct 17 15:49:36 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS    Sat Oct 17 14:49:36 2015 UTC    Free


    8.3. EDIT MENU ITEM

    1.  Sel. key  S
    2.  Type nr.  7        Run external program in shell
    3.  Opt. data /usr/local/bin/safe-door /usr/local/etc/safe/safe-config
                  1234567890123456789012345678901234567890123456789012345678901234
    4.  Display          ^[^S^]^    Safe Cracker
    5.  Security  0    ------------------------------
    6.  Min. age  0                   12. No door.sys No
    7.  Lo-colors Normal display color 13. Use COMport No
    8.  Hi-colors Bright display color 14. Run nosuid  Yes
    9.  Autoexec  No                   15. No Prompt   Yes
    10. Door Name Safe Door            16. Single User No
    11. Y2K style Yes                  17. Hidden door No



        Enter your choice > -

Select menu item (1..17), "-" for previous level
```

```
                    vince : mbsetup
```

1.  **Selection key.** This is the key a user must press to activate this menu. This field is ignored when AutoExec is set to Yes.
2.  **Type nr.** this is the menu type to execute. For a description of all available types see the links at the top of this page.
3.  **Optional data.** Some menus need optional data, for example the function goto another menu needs the name of that menu file here.
4.  **Display.** What is to be displayed to the user. You can use this instead of ANSI screens.
5.  **Security.** This is the minimum security level to execute this selection. The security is a level number combined with 32 bit mapped flags. NOTE: level 0 and no flags means everyone can select this menu. Good for logout options and all other options everyone must be able to execute.
6.  **Min. age.** The minimum age the user must be to execute this selection. You may want to restrict access to certain areas to users older than 18 years. If you leave this to 0, every one can execute this menu.
7.  **Lo-colors.** The normal display colour for the display line.
8.  **Hi-colors.** The bright display colour for the display line.
9.  **AutoExec.** If this is an automatic executed selection.
10. **Door Name** The name of the door that will be displayed to other users. This name is also used to count the same doors running if the door only allows a single user. It is important that you use the same name in every language menu file for the same door. This item is only visible with menu type 7.
11. **Y2K style** Writes the dates in the door.sys file in the new style, with 4 digit year numbers, else the old 2 digit style is used. This item is only visible with menu type 7.
12. **No door.sys** Suppress writing of a door.sys file in the users home directory. This item is only visible with menu type 7.
13. **Use Comport** Writes COM1: with 19200 as baud rate to the door.sys file, this is for dosemu with the vmodem patch. This item is only visible with menu type 7.
14. **Run nosuid** If set to Yes the door will run in non-suid mode. Most doors need this. This item is only visible with menu type 7.
15. **No Prompt** If set to Yes then when the door is finished there will be no prompt to press Enter, instead the bbs menu will show up immediately. This item is only visible with menu type 7.
16. **Single User** Set to yes of the door only allows one user at the same time. Make sure you have set a unique name that must be the same for the door in every language file so that the bbs is able to count the instances of the door running. This item is only visible with menu type 7.
17. **Hidden door** Set to yes to hide the door from being visible for other users. If someone is using that door the whoson list displays "External Door" instead of the door name. This may be good to hide some special sysop only doors. This item is only visible with menu type 7.

*Final warning.*

If a sub-menu is missing, the BBS falls back to the main menu. This menu must be called "main" (or else set another name in the global set up) or your BBS won't start and complain.
Sub-menus may be nested 50 levels deep.

# Global Menus.

1. **Goto another menu:** This will start the execution of another menu. The current menu level is not stored on the stack.
   **Optional data:** The name of the new menu.
2. **Gosub another menu:** This will start the execution of another menu. The current menu level is stored on the stack. Gosub's may be nested 50 levels deep.
   **Optional data:** The name of the new menu.
3. **Return from Gosub:** This will go back one gosub level. If you are already at the top level nothing happens.
   **Optional data:** None.
4. **Return to top menu:** Return to the top (main) menu. The name of this menu is set in the global setup. Default is main.mnu
   **Optional data:** None.
5. **Display .a?? file with control codes:** This will display an ANSI file to the user. If the user has *Graphics No* set then the ASCII version is shown. Search is done first in the users language path and if that fails the default language path is used. Control codes in the file are substituted with the current values the represent.
   **Optional data:** The name of the file to display. Do not give the file name extension!
6. **Show menu prompt:** Display the menu prompt.
   **Optional data:** The prompt to display. This string may contain some control characters that are replaced with information. The prompt is displayed in White on Black and is hard coded at the moment.
     - **~** This will insert the number of minutes the user has left.
     - **@** This will insert the name of the current file area.
     - **^** This will insert the name of the current message area.
     - **#** This will insert the current local time.
7. **Run external program:** This will execute external programs.
   **Optional data:** The full path and file name of the external program to run. There are a few switches you can give on the command line:
     - **/N** will be replaced by the current node number. The node number is faked by using the record number of the tty lines setup.
     - **/A** will prompt for a file name to enter. The file name the user enters is then replaced on the command line. This is a dangerous option!
     - **/T=your prompt** is an alternate prompt for entering a file name if used together with the **/A** option.
8. **Show product information:** This will show copyright information about MBSE BBS.
   **Optional data:** None.
9. **Display todays callers:** This will display a list of todays callers to the BBS.
   **Optional data:** "/H" Show handles instead of real names, "/U" show Unix names instead of real names.
10. **Display userlist:** Display all users in the users database except those that are hidden.
   **Optional data:** "/H" Show handles instead of real names, "/U" show Unix names instead of real names.
11. **Time statistics:** Display the users time statistics.
   **Optional data:** None.
12. **Page Sysop:** Page sysop for a chat.
   **Optional data:** A message to the user
   The message to the user could be something like "Calling sysop, please wait ..." or "I will see if Michiel wants to chat with you, please wait!" As sysop you will know best what to put in that line.
13. **Terminate call:** Terminate this call and hangup.
   **Optional data:** None.
14. **Make a log entry:** This will write a line in the logfile.
   **Optional data:** The information you want in the logfile.
15. **Print text to screen:** Write text to the users screen.
   **Optional data:** The text that must appear on the users screen. The @ character is replaced with a newline.
16. **Who is online:** Displays the who is online list and what they are doing. Users that are hidden are not displayed.
   **Optional data:** "/H" Show handles instead of real names, "/U" show Unix names instead of real names.

17. **Comment to sysop:** Enter the text editor and let the user write a message to the sysop. The area is predefined in the global setup.

    **Optional data:** None.

18. **Send online message:** Send an online message to a user on another line.

    **Optional data:** "/H" Use handles instead of real names, "/U" use Unix names instead of real names.

19. **Display textfile with more:** This will display a text file to the user. After each full screen the user is prompted with *More Y/n/=*.

    **Optional data:** The full path and file name to the file.

20. **Display .a?? file with control codes and wait:** This will display a ANSI or ASCII file to the user with control codes and wait for *Enter* when it is finished.

    **Optional data:** The file name without extension of the file to display.

21. **Display line** This entry does nothing except that it displays the text on the display line. This is always displayed, even if the display line is empty. In that case an empty line is displayed.

    **Optional data:** None.

22. **Chat session** This connects the user to the chat server. The user is connected with his unix name as default name. The chat server is a very simple IRC look alike server.

    **Optional data:** None yet, maybe the channel later.

# File Area Menus.

101. **Select another area:** This option will show a list of available areas and let the user select a new area. If there is optional data the new area will be selected without user intervention.
    **Optional data:** If there is an option the area is direct selected. Current options are: **F+** goto next available area. **F-** goto previous available area.

102. **File List:** This option will display a list of files with their dates, sizes and description. During the display of the list the user can select (Tag) files for later download.
    **Optional data:** None.

103. **View File:** Not yet implemented.
    **Optional data:** None.

104. **Download File(s):** This option will start to transmit files to the user if he has tagged files for download. Tagging files for download can be done during File List, Keyword Scan, File name Scan or New file Scan. If a user didn't select a transfer protocol before now he will be forced to select a file transfer protocol.
    **Optional Data:** None.

105. **Raw Directory:** This option will display the contents of a directory in raw format.
    **Optional data:** If the option is **/F** the contents of the current directory is shown. If the option is the full path to a directory, the contents of that directory is shown.

106. **Keyword Scan:** This option will search for files in the files database for a matching keyword. The search is not case sensitive. If there are files found the user is able to select (Tag) these files for later download.
    **Optional data:** None.

107. **file name Scan:** This option will search for a file name match in the files database. The search is not case sensitive. If there are files found the user is able to select (Tag) these files for later download.
    **Optional data:** None.

108. **Newfiles Scan:** This option will scan for new files available for download since the last time the user was online. As option the user can change that date from which to start the search. Any files found the user may select (Tag) for later download.
    **Optional data:** None.

109. **Upload:** This option will let the user upload files to the bbs. If the current area has an alternate upload area, the upload will end up in that area. If the user uses X-modem or another ancient protocol he will first be asked for a file name. Normal modern protocols don't need this. The file name is checked before the transfer is done to protect the bbs. Further the files the user will upload will at first be placed under the users home directory **~/upl**. After the upload(s) the files are checked for viruses. If all is well, the file is imported in the bbs. If the file contains a valid FILE_ID.DIZ file inside the archive, that file will be used for the description of the upload, if not, the up loader will have to describe the file.
    **Optional data:** None.

110. **Edit Taglist:** This option is for the user to edit the list of files he has tagged for later download.
    **Optional data:** None.

111. **View file in homedir:** Not yet implemented.
    **Optional data:** None.

112. **Download Direct:** Download a file direct.
    **Optional data:** The full path and file name to the file to download.

113. **Copy file to Homedir:** This option will copy a file from a download directory to the users home directory. It will be checked if the user has enough room in his directory, the default Quota for users is 10 Mbytes.
    **Optional data:** None.

114. **List Homedir:** This option will list the files in the users home directory.
    **Optional data:** None.

115. **Delete in Homedir:** This option will let the user delete one or more files from his home directory.
    **Optional data:** None.

116. **Unpack file in Homedir:** Not yet implemented.
    **Optional data:** None.

117. **Pack files in Homedir:** Not yet implemented.
    **Optional data:** None.

118. **Download Homedir:** This option will let the user download from his home directory.

**Optional data:** None.
119. **Upload Homedir:** This option will let the user upload files to his home directory.
     **Optional data:** None.

# Message Area Menus.

201. **Select another area:** This option will show a list of all available areas and let the user select a new area. If there is optional data the area will be selected without user intervention.

   **Optional data:** If there is an option the area is direct selected. Current options are **M+** goto the next available area. **M-** goto the previous available area. If you use the **U+** and **U-** instead, the area is changed to the previous or next area with unread messages, areas without unread messages are skipped. Another option is the **N**, if this is used for area select the areas with new unread mail are marked with a yellow star.

202. **Post a Message:** This option lets the user post a new message.

   **Optional data:** None.

203. **Read Messages:** This option lets the user read messages. If he has done that before in that area he will be suggested to start after the message he has last read. During reading of messages the user can reply to other messages.

   **Optional data:** None.

204. **Check for Mail:** Check for new arrived mail.

   **Optional data:** None.

205. **Quickscan Messages:** Make a quick overview list of all messages in that area.

   **Optional data:** None.

206. **Delete a Message:** This option will let the user delete a specific message. He must the the owner of that message or have sysop rights in that area to be able to delete a message.

   **Optional data:** None.

207. **Mail Status:** This gives a complete overview of all available mail at the bbs.

   **Optional data:** None.

208. **OLR: Tag Area:** This option lets the user tag one or more areas to be included in his off line mail packet.

   **Optional data:** None.

209. **OLR: Untag Area:** This option lets the user un tag one or more areas not to be included in his off line mail packet.

   **Optional data:** None.

210. **OLR: View Tags:** This option lets the user view which areas will be included in his off line mail packet.

   **Optional data:** None.

211. **OLR: Restrict Date:** Not yet implemented.

   **Optional data:** None.

212. **OLR: Upload Mail:** Let the user upload mail or a new off line reader set up. The packet format is automatic detected. Currently BlueWave is supported. QWK support will be added later.

   **Optional data:** None.

213. **OLR: Download BlueWave:** Download mail in BlueWave version 2 format.

   **Optional data:** None.

214. **OLR: Download QWK:** Download mail in QWK format.

   **Optional data:** None.

215. **OLR: Download ASCII:** Download mail in flat ASCII format. Not yet implemented.

   **Optional data:** None.

216. **Read Email** Read users private email.

   **Optional data:** None.

217. **Write Email** Post an email message.

   **Optional data:** None.

218. **Trash Email** Put email in the trash can. Not Yet implemented.

   **Optional data:** None.

219. **Choose Mailbox** Choose another private mailbox. Valid boxes: mailbox (normal in/out), archive and trash.

   **Optional data:** If there is an option the area is direct selected. Current options are **M+** goto the next mailbox. **M-** goto the previous mailbox.

220. **Quickscan Email** Make a quick overview list of all messages in the selected e-mail area.

   **Optional data:** None.

221. **Show area rules** Show the rules of a echo mail area if such rule file is present in the rules directory. The

file searched for needs to have a name that is the same as the area tag, or the area tag plus .rul, or the first eight characters of the area tag plus .rul like the rule files in the elist package.
**Optional data:** None.

# User Settings Menus.

301.**Change Transfer Protocol:** Let the user select a new file transfer protocol.
   **Optional data:** None.
302.**Change Password:** Let the user change his Fido Net password.
   **Optional data:** None.
303.**Change Location:** Let the user change his location.
   **Optional data:** None.
304.**Obsolete:** This command does not exist.
   **Optional data:** None.
305.**Change Voicephone:** Let the user change his voice phone number.
   **Optional data:** None.
306.**Change Dataphone:** Let the user change his data phone number.
   **Optional data:** None.
307.**Change Expertmode:** This command will be removed.
   **Optional data:** None.
308.**Obsolete:** This command does not exist.
   **Optional data:** None.
309.**Change Date of Birth:** Let the user set a new date of birth. Checks are done if the date is more or less realistic. This command should not be made available users if you use the regular date of birth validation check.
   **Optional data:** None.
310.**Change Language:** Let the user select a new default language.
   **Optional data:** None.
311.**Change Hotkeys:** Let the user toggle the use of Hot keys on or off..
   **Optional data:** None.
312.**Change Handle:** Let the user select a new handle (nickname).
   **Optional data:** None.
314.**Change Don't Disturb:** Let the user toggle the "do not disturb" flag.
   **Optional data:** None.
315.**Change check for new files:** Let the user toggle the "check for new files at logon" flag.
   **Optional data:** None.
316.**Change fullscreen editor:** Let the user toggle the "use full screen editor" flag.
   **Optional data:** None.
317.**Change FS edit shortcut keys:** Let the user toggle the "FS edit short cut keys between Emacs flag.
   **Optional data:** None.
318.**Change Address:** Let the user change his three lines home address.
   **Optional data:** None.
319.**Change signature:** Let the user change his mail signature file (.signature in his home directory). He is allowed to edit at most 4 lines of 74 characters.
   **Optional data:** None.
320.**Change OLR Extended Info:** Let the user toggle the setting of the Off line Reader download of extended info (kludges).
   **Optional data:** None.
321.**Change character set:** Let the user change the default character set for messages.
   **Optional data:** None.
322.**Change archiver:** Let the user change the archiver to use to build off line reader packages.
   **Optional data:** None.

# Oneliner Menus.

401. **Oneliner Add:** Let the user add a new one liner.
 **Optional data:** None.
402. **Oneliner List:** Let the user list all the available one liners.
 **Optional data:** None.
403. **Oneliner Show:** Let the user show a specific one liner.
 **Optional data:** None.
404. **Oneliner Delete:** Let the user delete a one liner. In order to do so he must be the owner of that one liner or he must have sysop access level. The one liner is not really removed, only marked for deletion.
 **Optional data:** None.
405. **Oneliner Print:** Show a random chosen one liner on the screen. If you make this command automatic, each time that this menu is executed a new one liner will pop up.
 **Optional data:** None.

# Setup - BBS Setup - File Areas.

**File Areas Introduction.**
This is the set up for the file areas in which users can up and download files. This database has fixed area numbers, the database can't be packed. Extending the database is always possible. One important note, before you can set this up, you need to define the new files groups.
If you change the path in an existing area, all files in that area are automatic moved to the new location.

| | |
|---|---|
| **Area Name** | The area name. |
| **Path** | The full path to the file area. |
| **Down Sec.** | The download security level. |
| **Upl. Sec.** | The upload security level. |
| **List Sec.** | The security level to list the files in this area. |
| **Available** | If this area is available. If you make an available area unavailable the area is checked if there are downloadable files in that area. If so, you can't remove the area. If there are none you will be asked if you are sure. After the area is made unavailable, the record is filled with reasonable default values. |
| **Check New** | Check this area for new files if a user logs in. |
| **Dupecheck** | Check this area for duplicates if a user uploads a file. |
| **Free area** | If all files in this area are free. |
| **Direct DL** | Allow direct download from this area. |
| **Pwd upl.** | Allow users to password protect their uploads. |
| **Filefind** | Search in this area for filefind requests. |
| **Add alpha** | Add .tic files alphabetic sorted or at the bottom of the list. |
| **File req.** | Allow File Request from this area. |
| **BBS group** | Not in use yet. |
| **New group** | New files announce group. See New files groups for more info. |
| **Min. age** | Minimum user age to access this area. |
| **Password** | The password for this area. If blank, no password is needed. |
| **DL days** | How long must a file not been downloaded to (re)move it. |
| **FD days** | How old must a file be to be (re)moved. |
| **Move area** | The area to move the file to, if zero it is deleted. |
| **Archiver** | The archiver to use to repack the files with. |
| **Upload** | Alternate upload area. If a user uploads a file in this area, it will be placed in the alternate area. If the value is zero, the file will be placed in the current file area. On CD-rom areas you must not leave this zero unless you set the upload security level so high that nobody can upload in this area. |

MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:06:40 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS    Sat Oct 17 15:06:40 2015 UTC    Down

8.4 EDIT FILE AREA

```
1.   Area Name Dos etc - Nodediffs all Nets
2.   Path      /home/mbse/ftp/pub/fido/nodelist
3.   Down Sec. 0      -------------------------------
4.   Upl. Sec. 0      -------------------------------
5.   List Sec. 0      -------------------------------
6.   Available Yes
7.   Check new Yes      13. Add alpha Yes        19. DL days   0
8.   Dupecheck Yes      14. File req. Yes        20. FD days   180
9.   Free area No       15. BBS Group NODEDIFFS  21. Move area 0
10.  Direct DL Yes      16. New group NODEDIFFS  22. Archiver
11.  Pwd upl.  No       17. Min. age  0          23. Upload    3
12.  Filefind  Yes      18. Password  ********************
```

          Enter your choice > -

Select menu item (1..23), "-" for previous level

# Setup - BBS Setup - File Transfer Protocols.

**Introduction.**

MBSE BBS has Xmodem, Ymodem, Ymodem-1K, Ymodem-G, Zmodem and Zmodem-8K (aka ZedZap) build in. In addition some external protocols are added to the setup but they are disabled by default. When the bbs is started the first time, a set of default protocols is created. The code used is based on the code from lrzsz package wich is based in the original code written by Chuck Forsberg.

Ymodem is receiver driven. That means if the user has selected plain Ymodem at the bbs and his local client is using Ymodem-G and when the user starts a download, the files are sent with Ymodem-G and not with plain Ymodem. With the same configuration an upload will be sent with plain Ymodem. With downloads, the Ymodem at the bbs will use what the client wants: 128 or 1K data blocks, crc of checksum, normal or streaming Ymodem-G.

Zmodem is transmitter driven. That means if the user has selected Zmodem-8K at the bbs and his local client is using normal Zmodem and when the user starts a download, the download is sent with Zmodem-8K. With the same configuration an upload will be sent with plain Zmodem. With uploads, the Zmodem at the bbs doesn't care what is being used, it will adapt to the client program.

These days nobody should use Xmodem any more but when I wrote Ymodem you also get Xmodem because they are the same. Only with Xmodem the user has to type in the file name to both sides. If you enable it you are on your own and you may need to change the sources to make it really work because I didn't add typing in the file name at the bbs. Also, Xmodem is restricted to 8.3 file names and the bbs uses long file names.

**Transfer Protocols Setup.**

| | |
|---|---|
| **Select Key** | The key the user has to press to select this protocol. |
| **Name** | The name of this protocol. |
| **Upload** | The full path and file name and parameters to upload files. |
| **Download** | The full path and file name and parameters to download files. |
| **Available** | If this protocol is available. |
| **Internal** | If this is internal or external protocol. |
| **Advice** | A small advice to the user shown before the transfer starts. |
| **Efficiency** | The efficiency in percent. Has no real meaning. |
| **Deleted** | If this protocol must be deleted. |
| **Sec. level** | The security level a user must have to select this protocol. |

Some fields cannot be changed when this is an internal protocol, as they are hard coded.
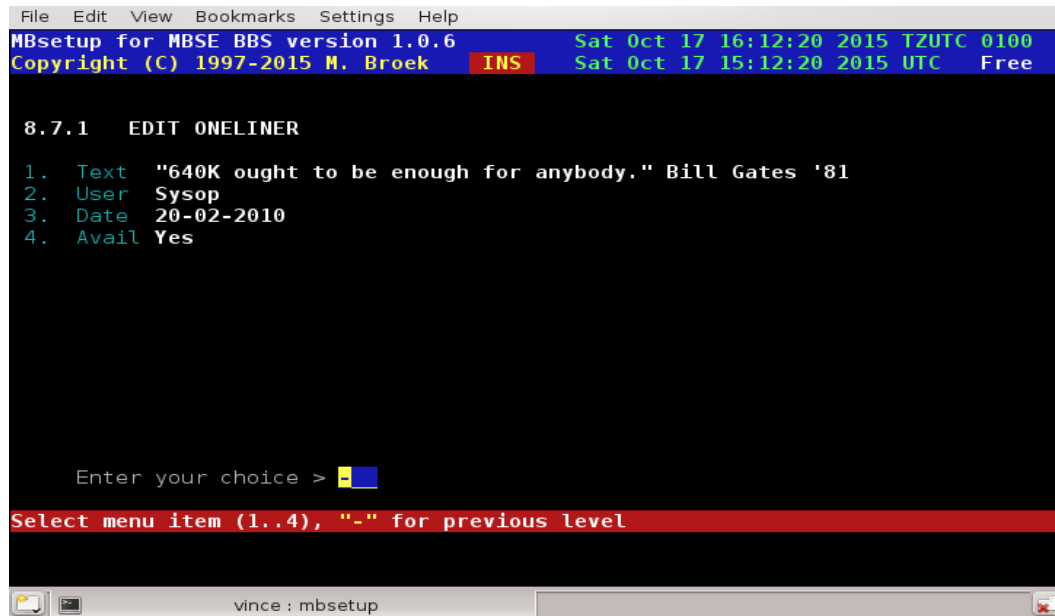
# Setup - BBS Setup - Oneliners.

**Oneliners.**

One liners are small quotes that can be random selected and displayed to your users. From the same database one liners can be selected and inserted at the bottom of messages. With the oneliners set up you can edit, add, delete and import one liners. Import is done from plain ASCII text files, one quote on each line. The lines should be maximum 70 characters long.

```
File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:12:20 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek     INS    Sat Oct 17 15:12:20 2015 UTC    Free


  8.7.1   EDIT ONELINER

  1.   Text   "640K ought to be enough for anybody." Bill Gates '81
  2.   User   Sysop
  3.   Date   20-02-2010
  4.   Avail  Yes







       Enter your choice > -
Select menu item (1..4), "-" for previous level


           vince : mbsetup
```

# Setup - Mail Setup.

**Edit Mail Setup.**

The Mail Set up is split in the following sections:
1. Echo mail groups
2. Echo mail areas

# Setup - Mail Setup - Message Groups.

**Introduction.**

Message Groups are to logically divide your echo mail areas in groups for different mail networks. It makes sense to select the groups by uplink, and areas file that is available for each network. When you have defined a Areas file and a message is received while you have no area for it, the area will be created with the defaults that are set in the message area group. This will only work if you receive a message from the uplink defined in this group, not from any downlinks. A note about the **Start at** area you can define, this may be higher then your number of areas currently is. As soon as a area is created in this group the areas database will be expanded so that it fits. Select these values carefully, make sure you have room for all possible areas in that group and skip a number of extra areas for future expansion.

**Message Group Setup.**

| | |
|---|---|
| **Name** | The name of the group. |
| **Comment** | The description of the group. |
| **Base path** | The base for the JAM path without the area name. |
| **Read sec** | The read security for bbs users for auto created areas. |
| **Write sec** | The write security for bbs users for auto created areas. |
| **Sysop sec** | The sysop security for bbs users for auto created areas. |
| **Link sec** | The link security to copy to new created areas. |
| **Start at** | Start area number (lowest) to create new areas. |
| **Net reply** | Netmail reply board for auto created areas. |
| **Users del** | User may delete for auto created areas. |
| **Aliases** | Are aliases allowed in auto created areas. |
| **Quotes** | Append a quote when user writes in auto created areas. |
| **Active** | If this group is active. |
| **Deleted** | If this group must be deleted. |
| **Auto change** | Change areas when running **mbfido areas**. |
| **User change** | Create/delete areas when users connect/disconnect. |
| **Use Aka** | The Fidonet aka to use. |
| **Uplink** | The Fidonet aka of your uplink. |
| **Areas** | Areas file in the path defined in 1.4.6 screen 2. |
| **Charset** | Default character set to use. |
| **GoldED grp** | Group number for GoldED include file. |

```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6        Sat Oct 17 16:14:11 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS   Sat Oct 17 15:14:11 2015 UTC    Down


   9.1 EDIT MESSAGE GROUP

   1.   Name        FIDONET
   2.   Comment     Fidonet areas
   3.   Base path   /home/mbse/var/mail/fido
   4.   Read sec  0        --------------------------------
   5.   Write sec 0        --------------------------------
   6.   Sysop sec 0        --------------------------------
   7.   Link sec           X-------------------------------
   8.   Start at  6         14. Deleted    No          20. Charset   CP437 2
   9.   Net reply 1         15. Auto change Yes         21. GED grp   #003
   10.  Users del Yes       16. User change Yes
   11.  Aliases   No        17. Use Aka     2:250/1@fidonet
   12.  Quotes    No        18. Uplink      1:261/38@fidonet
   13.  Active    Yes       19. Areas       BACKBONE.NA

        Enter your choice > _

Select menu item (1..21), "-" for previous level


              vince : mbsetup
```

# Setup - Mail Setup -> Mail Areas.

**Introduction.**
Mail areas have fixed area numbers, just like file areas. But if you want you can move areas. Some off line readers don't like changing the area numbers. All areas can be of the following types: local, netmail, echomail and news. If you gate news and echomail in the same area, then select echomail if this area is a Fidonet area, select news if this area is a Usenet news area.

There is a global set up tool, if you are in the main message area setup screen then you can choose the Global command. This will allow you to do bulk changes on areas selected by the mail groups. Options are delete connection, add new connection, replace connection, change connection status, change days old for purge, change maximum messages for purge, change user security, change aka to use, change origin line, change netmail reply board and delete message area.

**Message Area Setup.**

| | |
|---|---|
| **Area Name** | The description of the area. |
| **Area Tag** | The echomail area tag. |
| **Group** | The name of the echomail group. |
| **Newsgroup** | The newsgroup name if you are gating this area. |
| **JAM base** | The path and name of the JAM message base without extension. |
| **Origin** | The origin line to add to echomail messages. |
| **Fido Aka** | The Fidonet aka for this area. |
| **QWK name** | The name of the area for OffLine Readers. |
| **Distrib.** | Leave this blank for now! |
| **Area Type** | Local, Netmail, Echomail or News. |
| **Msg Kinds** | Private/Public, Private, Public, Read Only. |
| **Charset** | Default character set for the area |
| **Active** | If this area is active. |
| **Days Old** | The maximum age of the messages before purging. |
| **Max. Msgs** | The maximum messages in this area. |
| **Articles** | The maximum news headers to fetch. |
| **Netreply** | Netmail reply area if this is echomail. |
| **Read Sec.** | The security level to read messages. |
| **Write Sec.** | The security level to write messages. |
| **Sysop Sec.** | The security level to do sysop actions. |
| **User Del.** | Users may delete their own messages. |
| **Aliases** | Allow the use of an Alias. |
| **Quotes** | Add a random quote under new messages (One liner database). |
| **Mandatory** | If this area is mandatory for downlinks. |
| **UnSecure** | Don't check link address during toss (Dangerous!). |
| **OLR Default** | Switch area default on for OLR users. |
| **OLR Forced** | Area is always on for OLR users. |
| **Connections** | This will take you to the screen to edit up and downlinks. |
| **Security** | The security flags for downlinks to access this area. |

**Edit connection**

| | |
|---|---|
| **Aka** | The network address. |
| **Send to** | Export mail to this node. |
| **Recv from** | Import mail form this node. |
| **Pause** | The node is temporary disconnected. |
| **Excluded** | The node is disconnected by the sysop. |
| **Delete** | Delete this connection. |

A note about the excluded switch. This can be used to disconnect a node from the area by moderator request. AreaMgr requests from that node for this area are not processed any more, he cannot disconnect this area and reconnect. If you want to block access to the area in a certain group for a node who has access to the group you should use the Security flags and also set the correct security flags for the node.

**Global Commands.**

From menu 9.2 you can enter the global commands menu. In this menu you can:

1. Delete connection
2. Add new connection
   Replace connection
3. Change connection status
4. Change days old
5. Change max. messages
6. Change max. articles
7. Change bbs security
8. Change link security
9. Change aka to use
10. Change origin line
11. Change netmail reply
12. Delete message area

After you have selected the action you want and added the items to do, you will see a screen were you can select message area groups. You can then tag one or more groups and press enter when you are done. Then you have one chance to perform the actions or to bail out. All areas matching in that group are affected by your changes. If you are not happy with the result, don't save the database and no harm is done. The file mbsetup.log shows all affected areas.

**Automatic area creation.**

If you want to use the automatic area creation you have to define this in the message area groups. See option 19.

# Setup - File Echo's Setup.

The File Echo's Set up is split in the following sections:

1. File echo groups
2. File echo areas
3. Hatch manager
4. Magic files

# Setup - File Echo's Setup - File Groups.

**Introduction.**
File echo groups are to logically divide your file echo's for different file distribution networks. This screen also has a lot of fields that are needed for the auto area create. When a new area is created a bbs download area is made and a new tic area record is made. It makes sense to select the groups by uplink and area file that is available for that file distribution network. By doing that downlinks can connect areas that are not yet connected at your bbs but are available from your uplink. If you are connected to the filegate then there is one area file for all distribution groups. You must then create a file group for each distribution group, ie. Fidonet, SDS etc. The name of that group must exactly match the comment field in the group in this set up.

**File Group Setup.**

| | |
|---|---|
| **Name** | File Echo Group name. |
| **Comment** | The description of that group. |
| **Base path** | The base path for new created areas. |
| **Active** | If this group is active. |
| **Use Aka** | The Fidonet aka to use for this group |
| **Uplink** | The Fidonet aka of the uplink. |
| **Areas** | The name of the areas file (in ~/var/arealists). |
| **Filegate** | Is area file in filegate.zxx format or not. |
| **Banner** | Banner file to add to archives. |
| **Convert** | Archiver to convert archive to. |
| **BBS group** | BBS group name. |
| **New group** | New files announce group. |
| **Active** | If this group is active. |
| **Deleted** | If this group must be deleted. |
| **Start at** | New created areas must start at area number. |
| **Auto chng** | Auto change when running **mbfido areas** |
| **User chng** | Auto create areas on downlinks requests. |
| **Replace** | Set replace option in new areas. |
| **Dupecheck** | Set Dupe check option in new areas. |
| **Secure** | Set secure option in new areas. |
| **Touch** | Set Touch file date option in new areas. |
| **Virscan** | Set virus scanner option in new areas. |
| **Announce** | Set announce files option in new areas. |
| **Upd magic** | Set update magic name option in new areas. |
| **File ID** | Set File ID option in new areas. |
| **Conv. All** | Set convert all option in new areas. |
| **Send orig** | Set send original option in new areas. |
| **DL sec** | Set download security in new areas. |
| **UP sec** | Set upload security in new areas. |
| **LT sec** | Set list security in new areas. |
| **Upl. Area** | Set upload area number in new areas. |
| **Link sec** | Set link security to copy to new areas. |

MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:20:05 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS      Sat Oct 17 15:20:05 2015 UTC    Down

10.1 EDIT FILE GROUP

```
 1.  Name        DBRIDGE FDN
 2.  Comment     DB - D'Bridge Email software
 3.  Base path   /home/mbse/ftp/pub/local/dos-bbsmailers/
 4.  Use Aka     2:250/1@fidonet
 5.  Uplink      1:261/38@fidonet
 6.  Areas       FILEGATE.ZXX                    23. Upd magic Yes
 7.  Filegate    Yes          15. Auto chng Yes  24. File ID   Yes
 8.  Banner      banner.txt   16. User chng Yes  25. Conv. all No
 9.  Convert                  17. Replace   Yes  26. Send orig No
10.  BBS group   DBRIDGE FDN  18. Dupecheck Yes  27. DL sec    0
11.  New group   DBRIDGE      19. Secure    Yes  28. UP sec    0
12.  Active      Yes          20. Touch     Yes  29. LT sec    0
13.  Deleted     No           21. Virscan   Yes  30. Upl. area 3
14.  Start at    1            22. Announce  Yes  31. Link sec
```

        Enter your choice > -

Select menu item (1..31), "-" for previous level

vince : mbsetup

# Setup - File Echo's Setup - TIC Areas.

**Introduction.**
Here you can define the File Echo's or TIC areas. Files received or send from this areas are bound together with a *.tic file with information about the file and where to store that file. Each file echo must belong to a group, in this group record is the information about cost sharing and some other details. When a file is received at your system you can do several things with that file before it is stored in your download areas such as; scanning the file for viruses, extracting the FILE_ID.DIZ file to use as description, allow update of magic alias, convert to another compression format, replace the file archive comment with an add of your own bbs and limit the number of files (nodelists).

**TIC Area Setup.**

| | |
|---|---|
| **Comment** | A description for this area. |
| **Area tag** | The tag for this area. |
| **Security** | The security access flags for downlinks. |
| **BBS area** | The BBS download area number, 0 means pass thru. |
| **Message** | Not in use yet. |
| **Group** | The group where this area belongs to. |
| **Keep #** | The number of files to keep, the name must match. |
| **Fido aka** | The Fidonet aka to use for this area. |
| **Convert** | The archiver to convert to, leave blank for none. |
| **Banner** | The banner file (in ~/etc) to replace in the archive. |
| **Replace** | Honour the "Replace" command in the .tic file. |
| **Dupecheck** | Check for duplicates in this area. |
| **Secure** | Check if the sending system is connected. |
| **Touch** | Touch re-archived files to the original date. |
| **Virus sc.** | Try to scan for viruses. |
| **Announce** | Files may be announced in this area. |
| **Upd magic** | Allow update magic request name. |
| **File_id** | Try to use the FILE_ID.DIZ file for description. |
| **Conv.all** | Convert archive even if it is already right. |
| **Send org.** | Send original received file instead of the file from the BBS. |
| **Mandatory** | Downlinks can't disconnect from this area. |
| **Notified** | Not in use yet. |
| **Upl discon** | Not in use yet. |
| **Deleted** | If this area must be deleted. |
| **Active** | If this area is active. |
| **New SR** | Give new node links the status SR instead of S. |
| **Systems** | Displayed to screen all connected systems. |

MBsetup for MBSE BBS version 1.0.6        Sat Oct 17 16:21:34 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS      Sat Oct 17 15:21:34 2015 UTC    Down

10.2 EDIT TIC AREA

1.  Comment    FidoNet: Weekly FidoNews Newsletter
2.  Area tag   FIDONEWS
3.  Security   X------------------------------
4.  BBS area   129: Dos etc - Documents  12. Dupecheck  Yes    20. Send org.   No
5.  Message                              13. Secure     Yes    21. Mandatory   No
6.  Group      FIDONEWS                  14. Touch      Yes    22. Notified    No
7.  Keep #     12                        15. Virus sc.  Yes    23. Upl disc.   No
8.  Fido aka   2:250/1@fidonet           16. Announce   Yes    24. Deleted     No
9.  Convert                              17. Upd magic  Yes    25. Active      Yes
10. Banner     banner.txt               18. File_id    Yes    26. New SR      No
11. Replace    Yes                       19. Conv.all   No     27. Systems     5

        Enter your choice > -

Select menu item (1..27), "-" for previous level

vince : mbsetup

**Global Commands.**

From menu 10.2 you can enter the global commands menu. In this menu you can:

1. Delete connection
2. Add new connection
3. Replace connection
4. Change connection status
5. Change aka to use
6. Change security flags
7. Delete TIC area

After you have selected the action you want and added the items to do, you will see a screen where you can select TIC file area groups. You can then tag one or more groups and press enter when you are done. Then you have one chance to perform the actions or to bail out. All areas matching in that group are affected by your changes. If you are not happy with the result, don't save the database and no harm is done. The file mbsetup.log / system.log shows all affected areas.

# Setup - File Echo's Setup - Hatch Manager.

**Introduction.**
Hatch files is nothing more than entering a new file into a file echo. That file is searched for on your system in a specific directory with a certain file mask. This can be for example a new created node list or an allfiles listing. Every time **mbfido tic** is run it will scan for files defined in this set up. If such a file is found, a special .tic file is written and stored in the mailers inbound directory. When the **mbfido** starts processing .tic files, that new hatched file will be processed as if it was received form another system. To let this work, you obviously need an existing TIC area.

The file names to hatch may be long file names as long as there are no spaces in it. In the TIC file a short file name will be created for the old style tic processors. The long file name is inserted in the TIC file as well. The short file names are created in the same way as windows does.

**Hatch Manager Setup.**

| | |
|---|---|
| **Mask** | The path and file name mask to search for. "?" matches any character, "#" matches digits and "@" matches upper or lower case letters. Except for the "@" the test is case sensitive. |
| **Area** | The area to hatch this file in. |
| **Replace** | The file name to replace, ie. "nodelist.z*" |
| **Magic** | The file request magic name, ie. "nodelist" |
| **Desc** | The description of the file, %12 in the description means copy the 12th character of the name in place. |
| **Dupe** | Check for duplicates. |
| **Active** | If this area is active. |
| **Deleted** | If this area must be deleted. |
| **Days** | The days in the week to scan for this file. |
| **Month** | The dates in the month to scan for this file. |

# Setup - File Echo's Setup - Magics Files Setup.

## Introduction.

Magics are special actions that you can perform on files received in a .tic area. The actions are: copy file to a directory, unpack file in a directory, set number of files to keep, move file to another .tic area, update magic request alias, adopt file into another area, store in another path, delete file (don't process it further) and execute a command. The edit screen is different for all kinds of actions you select. More than one magic record may exist for each area. With all these actions you can for example can setup processing of nodediff's and unpacking nodelists in the nodelist directory. If you use the execute command, you can only execute commands that are in the directory $MBSE_ROOT/bin, ie. ~/bin.
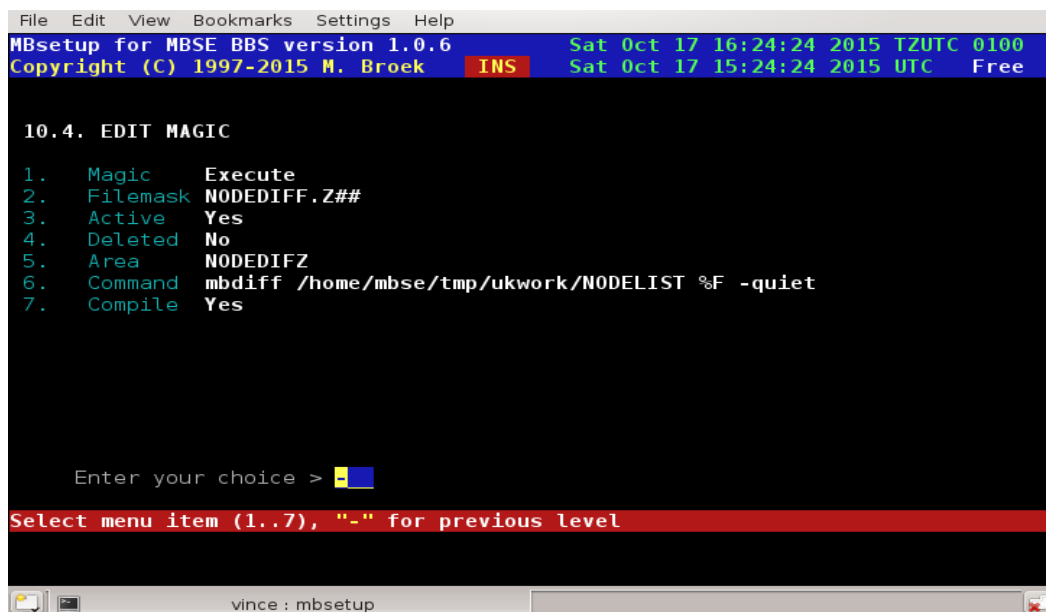
## Magics Setup.

| | |
|---|---|
| **Magic** | The action to perform, select with the space bar. |
| **Filemask** | The filemask to scan for. "?" Matches all characters, "#" matches any digit and "@" any upper or lower case letter. Except for the "@" the test is case sensitive. |
| **Active** | If this magic is active. |
| **Deleted** | If this magic must be deleted. |
| **Area** | The area in which this magic is found. |
| **To path** | The destination path. (Copy, Other path and Unpack). |
| **To area** | The destination area. (Adopt and Move). |
| **Command** | The command to execute. (Execute). |
| **Keep #** | The number of files to keep. (Keep). |
| **Compile** | Trigger "compile nodelists". (Copy, Unpack and Execute). |

## Macro's

In the command line for the magic execute command you may use macro's to replace parts of the command line. The following macro's are defined:

| | |
|---|---|
| **%F** | Replaced by the full path and file name of the file. |
| **%P** | Replaced by the full path to the file. |
| **%N** | Replaced by the file name without dot and extension. |
| **%E** | Replaced by the extension of the file name. |
| **%L** | The last 2 characters of the file name extension. |
| **%D** | The day number of the year, 3 digits. |
| **%C** | The last 2 digits of the day number of the year. |
| **%A** | The .tic area name. |

```
File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:24:24 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 15:24:24 2015 UTC    Free


   10.4. EDIT MAGIC

   1.    Magic     Execute
   2.    Filemask  NODEDIFF.Z##
   3.    Active    Yes
   4.    Deleted   No
   5.    Area      NODEDIFZ
   6.    Command   mbdiff /home/mbse/tmp/ukwork/NODELIST %F -quiet
   7.    Compile   Yes






        Enter your choice > -

Select menu item (1..7), "-" for previous level

              vince : mbsetup
```

# Setup - Newfiles Groups.

**Introduction**

The newfiles groups are there to create separate new files announcements for several networks and areas. Even if you don't want to make different announcements you still need to define at least 2 groups. One is a group where you don't announce files in and one where you do. These groups are linked to the BBS file areas and must be defined before you define the BBS file areas. As you can see in the example below, the groups are separated by subjects.

**Newfiles Groups Setup.**

**Name**          The tag name of the group.

**Comment**      The comment for this group.

**Active**          If this group is active.

**Deleted**       If this group must be deleted.

# Setup – New files Reports.

**Introduction.**
For each network you can define one or more new files reports to announce the new files that arrived on your BBS. The files to include in the reports are specified by the new files groups you can include or exclude for announcement.

**Reports Setup.**

**Comment**        The comment for this report.
**Msg area**        The JAM message base to write the report in.
**Origin line**        The origin line to use.
**From name**        The name to use in the "From:" field.
**To name**        The name to use in the "To  :" field.
**Subject**        The text to use in the "Subj:" field.
**Language**        Not in use yet, but DO select!
**Template**        Not in use yet.
**Aka to use**        The Fidonet aka to use in this area.
**Active**        If this report is active.
**Deleted**        If this report must be deleted.
**New groups**        The screen to define the groups to include.
**CHRS kludge**        The CHRS kludge an encoding to use.

# Setup - Filefind Areas.

**Introduction.**

The filefind idea on Fidonet came from the program Allfix written by Harald Harms. The idea is that a user writes a mail in a filefind area addressed to "Allfix" with in the subject line the items to search for. On all BBS's with a filefind utility those programs try to find the requested files and then produce a reply of which files they have found. That reply can be in the same area, in a special reply echo or can be sent by netmail. Usually the user gets a lot of replies from which he can see if someone has the file(s) available he was searching for.

**Filefind Setup.**

| | |
|---|---|
| **Comment** | The comment for this area. |
| **Origin** | The origin line to use for the reply. |
| **Aka to use** | The Fidonet aka to use in this area. |
| **Scan area** | The JAM area in which to scan for requests. |
| **Reply area** | The JAM area to put the replies in, leave blank if in the same area. |
| **Language** | The language to use for the reply. |
| **Template** | The name of the macro template file. |
| **Active** | If this area is active. |
| **Deleted** | If this area must be deleted. |
| **Net. Reply** | If the reply will be sent by netmail. |
| **CHRS kludge** | The CHRS kludge and encoding to use. |
| **Keywrd len** | The minimum keyword length allowed in search |

.

# Setup - Files Database.

**Introduction.**
This option allows you to manually edit the files in the files database. The option to edit the file description is not present at this moment. The basic file entries can't be changed.

**Edit File.**

| | |
|---|---|
| **file name** | The 8.3 file name of the file. |
| **Long fn** | The long file name of the file. |
| **FileSize** | The size of the file in bytes. |
| **FileDate** | The real date of the file. |
| **Last DL.** | The date of the last download of the file. |
| **Upl.Date** | The upload date. |
| **TIC Area** | The name of the tic area the file was received in. |
| **Magic** | The magic name to download/freq this file. |
| **Uploader** | The name of the uploader. Filemgr is used by mbse. |
| **Times DL** | The number of times downloaded from the BBS. |
| **Password** | The password to access this file. |
| **Deleted** | If this file should be deleted. |
| **No Kill** | Don't delete this file with **mbfile** |
| **Announced** | If this file is ever announced as new. |

# Setup - BBS Users.

**Introduction.**
This screen lets you edit some settings of the BBS users. Note that the users database will never be packed and users will always keep their record number. If a user is deleted the record will be blanked. New users will get a blank record if it exists, otherwise the database will be expanded. The reason for this, is the Last Read pointers of the message areas. BBS systems who do it in another method mostly can't keep track of the Last Read pointers and records, and they mix all users Last Read pointers. Fields that can be changed by the users themselves are in the second screen.

**Edit User**

| | |
|---|---|
| **Full Name** | The full (Fidonet) name of the user. |
| **Security** | The security level of the user. |
| **Expirydate** | The expiry date of the user DD-MM-YYYY. |
| **Expiry Sec** | The security level the user gets after expiry. |
| **Unix uid** | Displays the unix name of the user. |
| **1st login** | Displays the date of the first login. |
| **Last login** | Displays the date of the last login. |
| **Pwdchange** | Displays the date of the last password change. |
| **Credit** | The users credit. |
| **Hidden** | If the user is hidden from listings etc. |
| **Deleted** | User must be deleted. |
| **No Kill** | User can never be deleted. |
| **Comment** | The comment about this user. |
| **Locked** | Locked out of the BBS. |
| **Guest** | Is this a "guest" account on/off. |
| **Ext Info** | Send ^aKLUDGES with BlueWave downloads. |
| **Email** | If this user has an email address. |
| **Calls** | Displays the number of calls. |
| **Downlds** | Displays the number of downloads. |
| **Down Kb** | Displays the number of Kilobytes downloaded. |
| **Uploads** | Displays the number of uploads. |
| **Upload Kb** | Displays the number of Kilobytes uploaded. |
| **Posted** | Displays the number of messages posted. |
| **Time left** | Show the time left in minutes and reset the time. |
| **Screen 2** | Go to next setup screen. |

Watch out when changing the Linux / unix userid of the user, you will need to change the system password file and the users home directory manually.

```
 File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:29:05 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 15:29:05 2015 UTC    Down

   15. EDIT USER

   1.   Full Name    Vince Coen            10. Locked     No
   2.   Security     32000                 11. Guest      No
   3.   Expirydate   00-00-0000            12. Ext Info   No
   4.   Expiry Sec   32000                 13. Email      No
        Unix name    vbcoen                    Calls      30
        1st login    28-01-2008 01:11:49       Downlds    0
        Last login   06-06-2014 18:18:24       Down Kb    0
        Pwdchange    08-07-2014 13:23:25       Uploads    0
   5.   Credit       0                         Upload Kb  0
   6.   Hidden       No                        Posted     0
   7.   Deleted      No                    14. Time left  1440
   8.   No Kill      Yes                   15. Screen 2
   9.   Comment      Sysop



        Enter your choice > -

Select menu item (1..15), "-" for previous level


   vince : mbsetup
```

**Edit User Private Settings**

| | |
|---|---|
| **Handle** | The nickname of the user. |
| **Location** | The location of the user. |
| **Address 1** | The address of the user line 1. |
| **Address 2** | The address of the user line 2. |
| **Address 3** | The address of the user line 3. |
| **Voicephone** | The voice phone number of the user. |
| **Dataphone** | The data phone number of the user. |
| **Birthdate** | The birth date of the user DD-MM-YYYY. |
| **Password** | Change password for this user. |
| **Sex** | Users sex, male or female. |
| **Protocol** | The selected file transfer protocol. |
| **Archiver** | The selected archiver. |
| **Charset** | The users preferred character set. |
| **Screenlen** | The length (lines) of the users screen. |
| **Language** | The language the user uses. |
| **Hotkeys** | Hot keys on/off. |
| **colour** | Colour on/off. |
| **Fs Chat** | Full screen Chat on/off. |
| **Silent** | Do not disturb on/off. |
| **CLS** | Sent clear screen codes on/off. |
| **More** | More prompt on/off. |
| **Editor** | Select message editor. |
| **MailScan** | Scan for new mail at logon on/off. |
| **ShowNews** | Show "news" screens on/off. |
| **NewFiles** | New files scan at logon on/off. |
| **Emacs** | Use Emacs FS edit short cut keys. |

MBsetup for MBSE BBS version 1.0.6        Sat Oct 17 16:29:59 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS   Sat Oct 17 15:29:59 2015 UTC   Down

15. EDIT USER PRIVATE SETTINGS

```
 1.  Handle
 2.  Location      Roydon, Essex              14. Language   E
 3.  Address 1                                15. Hotkeys    Yes
 4.  Address 2                                16. Silent     No
 5.  Address 3                                17. CLS        Yes
 6.  Voicephone    01279-793009               18. More       Yes
 7.  Dataphone                                19. Editor     Ext.
 8.  Birthdate     23-06-1947                 20. MailScan   No
 9.  Password      **************             21. ShowNews   Yes
10.  Sex           Male                       22. NewFiles   No
11.  Protocol      Local disk                 23. Emacs      No
12.  Archiver      ZIP
13.  Charset       CP437 2
```

        Enter your choice > -

Select menu item (1..23), "-" for previous level

vince : mbsetup

# Setup - Edit Services.

**Introduction.**
Services are special mail accounts. Netmail addressed to one of these names will be handled according to the action that is selected. Current implemented actions are AreaMgr, FileMgr and Email. So if you name a service **Areamgr** and set the action to **AreaMgr** then an incoming netmail will be directed to the Areamgr function.
If you define a service **listserv** and set the action to **Email** then and incoming netmail will be converted to email and send to the **listserv** account at your host.


**Edit Services.**

| | |
|---|---|
| **Name** | The name of the Service. |
| **Type** | Toggle the service type with the space bar. |
| **Active** | If this service is active. |
| **Deleted** | If this service must be deleted. |

Here are some example services:

| | |
|---|---|
| UUCP | Email |
| allfix | FileMgr |
| areamgr | AreaMgr |
| fmail | AreaMgr |
| gecho | AreaMgr |
| mbtic | FileMgr |
| raid | FileMgr |

Note: the UUCP services is needed if you are gating email!

# Setup - Edit Domains.

**Introduction.**

The domains table is used to translate Fidonet domains to internet domains and back for the Fidonet <-> Internet gateway. When you add entries to this table make sure that the entry **.fidonet** and **.ftn** is always the last entry. This is the default entry. New added domains can be moved in place with the **M**ove command.

**Edit Domains.**

| | |
|---|---|
| **Fidonet** | The Fidonet domain to match. |
| **Internet** | The internet domain to match. |
| **Active** | If this domain is active. |
| **Deleted** | If this domain must be deleted. |

Next is an example table.

```
 File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:31:17 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS     Sat Oct 17 15:31:17 2015 UTC   Down


    17.    DOMAIN MANAGER

    1.   .z1.fidonet                    .z1.fidonet.org
    2.   .z2.fidonet                    .z2.fidonet.org
    3.   .z3.fidonet                    .z3.fidonet.org
    4.   .z4.fidonet                    .z4.fidonet.org
    5.   .z24.sportnet                  .z24.sportnet.org
    6.   .fidonet                       .ftn






       Enter your choice > -
 Rec. (1..6), "-" Back, Append, Delete, Move


            vince : mbsetup
```

# Setup - Task Manager.

**Introduction**

The task manager is the daemon which controls the MBSE BBS. It watches semaphore's and spawns programs in the background. It also watches the outbound to see if there are nodes to call. The behaviour is set up in this screen.

**Edit Task Manager**

| | |
|---|---|
| **Mailout** | Action for semaphore mailout. |
| **Mailin** | Action for semaphore mailin. |
| **Newnews** | Action for semaphore newnews. |
| **Index 1** | Nodelist compiler 1 for semaphore mbindex. |
| **Index 2** | Nodelist compiler 2 for semaphore mbindex. |
| **Index 3** | Nodelist compiler 3 for semaphore mbindex. |
| **Msglink** | Action for semaphore msglink. |
| **Reqindex** | Action for semaphore reqindex. |
| **Ping #1** | IP address of node to ping to check the internet. |
| **Ping #2** | IP address of second node to ping to check the internet. |
| **Max TCP** | Maximum simultaneous outgoing calls over the internet. |
| **Max Load** | Max. system load until processing is suspended. |
| **ZMH start** | Start of Zone Mail Hour in UTC time. |
| **ZMH end** | End of Zone Mail Hour in UTC time. |

Default are the original MBSE commands filled in, but you could also call shell scripts.

The two IP addresses to ping, need to be IP addresses, not host names. This is the most reliable way to check the connection. You should enter the IP addresses of the name servers of your own ISP here. One of these will always be up, so if one of these can be reached, the internet connection is assumed to be alive. The status of the internet connection is used to decide if it is allowed (and possible) to call TCP/IP nodes. Calling these nodes is not depending on ZMH unless the destination is not a CM node.

The **max load** setting decides at which system load point processing of background tasks will be suspended. The default value should be correct for machines slower then 200 MHz or with low ram. On faster machines you may set this value higher.

# Setup - Edit Routing Table.

**Introduction.**
First an important point: normal nodes, hubs and hosts should not need this to setup. MBSE is smart enough to do normal netmail routing using the nodelist. There are cases when you might need this, for example if you are a RC or if you have special routing arrangements via internet nodes. Think at least twice before adding entries here. You can always use the **mbfido test** command to check if the standard routing is enough. You should also check to see what you have done with **mbfido test** if the entries in this table do what you expect them to do. During route tests it is wise to temporary switch on extra debug logging with mbsetup menu 1.5.16.17 and check the information in the logfile, there is a lot of information about the decisions that are made.

When you edit a route table, the entries you add might not be in the right order, this will be done after you leave the editor. Sorting is done in order of zone, exact nodes first, then global's. If you end the routing table with a global route, ie. to All zones, then the standard routing isn't used any more and only the table is consulted. This might not be needed, most likely is that you only need to add the routes to the non-standard nodes. The table is processed from the first entry to the last and stops when an entry matches. If there was no match, the default routing will be used.

**Edit Routing Table.**
When you open the table you see something like this:

The edit screen looks like this:



| **Mask** | The destination mask to test for. |
| **Action** | Route to, Direct or Default. |
| **Dest addr** | The destination node in case of Route to. |
| **Active** | If this rule is active. |
| **Deleted** | If this rule must be deleted. |

If the Action is **Route to** then all mail matching the Mask will be routed to the node set up in Dest addr. This node must be present in your set up. If the Mask is **Direct** then mail to node node(s) matching the Mask will be sent direct, not routed. If the action is **Default** then the nodes matching the Mask will use the standard default routing. More actions will be added later.

# Setup - Internet BBS Chat.

**Introduction.**
MBSE BBS has the chat system that can be linked via internet, Internet BBS Chat (IBC) using the protocol described in FSP-9999.999. With this setup you can define other BBS systems to link with. To use IBC with other systems you need to make sure that UDP port 60179 is open on your firewall.
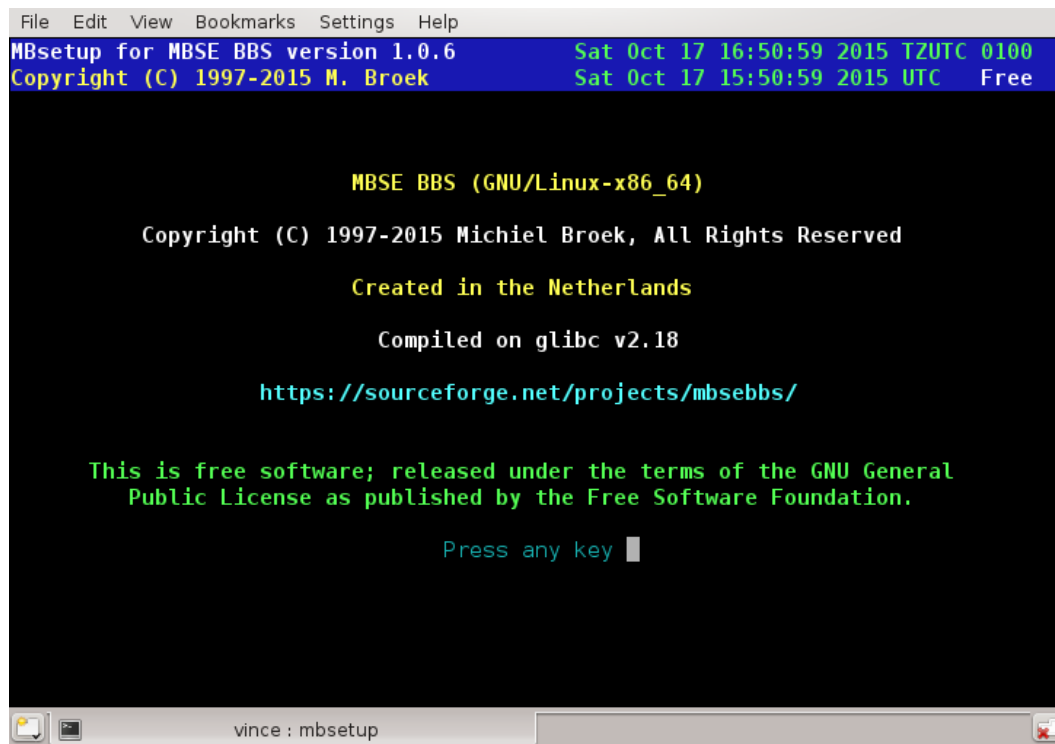
**Edit systems**

| | |
|---|---|
| **Comment** | The remote BBS name or a comment. |
| **Server** | The Full Qualified Domain Name of the remote. |
| **Dyn. DNS** | Yes if the remote uses a dyndns name. |
| **Myname** | My FQDN (default copied from 1.2.10). |
| **Password** | The password for the link. |
| **Active** | Is the link active. |
| **Deleted** | If this records must be deleted. |
| **Compress** | Use zlib compression on the link (Not yet ?). |

# Setup - Show Software Information.

**Introduction**

This screen shows the information about the MBSE BBS software including version, copyright and release policy.

# Setup - Create Sitedocs.

**Create Sitedocs**

This option creates several documents in the share/doc and share/doc/html directories under the home directory of MBSE BBS. In the first directory is a file called site.doc that is formatted to print. In the html directory are a lot of html pages with cross reference links. Since these pages contain confidential information of your users and links, you should not make these html pages publicly available, but you can browse these with a browser with → open file and point to ~/share/doc/html/index.html.

# Starting and Stopping the BBS.

Now it is time to check the starting and stopping of the BBS. As you have installed everything, setup the BBS etc, you must check if the shut down and reboot work properly. As root type **shutdown -r now** and watch the console. You should see messages that the BBS is closing while the systems shuts down. This should be one of the first things to happen. Because Slackware up to version 7.0.0 is tricky to automatic install the shutdown scripts, you won't see this happen on older Slackware versions. If you want, you can edit /etc/rc.d/rc.6 and /etc/rc.d/rc.K and insert the line ~/etc/rc.shutdown at the proper places.

When your system comes up again, one of the last messages before the login prompt appears or just before X-windows starts, you should see messages that the BBS is started.

Login as user **mbse** and check the log files if everything looks good. If something is wrong, reread the previous documentation and check if you did everything right.

Next logon to your BBS locally using the account "bbs". You can do that by typing **su - bbs** or if you already have installed **mblogin** as login replacement for telnet, then type **telnet localhost**. You will then create the first user of your BBS, this will be you, the sysop of course. After you logout the BBS start as user **mbse** the program **mbsetup** and edit your user record to set your level to that of the sysop. One more thing, the unix account you must create when you logon as new BBS user may not be **mbse** as this is the normal Admin account the BBS and its utilities use.

Now login with your unix account and see if everything still works. If you have setup **mgetty** you may want to test if users really can login with a modem. Also check a mailer session, can you dialout, ie. poll other nodes and can they call you. There is a lot that can go wrong with unix permissions if you are not precise in what you are doing.

If everything is working it is time to create poll events, and adjust other scripts to your local needs to get your BBS full up and running.

To do this you must install a crontab for user **mbse**. As user **mbse** go to the directory **~/mbsebbs-1.0.6**. In that directory type **sh ./CRON.sh** and a default crontab will be installed.

To add poll events, edit the crontab with the command **crontab -e** At the bottom of that file there is an example of how to do that. Now that the crontab is installed, all maintenance will now work, automatic dialout, scanning and tossing mail etc. In other words, the bbs is up and running.

# Setup mgetty for MBSE BBS.

To handle incoming calls you can use **mgetty** written by Gert Doering, (gert@greenie.muc.de). Others may work. You have to compile **mgetty** with the -DFIDO flag to accept Fidonet mailer calls. If you want incoming PPP calls as well, add the -DAUTO_PPP as well. If you want to use ISDN, make sure you have ISDN support compiled in the kernel. Below you can see the mgetty.config and login.config for mgetty that you may need. I have also included a part of my /etc/inittab to show how **mgetty** will spawn from **init**.

```
# inittab       This is only a part of /etc/inittab!
#               In this example it runs in runlevel 3 and 4.
#
# Serial lines
s1:34:respawn:/usr/local/sbin/mgetty -i /opt/mbse/etc/issue ttyS0
#
# ISDN lines, two channels
I1:34:respawn:/usr/local/sbin/mgetty -i /opt/mbse/etc/issue ttyI0
I2:23:respawn:/usr/local/sbin/mgetty -i /opt/mbse/etc/issue ttyI1
#
# End of /etc/inittab


# mgetty configuration file: mgetty.config
#
# ----- global section -----
#
# In this section, you put the global defaults, per-port stuff is below
#
# set the global debug level to "4" (default from policy.h)
debug 4
#
# set the local fax station id
fax-id ++31-255-515973
#
# access the modem(s) with 38400 bps
speed 38400
#
#  use these options to make the /dev/tty-device owned by "uucp.uucp"
#  and mode "rw-rw-r--" (0664). *LEADING ZERO NEEDED!*
port-owner uucp
port-group uucp
port-mode 0664
#
#  use these options to make incoming faxes owned by "root.uucp"
#  and mode "rw-r-----" (0640). *LEADING ZERO NEEDED!*
fax-owner root
fax-group uucp
fax-mode 0640
#
#
# ----- port specific section -----
#
# Here you can put things that are valid only for one line, not the others
#
# Dynalink 1428EXTRA faxmodem at port 0 (COM1).
#
port ttyS0
speed 57600
switchbd 19200
modem-type cls2
init-chat "" \d\dAT&F&C1&D3X4W2B0M0Q0V1H0&K3S0=0 OK
#
#
# ISDN ports.
# With AT&E you set the MSN/EAZ the device should listen to.
# This should for an MSN be your telephone number _with_ the area
# code but _without_ the leading 0.
# For an EAZ the last digit of your EAZ.
#
```

```
port ttyI0
modem-type data
init-chat "" ATZ OK AT&E714015437&W0 OK AT&B512 OK
#
port ttyI1
modem-type data
init-chat "" ATZ OK AT&E714017198&W0 OK AT&B512 OK
#
# end of mgetty.config


# login.config
#
# This is a sample "login dispatcher" configuration file for mgetty
#
# Format:
#       user name userid utmp_entry login_program [arguments]
#
# Meaning:
#       for a "user name" entered at mgettys login: prompt, call
#       "login_program" with [arguments], with the uid set to "userid",
#       and a USER_PROCESS utmp entry with ut_user = "utmp_entry"
#
#
# Use this one for fido calls (login name /FIDO/ is handled specially)
#
#  mgetty has to be compiled with "-DFIDO", otherwise a fido call won't
#  be detected.
#
/FIDO/  mbse    fido    /opt/mbse/bin/mbcico @
#
#
# Automatic PPP startup on receipt of LCP configure request (AutoPPP).
#  mgetty has to be compiled with "-DAUTO_PPP" for this to work.
#  Warning: Case is significant, AUTOPPP or autoppp won't work!
#  Consult the "pppd" man page to find pppd options that work for you.
#  See also PPP-HOWTO on how to set this up.
#
/AutoPPP/ -     a_ppp   /etc/ppp/paplogin
#
# This is the "standard" behaviour - *dont* set a userid or utmp
#  entry here, otherwise /bin/login will fail!
#  This entry isn't really necessary: if it's missing, the built-in
#  default will do exactly this.
#  Here we replace the standard login with MBSE BBS login:
#
*        -       -       /opt/mbse/bin/mblogin @
#
#
# end of login.config
```

I have added the **-i /opt/mbse/etc/issue** options to the **mgetty** line in /etc/inittab. The file /opt/mbse/etc/issue is a plain text file to give a small banner just before the login prompt. It could look like this:

```
       .--.        Welcome at MBSE BBS Development.
      |o_o |       -------------------------------
      |:_/ |
     //   \ \      This may or may not work today...
    (|     | )
   /'\_   _/`\
   \___)=(___/
    Powered by GNU/Linux.
```

There is a default /opt/mbse/etc/issue installed by the installation script. You need to edit this to insert your bbs name in it or even completely replace this file for a nicer one. Don't make it too big, don't put control characters or ANSI sequences in it as this may prevent some mailers to connect to your system.
I discovered that some systems don't have the right permissions on the serial port for MBSE BBS. To fix this, type

the following commands:

```
su
password: enter root password here
chmod 666 /dev/ttyS0
chown uucp.uucp /dev/ttyS0
exit
```

Note that /dev/ttyS0 is for COM1, /dev/ttyS1 for COM2 etc. This is not a nice solution. If your system allows it, try to add the the user **mbse** as a legal user of your serial ports. Most distributions have some tool for this.

# Netmail routing behaviour.

**Introduction**
The **mbfido** program that is responsible for unpacking, importing, exporting and routing of netmail has a build in default routing plan. In general this is quite simple, if we know the destination node or his uplink, (that node or uplink is in our setup), then we will route via that node in our setup. If the node or his uplink is not in our setup, then the node list is used and normal fidonet routing is used. This means, if you are a node, everything goes to your hub, if you are a hub, then mail for your downlinks will go direct to the downlinks because they are in your setup, everything else goes to the host. If you are a host, then your own downlinks will get the mail direct, the downlinks of the hubs in your net well be routed via the hubs below you. If it is for a node in your region but outside your net, mail will be routed via the other hosts in your region. Mail to outside your region will go to the region coordinators system.

**Tracking and bouncing**
At this moment there is no bouncing of undeliverable mail. I will built this in, but it will only work inside your own net. I will never include code for bouncing mail outside your net, because node lists are always not up to date.

**Special routing**
What if you need special routing. The solution is simple, add the routing nodes to your set up and fill in the "route via" field. If you don't have a session password with that node, leave the password fields blank. This node will never know that he is in your set up as long as you have the notify settings for that node switched off. To figure out such solutions yourself, I have included the flow diagrams for the tracking module.

# Main tracking routine:

```
+===========================+
|     Trackmail to dest.    |
+--------------+------------+
               |
++-------------+------------++
||   rc = GetRoute to dest  || (See next diagram).
++-------------+------------++
               |
+-------------+-------------+  yes
|       rc = R_NOROUTE      +----------------+
+-------------+-------------+        +-------+--------+
              | no                   | res: R_NOROUTE |
              |                      +===============+
+-------------+-------------+  yes
|       rc = R_UNLISTED     +----------------+
+-------------+-------------+        +-------+--------+
              |                      | res: R_UNLISTED|
              |                      +---------------+
+-------------+-------------+  yes
|        rc = R_LOCAL       +----------------+
+-------------+-------------+        +-------+--------+
              | no                   | result: R_LOCAL|
              |                      +===============+
+-------------+-------------+  no
|  routing node in setup ?  +----------------+
+-------------+-------------+        +-------+--------+
              | yes                  |   result:  rc  |
              |                      +===============+
+-------------+-------------+  no
|   "Route via" filled in ? +----------------+
+-------------+-------------+        +-------+--------+
              | yes                  |  res: R_ROUTE  |
              |                      +===============+
+-------------+-------------+
|   Change route to address |
|       result = R_ROUTE    |
+===========================+
```

# Sub function GetRoute:

```
+===========================+
|         GetRoute          |
+--------------+------------+
               |
+--------------+------------+
|      Add domain name      |
+--------------+------------+
               |
+--------------+--------------+ yes
|  Is dest our own address ?  +------------------+
+--------------+--------------+          +--------+-------+
               | no                      |   rc = R_LOCAL |
               |                         +===============+
+--------------+--------------+ yes
| Is dest our point address ? +------------------+
+--------------+--------------+          +--------+-------+
               | no                      |  rc = R_DIRECT |
               |                         +===============+
+--------------+--------------+ yes (route to boss)
|    Are we a point system    +------------------+
+--------------+--------------+          +--------+-------+
               | no                      | dest is my Boss|
               |                         |  res: R_DIRECT |
               |                         +---------------+
+--------------+--------------+ yes
| Dest. addr. in nodes setup? +------------------+
+--------------+--------------+          +--------+-------+
               | no                      |  rc = R_DIRECT |
               |                         +===============+
+--------------+--------------+ yes
| Boss of point dest in setup +------------------+
+--------------+--------------+          +--------+-------+
               | no                      |  rc = R_DIRECT |
               |                         |   dest = Boss  |
               |                         +===============+
+--------------+--------------+
| Is node listed and do we    | yes
| know his uplink in setup ?  +------------------+
+--------------+--------------+          +--------+-------+
               | no                      | dest is uplink |
               |                         |  rc = R_DIRECT |
               |                         +===============+
+--------------+--------------+ yes
|   Are we host in network ?  +------------------+
+--------------+--------------+          +--------+-------+
               | no                      | Set host addr. |
               |                         +-------+-------+
               |                                 +----------+
+--------------+--------------+ yes                         |
|    Are we hub in domain ?   +------------------+          |
+--------------+--------------+          +--------+-------+  |
               | no                      |  Set hub addr. | v
               |                         +-------+-------+  |
               |                                 |          |
               +--------------<------------------+-----<----+
               |
+--------------+--------------+
|    Set our region number    |
+--------------+--------------+
               |
               |
+--------------+--------------+ no
|      Host address set ?     +-----------------------------+
+--------------+--------------+                             |
               | yes                                        |
               |                                            |
```

```
      +--------------+--------------+
      |  Dest region <> our region  | yes                         |
      |  or Dest zone <> our zone    +------------------+          |
      +--------------+--------------+          +--------+-------+   |
                     | no                      |    Dest to RC  |  |
                     |                         |  rc = R_ROUTE  |  |
                     |                         +===============+   |
      +--------------+--------------+ yes                          |
      |      Dest net <> our net     +------------------+          |
      +--------------+--------------+          +--------+-------+   |
                     | no                      | to host destnet|  |
                     |                         |  rc = R_ROUTE  |  |
                     |                         +===============+   |
      +--------------+--------------+ yes                          |
      |       Has node a hub         +------------------+          |
      +--------------+--------------+          +--------+-------+   |
                     | yes                     | to node's hub  |  |
              +------+--------+                |  rc = R_ROUTE  |  |
              | dest is direct|                +===============+   |
              |  rc = R_ROUTE |                                    |
              +==============+                                     |
                                                                   |
                      +------------------------<------------------+
      +--------------+--------------+ no
      |      Hub address set ?        +------------------+
      +--------------+--------------+          +-------+--------+
                     | yes                     |  via our hub   |
                     |                         |  rc = R_ROUTE  |
                     |                         +===============+
      +--------------+--------------+ yes
      |  Dest node of our hub addr   +-----------------+
      +--------------+--------------+          +-------+--------+
                     | no                      |  rc = R_DIRECT |
                     |                         +===============+
              +------+-------+
              | dest is host |
              | rc = R_ROUTE |
              +==============+
```

# Internet Gateway.

**Introduction.**
The email gateway and the news gateway (along with to UUCP) is build into MBSE BBS. To route email traffic to and from the internet you need a internet MTA. I stopped using **sendmail** for this because it gave too much trouble setting it up together with MBSE BBS. Today I use Postfix, a well documented, secure and easy to set up MTA.
For the actual gate from Postfix to the BBS, use **mbmail** which you need to add to the Postfix configuration.
There may be two reasons to create a gateway, one is to gate internet news and email to the Fidonet bbs users, another reason may be that you want to make echomail as news, available on your system so that users can connect to your bbs with their favourite browser and get the mail and news using pop3 and nntp protocols. The setup is the same for both reasons so I will make one description for the whole set up.

**Setup a newsgate node with inn.**
If you only want to gate internet news to your bbs users and not want to make echo mail available as news, and you have a permanent internet connection then you don't need your own news server. This will also work on dialup links, once the internet connection is up, make sure that **mbfido news -quiet** is started as user mbse. To do this configure MBSE BBS to use the news server of your ISP in screen 1.15 with mbsetup. If you don't want to use the news server of your ISP or want to make echo mail available as news for your local users you must install your own news server. You could use inn news for a news server. To connect a small feed with your ISP you could use **suck**.
In each echo mail area you want to gate you need to fill in the newsgroup name of that area and echo mail received in that area will automatic be posted to that newsgroup. The command **mbfido news** will check all configured newsgroups for new news articles. If you set it up for the first time you need to run **mbfido news -learn** to fill the dupes database for news with all the already existing news articles. If you skip that, you may get a lot of old articles that will be gated. Just run that command once after you have set this up. Later when you receive fresh articles the command **mbfido news** will only gate new arrived articles. In each mail area you can set the number of articles to get in one run. If you set this too high, mbfido will scan all headers on the news server for new articles. This can cost a lot of time. The best thing to do is to set this value about four times higher then the average number of messages per day in that group. If you set it too low, warning messages will appear in the log file.

**Setup a newsgate with rnews.**
This is the setup if you don't want an NNTP news server like inn, but a simple cnews setup for UUCP links only. In **mbsetup** menu 1.15 (1.12 ???) you need to set this up. You need to fill in the path to the rnews program so that **mbfido** can post articles to cnews. MORE INFO NEEDED.
In each echomail area you want to gate you need to fill in the newsgroup name of that area and echomail received in that area will automatic be posted to that newsgroup.

Setup a newsgate via UUCP.
With this setup you don't run a local news server, only your bbs users and Fidonet links can then use news. You need to install **uucp** on your system. With **mbsetup** menu 1.15?? 1.12?  you need to set this up. Suppose your ISP's nodename is xs4all then all you probably need to set the UUCP path to `/var/spool/uucp/xs4all` and the UUCP node to `xs4all`. Your own node name will be your system's host name without the domain part.
In each echomail area you want to gate you need to fill in the newsgroup name of that area and echomail received in that area will automatic be posted to that newsgroup.

**Setup a email gate.**
See the Postfix (email) configuration

# Internet Gateway - Postfix setup.

Of course you need to make all these changes as root. Add the **mbmail** program as service to the postfix system by adding two lines to **master.cf**.

```
#
# Postfix master process configuration file.  Each line describes how
# a mailer component program should be run. The fields that make up
# each line are described below. A "-" field value requests that a
# default value be used for that field.
#
# Service: any name that is valid for the specified transport type
# (the next field).  With INET transports, a service is specified as
# host:port.  The host part (and colon) may be omitted. Either host
# or port may be given in symbolic form or in numeric form. Examples
# for the SMTP server:  localhost:smtp receives mail via the loopback
# interface only; 10025 receives mail on port 10025.
#
# Transport type: "inet" for Internet sockets, "unix" for UNIX-domain
# sockets, "fifo" for named pipes.
#
# Private: whether or not access is restricted to the mail system.
# Default is private service.  Internet (inet) sockets can't be private.
#
# Unprivileged: whether the service runs with root privileges or as
# the owner of the Postfix system (the owner name is controlled by the
# mail_owner configuration variable in the main.cf file).
#
# Chroot: whether or not the service runs chrooted to the mail queue
# directory (pathname is controlled by the queue_directory configuration
# variable in the main.cf file). Presently, all Postfix daemons can run
# chrooted, except for the pipe and local daemons. The files in the
# examples/chroot-setup subdirectory describe how to set up a Postfix
# chroot environment for your type of machine.
#
# Wakeup time: automatically wake up the named service after the
# specified number of seconds.  Specify 0 for no wakeup. Presently,
# only the local pickup and queue manager daemons need a wakeup timer.
#
# Max procs: the maximum number of processes that may execute this
# service simultaneously. Default is to use a globally configurable
# limit (the default_process_limit configuration parameter in main.cf).
#
# Command + args: the command to be executed. The command name is
# relative to the Postfix program directory (pathname is controlled by
# the program_directory configuration variable). Adding one or more
# -v options turns on verbose logging for that service; adding a -D
# option enables symbolic debugging (see the debugger_command variable
# in the main.cf configuration file).
#
# In order to use the "uucp" message tranport below, set up entries
# in the transport table.
#
# In order to use the "cyrus" message transport below, configure it
# in main.cf as the mailbox_transport.
#
# SPECIFY ONLY PROGRAMS THAT ARE WRITTEN TO RUN AS POSTFIX DAEMONS.
# ALL DAEMONS SPECIFIED HERE MUST SPEAK A POSTFIX-INTERNAL PROTOCOL.
#
# =========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (50)
# =========================================================================
smtp      inet  n       -       n       -       -       smtpd
pickup    fifo  n       n       n       60      1       pickup
cleanup   unix  -       -       n       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
rewrite   unix  -       -       n       -       -       trivial-rewrite
```

```
bounce     unix  -      -      n      -      0      bounce
defer      unix  -      -      n      -      0      bounce
smtp       unix  -      -      n      -      -      smtp
showq      unix  n      -      n      -      -      showq
error      unix  -      -      n      -      -      error
local      unix  -      n      n      -      -      local
cyrus      unix  -      n      n      -      -      pipe
    flags=R user=cyrus argv=/cyrus/bin/deliver -e -m ${extension} ${user}
uucp       unix  -      n      n      -      -      pipe
    flags=F user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
ifmail     unix  -      n      n      -      1      pipe
    flags=F user=fido argv=/usr/local/bin/ifmail -r $nexthop ($recipient)
mbmail     unix  -      n      n      -      1      pipe
    flags=F user=mbse argv=/opt/mbse/bin/mbmail ($recipient)
bsmtp      unix  -      n      n      -      -      pipe
    flags=F. user=foo argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient
```

In **main.cf** change or add the line:
relay_domains = $mydestination, f2802.n280.z2.fidonet.org
The fidonet (f2802.n280.z2.fidonet.org) address will be your fidonet address of course. If you have more fidonet aka's, then add them as well separated with commas.

Next you need to add **mbmail** to the **transport** file.

```
# /etc/postfix/transport
#
# execute "postmap /etc/postfix/transport" after changing this file
#
# Local destinations
#
seaport.mbse.nl        local:
www.mbse.nl            local:
news.mbse.nl           local:
#
# Fidonet mailers at this machine. Test on several strings to make sure
# it will catches everything.
#
z1                     mbmail:f2802.n280.z2.fidonet
.z1                    mbmail:f2802.n280.z2.fidonet
z2                     mbmail:f2802.n280.z2.fidonet
.z2                    mbmail:f2802.n280.z2.fidonet
z3                     mbmail:f2802.n280.z2.fidonet
.z3                    mbmail:f2802.n280.z2.fidonet
z4                     mbmail:f2802.n280.z2.fidonet
.z4                    mbmail:f2802.n280.z2.fidonet
z5                     mbmail:f2802.n280.z2.fidonet
.z5                    mbmail:f2802.n280.z2.fidonet
z6                     mbmail:f2802.n280.z2.fidonet
.z6                    mbmail:f2802.n280.z2.fidonet
fidonet                mbmail:f2802.n280.z2.fidonet
.fidonet               mbmail:f2802.n280.z2.fidonet
fidonet.org            mbmail:f2802.n280.z2.fidonet
.fidonet.org           mbmail:f2802.n280.z2.fidonet
```

Don't forget to run **postmap /etc/postfix/transport**. Now all files are changed, run **postfix reload** to activate the changes.

# Nodelist and Nodediff processing.

**Introduction.**

We received a lot of questions about nodelist and nodediff processing, so I will describe here the set up of the development system for the Fidonet nodelist. First of all, it is **very important** that you use three separate directories to do the nodelist processing. This is to make sure that all stages are independent of each other, and if something goes wrong, you still have a working system. The three directories are:

1. **/opt/mbse/ftp/pub/fido/nodelist**, this is the public download area, the received diff's are stored here as well as the final compressed nodelists for download.
2. **/opt/mbse/tmp/nlwork**, this is the working directory to apply diffs to the previous nodelist. This directory should always contain the latest **uncompressed** nodelist.
3. **/opt/mbse/var/nodelist**, this is the systems nodelist directory defined in mbsetup, menu 1.4

In short, the steps to process the nodediff's is as follows:

1. Receive the nodediff and store it for download.
2. Apply the diff to the latest nodelist.
3. Hatch the new compressed nodelist.
4. Store the new nodelist for download.
5. Unpack the new nodelist in the nodelist compiler directory.
6. Set the compile semaphore.
7. Compile the nodelists.

Now for these steps in detail.

**The download area.**

First define the download area for the bbs. In my case, this is area 20. From here users can download the nodelists and nodediffs, files to the downlinks are send from here. Below is the example of my system.



**The NODEDIFF tic area**

From your uplinks you usually receive NODEDIFF or NODEDIZZ files. Create a tic area for that purpose. I have keep# set to 5, this means the last 5 diff's are stored in the download directory, older ones are removed. Now you can receive nodediff files, store them for download, and send them to other nodes.

```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:56:04 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS    Sat Oct 17 15:56:04 2015 UTC    Down

   10.2 EDIT TIC AREA

1.   Comment    Fido Nodelist in Zip format from 2:0/0
2.   Area tag   NODELISZ
3.   Security   X----------------------------
4.   BBS area   132: Dos etc - Nodediffs  12. Dupecheck Yes  20. Send org. No
5.   Message                              13. Secure    Yes  21. Mandatory No
6.   Group      NODEDIFFS                 14. Touch     No   22. Notified  No
7.   Keep #     4                         15. Virus sc. Yes  23. Upl disc. No
8.   Fido aka   2:250/1@fidonet           16. Announce  Yes  24. Deleted   No
9.   Convert                              17. Upd magic Yes  25. Active    Yes
10.  Banner     banner2.txt               18. File_id   Yes  26. New SR    No
11.  Replace    Yes                       19. Conv.all  No   27. Systems   7



         Enter your choice > -

Select menu item (1..27), "-" for previous level


       vince : mbsetup
```

**Apply the diff.**

We do this with the tic **magic** processor. In this example I have NODELIST.007 in the **/opt/mbse/tmp/nlwork** directory. Note that this file name is upper case, they are usually stored and distributed as upper case names. As I receive the diff files as zip, the file mask on my system is **nodediff.z##**. This means that the file with the name nodediff.z14 in the area NODEDIZZ is a match. The command that is executed expands to **mbdiff /opt/mbse/tmp/nlwork/NODELIST  /var/spool/mbse/ftp/pub/fido/nodelist/nodediff.z14  -quiet** if the received nodediff is **nodediff.z14.**

The mbdiff program applies **nodediff.z14** against **NODELIST.007** in the **/opt/mbse/tmp/nlwork** directory. If this is successful, a new **NODELIST.014** is created there, as well as a compressed **nodelist.z14** and **NODELIST.007** is removed.

If this operation fails, only **NODELIST.007** will stay in that directory. Because the ARC program for GNU/Linux isn't good for files, I left the Arc files command empty in the archiver setup. As a fall back the mbdiff program uses **zip** to create the compressed archive.

If creating the new nodelist fails for some reason, a missed diff or so, the whole processing stops here. The previous nodelist is still here and you can manually correct the situation. So, if you missed a diff, see that you get it and manually give the **mbdiff** commands as user **mbse** until you are up to date. Or, place the latest uncompressed nodelist in the directory **/opt/mbse/tmp/nlwork**.



```
File  Edit  View  Bookmarks  Settings  Help
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 16:57:14 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek    INS    Sat Oct 17 15:57:14 2015 UTC    Free


   10.4. EDIT MAGIC

1.    Magic     Execute
2.    Filemask  NODEDIFF.Z##
3.    Active    Yes
4.    Deleted   No
5.    Area      NODEDIFZ
6.    Command   mbdiff /home/mbse/tmp/ukwork/NODELIST %F -quiet
7.    Compile   Yes






         Enter your choice > -

Select menu item (1..7), "-" for previous level


       vince : mbsetup
```

**Processing the new nodelist.**

Now that we have created the new compressed nodelist, it has to go somewhere. The file **nodelist.z14** is in the directory **/opt/mbse/tmp/nlwork**. The example for the hatch manager is shown below. The hatch manager runs automatic with the command **mbfido tic**. This set up will hatch the new nodelist in the tic area **NODELIST** The two screens below show the hatch and tic set up for this area.



```
File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6              Sat Oct 17 16:59:10 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS     Sat Oct 17 15:59:10 2015 UTC    Free


   10.3 EDIT HATCH MANAGER

   1.   Mask          /home/mbse/tmp/ukwork/nodelist.z??
   2.   Area          NLISTZIP
   3.   Replace
   4.   Magic         NLISTZIP
   5.   Desc          Zip'd Fidonet Nodelist
   6.   Dupe          Yes
   7.   Active        Yes
   8.   Deleted       No
   9.   Days          Sun, Mon, Tue, Wed, Thu, Fri, Sat
   10.  Month         None




        Enter your choice > -
   Select menu item (1..10), "-" for previous level


                   vince : mbsetup
```



```
File   Edit   View   Bookmarks   Settings   Help
MBsetup for MBSE BBS version 1.0.6              Sat Oct 17 17:00:32 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS     Sat Oct 17 16:00:32 2015 UTC    Down

   10.2 EDIT TIC AREA

   1.   Comment    Fido Nodelist in Zip format from 2:0/0
   2.   Area tag   NODELISZ
   3.   Security   X-----------------------------
   4.   BBS area   132: Dos etc - Nodediffs   12. Dupecheck Yes   20. Send org. No
   5.   Message                               13. Secure    Yes   21. Mandatory No
   6.   Group      NODEDIFFS                  14. Touch     No    22. Notified  No
   7.   Keep #     4                          15. Virus sc. Yes   23. Upl disc. No
   8.   Fido aka   2:250/1@fidonet            16. Announce  Yes   24. Deleted   No
   9.   Convert                               17. Upd magic Yes   25. Active    Yes
   10.  Banner     banner2.txt                18. File_id   Yes   26. New SR    No
   11.  Replace    Yes                        19. Conv.all  No    27. Systems   7



        Enter your choice > -
   Select menu item (1..27), "-" for previous level


                   vince : mbsetup
```

Now that we have hatched the new nodelist and processed it the new nodelist is stored in in the download area, and maybe send some copies to downlinks. We now have to feed it to the nodelist compiler for our own system. We use a tic **magic** command to do that. In this case we unpack the nodelist in **/opt/mbse/var/nodelist** and set the **compile** semaphore so that the **mbindex** will compile the new nodelist. Don't be afraid that the unpacked nodelists will accumulate in the nodelist directory, **mbindex** will handle that, only the latest two nodelists are kept there. The **mbindex** program is started by the task manager **mbtask**.

File   Edit   View   Bookmarks   Settings   Help

**10.4. EDIT MAGIC**

1.    Magic       Unpack file
2.    Filemask    NODELIST.Z##
3.    Active      Yes
4.    Deleted     No
5.    Area        NLISTZIP
6.    To path     /home/mbse/var/nodelist/
7.    Compile     Yes

Enter your choice > -

Select menu item (1..7), "-" for previous level

vince : mbsetup

# Using UPS semaphore's.

If you have a UPS and you are able to let your UPS software create semaphore's when power fail conditions occur then read on. The MBSE BBS task manager and a lot of utilities will act on two special semaphore's, they are:

1. `upsalarm`, this semaphore should be set when there is no mains power, but there is enough power left to operate your system. All background tasks will be suspended as long as this condition is true. If the power comes back, the UPS software should remove this semaphore.

2. `upsdown`, this semaphore should be set when the UPS sofware signals your system to go down. This is a fatal condition and there is no way back. Even if the power comes back your system should shut down and the UPS will disconnect the power to your system. After a while it will turn the power on again and your system boots. MBSE BBS will, if this semaphore is seen, kick users out of the bbs, and the system shut down script will try to close MBSE BBS as quick as possible. Normal the close time out is one hour to let users normal finish what they were doing, now it is only 30 seconds and if they did not log out, they will be disconnected anyway.

I know not all UPS software can do this but most UPS software is open source so you can change it to create these semaphore's. It is not a problem that UPS semaphore's still exist if the systems boots, the MBSE BBS startup scripts will remove them before the bbs is started. Recommended is apcupsd at least for APC's range of UPS's and compatables.

# Running DOS doors.

**Introduction.**
Running DOS doors is possible on systems that can run [dosemu](). The latest versions of dosemu don't need to be started suid root or started using sudo, it can run as the user that is logged in the bbs. Since that is possible, a new location for the virtual C: drive is now being used within the $MBSE_ROOT file structure. This virtual C: drive is owned by the group bbs. The basic idea in how doors should be run written by Redy Rodriguez are still valid, but some scripts are changed and some dosemu files must be copied into the mbse directory tree.
*One very important note: the virtual dos drive is now $MBSE_ROOT/var/dosemu/c Don't use another location or change the access rights, there will be several references from mbse to that and only that location for dos support. Choose another one and you are on your own!*

**Installing dosemu.**
Some distributions have good dosemu packages, others don't. You need a dosemu that can be used in a system wide multiuser setup. The bad packages want to have the dos drive in each users home directory, this is not what we want for doors. If you don't have a good dosemu you have to compile your own. This is not too complicated. First download the source from dosemu. You don't need the freedos-bin source if you are going to use ms-dos or pc-dos. Unpack the source in some directory of your choice and compile the sources:

```
mbse@seaport$ tar xfvz dosemu-1.2.2.tgz
mbse@seaport$ cd dosemu-1.2.2
mbse@seaport:~/dosemu-1.2.2$ ./configure --prefix=/usr --sysconfdir=/etc/dosemu --with-x
mbse@seaport:~/dosemu-1.2.2$ make
Note that the --with-x is not needed for running doors, but it might be handy that
you can use dosemu as user mbse in X to do maintenance on your doors.
Then as root do:
root@seaport:~/dosemu-1.2.2# make install
```

**Setup dosemu in mbse.**
First copy `/etc/dosemu/dosemu.conf` into `~/etc/dosmeu`. Then edit `~/etc/dosemu/dosemu.conf` so that we have a version for mbse users. Set the following settings in that file:

```
$_hdimage = "/opt/mbse/var/dosemu/c"
$_floppy_a = ""
$_xms = (1024)
$_ems = (2048)
$_dpmi = (0x1000)
$_layout = "us"
```

Next copy this new file as `virtual.conf` in `~/etc/dosemu/`. Then add the following line:
$_com1 = "virtual"
Note that in older dosemu versions you could use a command line option to do this, but this isn't supported anymore since dosemu-1.2.x. The `rundoor.sh` now also uses two configuration files.
Next we need to setup the C: drive. Do this as user mbse. First type `umask 007`, this sets the create mode for directories to 0770 and files to 0660 so that we create all files and directories read/write for all bbs users. This is very important! Type the following commands:
```
mbse@seaport:~$ umask 007
mbse@seaport:~$ cd var/dosemu/c
mbse@seaport:~/var/dosemu/c$ mkdir dos dosemu temp util doors
```

Now we must install dos. If you use ms-dos then in `~/var/dosemu/c` copy the ms-dos files `command.com`, `io.sys` and `msdos.sys`. In `~/var/dosemu/c/dos` copy all ms-dos commands. Next install the dosemu utilities, copy these from `/usr/share/dosemu/freedos/dosemu` to `~/var/dosemu/c/dosemu`. Now there is a empty `~/var/dosemu/c/util`, copy some utilities here such as your favourite dos editor, zip and unzip etc. Next we must create `config.sys` and `autoexec.bat`. These files must be dos crlf formatted. You can do that with the joe editor, for example `joe -crlf config.sys`.
**Examples:**

```
config.sys:
BUFFERS = 20
DEVICE=C:\DOS\SETVER.EXE
SET TEMP=C:\TEMP
DOS=HIGH,UMB
FILES=50


autoexec.bat:
@ECHO OFF
PROMPT $P$G
PATH C:\DOSEMU;C:\DOS;C:\UTIL;
SET TEMP=C:\TEMP
Lh C:\DOS\DOSKEY /INSERT
Lh C:\DOS\SHARE
```

Now we are ready to try it, type `mbsedos` and the dos emulator should start. You can leave dosemu with the command `exitemu`.

**Installing a door.**

All doors are started using the script `~/bin/rundoor.sh`. This script is never started directly, you should make a copy of that with the name of the door, but the best solution is to make a symlink with the name of the door. For example:

```
mbse@seaport$ cd ~/bin
mbse@seaport:~bin$ ln -s rundoor.sh 8ball
```

See also inside `rundoor.sh` for the instructions. This file does several things, first it prepares the users home directory with the dosemu environment so that dosemu will run for the user. Then it creates a node directory in the dos C: drive and copies the door dropfiles into that node directory. Finally it starts dosemu in virtual comport mode and inserts the commands in dos to start the door.

But before we can run the door, the door itself must be installed in the dos partition. In this example I will explain how to install the door 8ball. Start `mbsedos` and create a directory `c:\doors\8ball`. Unpack 8ball in that directory and run `setup`. Test the door with the command `local`. Now goto the directory `c:\doors` and create the file `8ball.bat`. That file will be used to start the door. It will be called by `c:\doors.bat` with two parameters, the name of the door and the node number.

```
@ECHO OFF
C:
C:\UTIL\X00 E B,0,57600
CD \DOORS\8BALL
ASKME.EXE /E /C1 /PC:\DOORS\NODE%1
C:\UTIL\X00
CD \DOORS\NODE%1
DEL DOOR.SYS
DEL DOOR32.SYS
```

Finally we have to make a menu entry to start the door. The Opt. Data line is the command to start the door, the optional `/N` is replaced by the current node number of the user. A second option may be used, `novirtual`, if that is present the `rundoor.sh` will start dosemu without setting the comport in virtual mode. Use this if your door isn't a real door but a normal dos program.

```
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 17:03:28 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS   Sat Oct 17 16:03:28 2015 UTC    Free


     8.3. EDIT MENU ITEM

     1.   Sel. key   S
     2.   Type nr.   7          Run external program in shell
     3.   Opt. data  /usr/local/bin/safe-door /usr/local/etc/safe/safe-config
                      12345678901234567890123456789012345678901234567890123456789012345678901234
     4.   Display            ^[^S^]^    Safe Cracker
     5.   Security    0      -------------------------------
     6.   Min. age    0                    12. No door.sys No
     7.   Lo-colors   Normal display color 13. Use COMport No
     8.   Hi-colors   Bright display color 14. Run nosuid  Yes
     9.   Autoexec    No                   15. No Prompt   Yes
     10.  Door Name   Safe Door            16. Single User No
     11.  Y2K style   Yes                  17. Hidden door No


          Enter your choice > -

Select menu item (1..17), "-" for previous level


             vince : mbsetup
```

Note that I use the Display lines to make the menu, there are only ansi screens for the top and common lines on the screen, see the second image how to build the whole doors menu.

```
MBsetup for MBSE BBS version 1.0.6          Sat Oct 17 17:04:19 2015 TZUTC 0100
Copyright (C) 1997-2015 M. Broek      INS   Sat Oct 17 16:04:19 2015 UTC    Free


        8.3 EDIT MENU "DOORS" (ENGLISH)

        1.  a Display A?? file w/Ctrl codes     0 door
        2.  M Run external program in shell     0 /usr/local/bin/nu-door /usr/loca
        3.  S Run external program in shell     0 /usr/local/bin/safe-door /usr/lo
        4.  D Run external program in shell     0 /opt/mbse/bin/darkland /N
        5.  8 Run external program in shell     0 /opt/mbse/bin/8ball /N
        6.  - Return from gosub                 0
        7.  * Return to top menu                0
        8.  ! Gosub another menu                0 logoff.mnu
        9.  ^ Page sysop for a chat             0 Calling sysop please wait ...
        10. a Display A?? file w/Ctrl codes     0 include


        Enter your choice > -

Rec. (1..11), "-" Back, Append, Delete, Move, Previous, Next


             vince : mbsetup
```

**Netmail support.**

In the virtual c: drive you need some directories that point to several other mbse subdirectories. You may need at least three of them, one to put netmail in, one to put files in to send and one semaphore directory to tell mbse there is something to send. For example like this:

```
ln -s /opt/mbse/var/msgs /opt/mbse/var/dosemu/c/msgs
ln -s /opt/mbse/var/boxes/node92_100_60 /opt/mbse/var/dosemu/c/outbox
ln -s /opt/mbse/var/sema /opt/mbse/var/dosemu/c/sema
```

In dos you now have:
```
C:\MSGS
C:\OUTBOX
C:\SEMA
```

In C:\MSGS doors may put *.msg netmails. File attaches won't work from these netmails because the netmails point to files with dosnames. (May be solved later). The script that calls the door, for example c:\doors\bre.bat

must copy the file to send to the node outbound box. In the above example that would be copy to `C:\OUTBOX`. To create the semaphore to tell mbse to scan add to the batfile (for example `C:\DOORS\BRE.BAT` the following two lines:

```
CD C:\SEMA
ECHO "">MAILOUT
```

See also the F.A.Q. for some very useful tips and tricks. Using the same method with symlinks, you can make all paths you need from the rest of mbse visible in dos. Use with care.

**What if I don't upgrade.**
That's your choice, but for a while the old method starting dosemu via sudo is still supported. The file `~/bin/bbsdoor.sh` is still there, but the file `~/bin/rundoor.sh` is renamed to `~/bin/runvirtual.sh`. So you should change all symlinks of your door names in `~/bin` (or copies) to `~/bin/runvirtual.sh`.
You should not need to change anything else.

**Problems.**
Currently there are some troubles with modem calls.

# Error return codes.

Following is a list of return codes from all official mbse programs and utilities. These codes can be used in scripts. If a program exits normally, the return code is 0. If a program exits by a signal, the return code is the signal number. This means if you kill a running program with a SIGTERM, the programs exits with the SIGTERM error number. On most platforms this is 15. These error codes are for the most part the same on all *nix platforms but because of some small differences I cannot list these here. Just do a **kill -l** to see a list of errors for your own platform.
Here is the list of other return codes:

| | | |
|---|---|---|
| 100. | Command line error |
| 101. | Configuration error |
| 102. | Initialisation error |
| 103. | A full disk partition |
| 104. | UPS alarm detected |
| 105. | No valid recipients |
| 106. | Execute external program failed |
| 107. | Set tty failed |
| 108. | File transfer error |
| 109. | File attach failed |
| 110. | Can't lock program, retry later |
| 111. | Node not in nodelist |
| 112. | Node may not be called |
| 113. | Can't make connection |
| 114. | Can't open tty port |
| 115. | Node is locked |
| 116. | Node IP address not found |
| 117. | Unknown session type |
| 118. | Not Zone Mail Hour |
| 119. | Modem error |
| 120. | No modem port available |
| 121. | Session error (password) |
| 122. | EMSI session error |
| 123. | FTSC session error |
| 124. | WaZoo session error |
| 125. | YooHoo session error |
| 126. | Outbound scan error |
| 127. | Can't make poll |
| 128. | File request error |
| 129. | Error processing nodediff |
| 130. | Virus found |
| 131. | General error |
| 132. | Timeout error |

The mailer can also return the following error codes:

| | | |
|---|---|---|
| 201. | TTY general error |
| 202. | TTY timeout |
| 203. | TTY EndOfFile |
| 204. | TTY Hangup |
| 205. | TTY Empty |

When an external program is run, a mbse program can exit with return code 256 + the return code of the external program. So if the external program exits with error code 7, the code returned by the mbse program is 263.

# BBS Programs.

1. mbaff, Announce newfiles and filefind
2. mball, Allfiles and newfiles list creator
3. mbcico, The Fidonet mailer ala ifcico
4. mbdiff, Nodelist difference processor
5. mbfido, Fidonet mail and files procesor
6. mbfile, Files database maintenance program
7. mbindex, Nodelist index compiler
8. mblang, Language datafile compiler
9. mblogin, Unix login replacement
10. mbmon, The monitor program
11. mbmsg, The messagebase utility program
12. mbnntp, The news server
13. mbout, The mailer outbound program
14. mbpasswd, The passwd wrapper
15. mbsebbs, The bbs program
16. mbseq, Sequence number creator
17. mbsetup, The setup program
18. mbstat, The bbs status change program
19. mbtoberep, The toberep.data lister
20. mbuser, The userbase maintenance program
21. mbuseradd, The adduser wrapper

# mbaff - Announce new files and FileFind processor.

**Synopsis.**
`mbaff [command] <options>`


**Description.**
**mbaff** is the new files report generator and file find server for mbsebbs. In order to run **mbaff** you must first start **mbsed**, this is the deamon which controls all bbs activities.

When **mbaff** is run with the command line command **announce** the first thing it does is to scan all the file databases for files from which the announced flag is not yet set, and that area has a valid new files group name. These files are uploads for example. If such a file is found the announced flag is set and the file is added to the **toberep.data** file. This file may already contain new files who were received as .tic files and processed by the **mbfido** program. After this is done the **toberep.data** file is compared against the new files reports to see if there is anything to report. If that's the case the creation of reports begins in the echomail areas specified. After that the **toberep.data** file is erased and the mailout semaphore set.

The files to announce are divided into groups, the names of the groups are set in the file download areas. If you plan this well, you can make separate announcements for several networks, announce files, big groups of file, i.e., HAM or .jpg pictures, GNU/Linux etc.

When **mbaff** is run with the command line command **filefind** it will search each echomail area for unreceived messages addressed to **allfix** or **filefind**. It will read the message header and mark the message as received. The search options are set on the subject line. All file areas for which the filefind flag is set to true will be searched for the requested search patterns. If there are files found a reply will be generated for the user who wrote the request. If the reply area is different from the scan area, the reply is placed in the reply area. If it's not set, the reply goes into the same area. If the netmail option is set, the reply will be sent by netmail. To prevent echomail overflow the replies in the same area are limited to 15 found files, replies in the other echomail area are limited to 50 files. Netmail replies will contain up to 100 files.


**Environment.**
In order to run **mbaff** you need to set one global environment variable **$MBSE_ROOT**. This variable must point to the root of the bbs directoy structure. The main configuration file **config.data** must be present in the ~/etc subdirectory.


**Commands.**
`mbaff announce` - Announce new files.
`mbaff filefind` - Process filefind requests.


**OPTIONS**
`mbaff [command] -quiet` - Quiet mode, no screen output. Use this switch if you run **mbaff** from the crontab.

# mball - Allfiles listing Generator.

**Synopsis.**
**mball** [commands] <options>

**Description.**
**mball** is the allfiles and new files listing generator that can be made available on your bbs for your users to get a complete listing of wat is available for download. When used with the **-zip** option it can also produce complex versions of these two listings. The resulting files are created in the current directory. After the creation of these files you can hatch them into your bbs with the programs **mbfido tic** when you properly setup a .tic file area for this purpose and create records for the hatch manager.

Just before the reports are created, and if you defined WWW and FTP log files with **mbsetup** in menu 1.13, the log files are analysed and and files downloaded from the bbs are counted as valid downloads. To make sure downloads are only counted once, a status file is created in ~/var, this marks the date the last time the log files are checked. For the best results make sure you run **mball** before any WWW and FTP log files are rotated, else you might miss some downloads.

**Environment.**
In order to run **mball** you need to set the global variable **$MBSE_ROOT**. This variable must point to the root of the bbs directory structure. The main configuration file **config.data** must be present in the ~/etc directory.

**Commands.**
**mball list** - Create allfiles.txt and newfiles.txt files.

**Options.**
**mball [command] -quiet** - Quiet mode, suppress screen output.
**mball list -zip** - Create zipped listings as well.

**Setup.**
In **mbsetup** menu 1.13 you need to set the days to include in new files listings and the maximum security level. Also define valid paths to log files that are created by your web and ftp server.

# mbcico - The Fidonet mailer.

## Synopsis.
```
-a<inetaddr>
-l<ttydevice> <node> …
-n<phone>       forced phone number
-l<ttydevice> forced tty device
-t<tcpmode>    telnet TCP/IP mode, must be one of ifc|itn|ibn, forces TCP/IP
-a<inetaddr>   supply internet hostname if not in nodelist
<node>         should be in domain form, e.g. f11.n22.z3
-h             show this help message
 or: mbcico tsync|yoohoo|**EMSI_INQC816|-t ibn|-t ifc|-t itn (this is answer mode)
```

## Description.
**mbcico** stands for MBse "Internet - Fidonet Copy In /Copy Out", this is a FidoNet compatible transport agent. It is based on **ifcico** written by Eugene G. Crosser, <crosser@average.org>, 2:5020/230@FidoNet. I changed the name of the program to make the difference between **ifcico** and **mbcico**. Nowadays it is quite different then ifcico. Currently it supports FTS-0001, YooHoo/2U2 and EMSI handshake protocols, Xmodem, Telink, Modem7, Hydra with zlib compression extension (FSP-xxxx), SEAlink with and without overdrive and crash recovery, Bark file and update requests, WaZoo protocols: DietIFNA, plain Zmodem (aka ZedZip, EMSI flag "ZMO") and ZedZap, WaZoo file and update requests (nodelist flag should be XA). WaZoo file and update requests do also work with FTS-0001 sessions, this is supported by several well known DOS mailers also. Password protected requests and update requests are implemented.

There is also a special protocol optimized to use over TCP/IP connections, contributed by Stanislav Voronyi <stas@uanet.kharkov.ua>, it is identified by EMSI proto code TCP (not registered) and nodelist flag IFC. The default port is 60179. A telnet variant is installed at port 60177, the nodelist flag is ITN:60177. The port number is needed because the default port in the nodelist is port 23.

There is also a **Binkp/1.1** implementation, this is a bi-directional TCP/IP protocol. This protocol is preferred over the IFC protocol because it is more efficient. Nodelist flag is IBN, the default port is 24554, and the nodelist request flag is XX. This binkp implementation uses zlib packet compression opt PLZ (FSP-1032) to increase the transfer speed and to lower the network bandwidth usage. There is also support for the stream compression modes GZ and BZ2 (compatible with binkd).

mbcico can use both IPv4 and IPv6 TCP/IP connections.

Outbound directory structure is BinkleyTerm compatible, with domains and point subdirectories (full 5d). There are separate "protected" and "unprotected" inbound directories for the incoming sessions with the nodes that are in your setup. Files received during outbound sessions are always stored in the "protected" inbound.

"Magic" file request processors are executable files placed in the "magic" directory. If a request is made for a file with matching name, the executable from the "magic" directory is run, and its stdout output is mailed to the requester. Full requester's address, in the form of "John Smith of 1:234/56/7" is passed to the executable in the command line. An example of such file is on my system, the file name in the magic directory is **STATUS**.

```
echo "    Hello $1 $2,"
echo ""
echo "              Status report for MBSE BBS Development"
echo "              ------------------------------------"
echo ""
echo "Date  : `date`"
echo "System: `uname -mrs`"
echo "Uptime: `uptime`"
echo ""
echo "`df -T`"
echo ""
echo "`free`"
echo ""
echo "Gtx, Michiel Broek"
```

For example - If you file request STATUS from my system you will get something like:

```
    Hello John Doe,

                Status report for MBSE BBS Development
                --------------------------------------

Date  : Sat Nov  8 17:29:07 CET 2003
System: Linux 2.4.20 i586
Uptime:  17:29:07 up 88 days, 20:02,  1 user, load average: 0.00, 0.00, 0.00


file system    Type    1k-blocks      Used Available Use% Mounted on
/dev/hda2      ext3     5921096    3405184   2210276  61% /
/dev/hdb1      ext3     6198404    5133056    750476  88% /opt


            total        used        free      shared     buffers      cached
Mem:        94280       91360        2920           0       13152       46276
-/+ buffers/cache:      31932       62348
Swap:      136512       32380      104132


Gtx, Michiel Broek
```

Non-executable files in the magic directory contain the full names to magic file names. The magic NODELIST can thus point to the full path and file name of your latest nodelist. These magic names are automatic maintained by the **mbfido** program when the magic name is set in the .tic file that you receive.

To run **mbcico** in master mode, you need to make dialout devices read/writeable for **mbcico**, and do the same for the directory where your uucp locks are created (usually /var/locks).


**Answer Mode.**

To make **mbcico** work in answer mode, you need **mgetty** written by Gert Doering. **mbcico** must be started with one of the following parameters:

```
FTS-0001 call:         "/opt/mbse/bin/mbcico tsync"
FTS-0006 call:         "/opt/mbse/bin/mbcico yoohoo"
EMSI call:             "/opt/mbse/bin/mbcico **EMSI_....."
```

In the latter case the received EMSI packet should be passed whitout trailing CR). To make this work **mgetty** must be compiled with the -DFIDO option. Other getty programs might work.

To answer TCP/IP calls the following lines should be added to /etc/inetd.conf:

```
    binkd   stream  tcp    nowait  mbse   /opt/mbse/bin/mbcico   mbcico -t ibn
    fido    stream  tcp    nowait  mbse   /opt/mbse/bin/mbcico   mbcico -t ifc
    tfido   stream  tcp6   nowait  mbse   /opt/mbse/bin/mbcico   mbcico -t itn
```

The tfido line is configured to answer on IPv4 and IPv6. If your system uses xinetd the file /etc/xinetd.d/mbsebbs could be:

```
#:MBSE BBS services are defined here.

service binkp
{
        socket_type    = stream
        protocol       = tcp
        wait           = no
        user           = mbse
        instances      = 10
        server         = /opt/mbse/bin/mbcico
        server_args    = -t ibn
}

service tfido
{
        socket_type    = stream
        protocol       = tcp
        wait           = no
        user           = mbse
        instances      = 10
```

```
        server          = /opt/mbse/bin/mbcico
        server_args     = -t itn
        flags           = IPv6
}

service fido
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = mbse
        instances       = 10
        server          = /opt/mbse/bin/mbcico
        server_args     = -t ifc
}
```

If you want to use IPv6, add the line flags = IPv6 to the protocol like in the example of tfido. In the file /etc/services the following lines must be present:

```
binkd           24554/tcp               # mbcico IBN mode
fido            60179/tcp               # mbcico IFC mode
tfido           60177/tcp               # mbcico ITN mode
```

**Calling Mode.**
You never need to call nodes with **mbcico** by hand, **mbtask** will start **mbcico** with the right command line.
**Note:** you should not call nodes with mbcico directly, let **mbtask** do the calling. If you want to call a node make a poll command.

**Environment.**
In order to run the mbcico you need to set one global environment variable **$MBSE_ROOT** This variable must point to the root of the bbs directoy structure.

**Return Codes.**
0       - No errors
1..32   - OS errors, SIGHUP, SIGKILL, etc.
100     - Command line error.
101     - Configuration error.
103     - Disk full.
108     - File transfer error.
111     - Node not in nodelist.
112     - Node may not be called (Hold, Down, not ZMH).
113     - Could not make connection.
114     - Cannot open tty port.
115     - Node is locked.
116     - No IP address.
117     - Unknown session type.
118     - Not Zone Mail Hour.
119     - Modem error.
120     - Not port available.
121     - Session error.
122     - EMSI session error.
123     - FTSC session error.
124     - Wazoo session error.
125     - YooHoo session error.
126     - Outbound scan error.
127     - Cannot poll.

These codes are also stored in status files for nodes, with the extension of ".sts". These are small datafiles containing three decimal numbers.

1. Time retry code, this is the next call attempt time. This is an unsigned long representing the number of seconds since the epoch. Before this time the node may not be called. This is set after a failed call, a random time in the near future is selected.
2. Retries, this is the number of consecutive call attempts made that returned "call failed" or other errors. This field is zeroed when the call succeeds and when a new "poll" is created. If the value is 30, the node won't be called any more.
3. Code, this is the return code of the last attempt.

**Configuration.**

The behaviour of mbcico can be configured with **mbsetup**, section 1.14 If something doesn't do what you want, set the debug on for that problem. This will produce huge logfiles, but also a lot of information. Important flags are Device IO, EMSI debug, File forward, Locking, Outbound scan and Session.

**Bugs.**

Incoming calls from McMail mailers, McMail is quite hasty to establish an EMSI session, and times out in less than a second. So if your system is heavy loaded, or not too fast, McMail cannot connect with your system. This is a known problem with McMail 1.0 and older, later versions are ok.

# mbdiff - Nodelist difference file processor.

**Synopsis.**
**mbdiff** [nodelist] [nodediff] <options>

**Description.**
**mbdiff** applies a (compressed) nodediff file against the nodelist of the week before to create a new nodelist. The result is a new plain nodelist and a nodelist compressed with zip.

**Environmet.**
In order to run **mbdiff** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

**Commands.**
**mbdiff** [nodelist] [nodediff] The nodelist must be the full path and file name without the dot and day number extension. The nodediff is the full path and file name to the (compressed) nodediff file fitting on the latest nodelist. It is adviced to make a separate working directory where you keep the nodelist's. Don't do this in your normal nodelist directory. When the operation is successful, the new nodelist is in the working directory and the old list is removed. A compressed version of the nodelist is also placed in the working directory. From here you can hatch the new compressed nodelist with the **mbfido** program.

**Options.**
**-quiet** - suppress screen output, this switch is needed when **mbdiff** runs on the background.

# mbfido, the fidonet mail and files processor.

**Synopsis.**
```
mbfido [command(s)] <options>
```

**Description.**
**mbfido** is the program to process fidonet mail and files. In order to run mbfido you must first start **mbtask**, this is the deamon which controls all bbs activities. To prevent that **mbfido** will run more than once at the same time a lock is placed in the protected inbound with the pid of the running **mbfido** program. The gateway to and from internet is also handled by **mbfido**.

**Specifications.**
The recognized mail packets are type 2+ following the FSC-0039 standard with a fall back to the old stone age packets. Can handle messages of maximum 429467295 bytes, or less if you have less memory available, the practical limit is about 1 Meg. Note that most mail processors are only guaranteed to work to maximum 16 KBytes. Experiments between 1997 - 2003 in the LINUX echo show that this is still true, although most tossers seem to process mail up to 32 KBytes.

**Tossing Mail.**
First make sure you have the necessary message areas in your setup. At least you need the badmail and dupemail areas and a netmail area for each network you participate in. If you don't create badmail and dupemail areas then bad (unknown area etc) and dupes are lost and you cannot check the reason why. If you don't create the netmail areas for each network, then netmail to your system will disappear. It is not possible to "retoss" the badmail yet after you have created any missing echomail areas.

To prevent .pkt name collision the toss of incoming mail is done in parts. The first run is to process all uncompressed mail packets and add mail to the outbound. Then each compressed ARCmail archive will be uncompressed and processed and mail will be imported and forwarded as necessary. During all these passes all file names are sorted by date and time, the oldest files are processed first.

The recognized mail packets are type 2+ following the FSC-0039 standard with a fallback to the old stone age packets. The packets are checked for being addressed to one of your own aka's and for a correct password. The password check may be switched off in the nodes setup. After all the packet header checks the messages will be extracted from the packet file.

When messages are extracted from the packets, the date format is checked for Year2000 bugs from other tossers. Several checks are done by ideas of Tobias Ernst of 2:2476/418. It is also possible to run the **pktdate** utility before each packet will be processed. Whatever date format us used in the original message, **mbfido** will always rewrite the date field in the right FTS-0001 format.

If the message is a netmail the message is checked for DOMAIN, INTL, FMPT and TOPT kludges so that full 4d or 5d addressing will be possible. Then a check is done if this netmail is addressed to one of our aka's. If it's addressed to "sysop" or "postmaster" the name is replaced with the sysop's name. If the message is addressed to one of the names defined in the service setup, that mail will be handled by the service manager, ie. given to areamgr, filemgr or send further as email to your local system.

Then the message is checked if it is addressed to an existing bbs user, and if so it will be imported into the netmail area of the main zone of the bbs. If it's not addressed to a bbs user, the message will be readdressed to the sysop. If the message is not for one of our aka's the message will be put in the mail queue for further routing.

If the message is a echo mail message it will be checked for being a duplicate by storing the CRC32 value of the AREA: line, message subject, origin line, message date and msgid kludge and testing if that CRC32 value exists in the echo mail duplicate database. If there is no msgid in the message, the whole message body will be include to complete the CRC32 dupe check. Also the existence of the echomail area is checked and the node must be linked to that area. If the message is not in a passthru area and is not a duplicate it is finally imported in the message base. After that is the message will be forwarded to downlinks by adding the message to the mail queue.

**Adding mail and files to the outbound.**
Adding mail and file to the outbound is done in two passes. The first pass is to put all outgoing mail into the ~/var/queue/z.n.n.p directory, the last letters are replaced by the digits of the node number. The extension can be

qqq for packed mail, nnn for normal unpacked mail, hhh for hold unpacked mail and ccc for crash unpacked mail. Adding mail and files to this directory can always be done, even if one of the nodes you are adding mail for is having a mail session with your system. This is a safe operation.

In the second pass, these temp files are really added to the outbound, but only if the destination node is not locked, ie. there is no current mail session with that node. If there is a mail session going, these temp files will stay in the temp directory and are added to the outbound in a later run of **mbfido**. If adding the mail to the outbound succeeds the temporary files and directory is removed.

**Alias file.**

If the file /opt/mbse/etc/aliases exists, mbfido will try to fetch ftn-style aliases from there. If "from" address of a message from FidoNet matches **right** side of some entry in alias file, then the Reply-To: header is created in the RFC message with the name part taken from the left side of the sysalis entry and domain part taken from myfqdn (below). E.g., if a fidonet message comes from "John Smith of 1:234/567.89@fidonet" and there is an entry in the sysalias file:

`"jsmith:    John.Smith@p89.f567.n234.z1.fidonet.org"`

and Domain name value is "mbse.nl", then the resulting message will contain a line: "Reply-To: jsmith@mbse.nl".

**Commands.**

**mbfido** `areas` This command will check all file and mail groups if areas files are defined and if the setting **Auto change** is set to Yes. Then the areafile is read and the areas in that file are compared with the defined areas. Missing areas are created and areas not in the areafile are removed or blocked depending if there are files or messages present in these areas. This is also a good command to create large bulks of new areas on your system.

**mbfido** `mail <recipient>` This command is used by your MTA to deliver email addressed to for example Michiel_Broek@f2802.n280.z2.fidonet.org

**mbmail** `<recipient>` This is the same as above.

**mbfido** `notify <nodes>` This command will send notify messages to all nodes in your set up which have the notify option set to on. If you enter nodes as option you may use wild cards, ie 2:*/* to send messages to all nodes in zone 2. If you specify exactly one node, messages will be generated even if that node has the notify function off. From cron you should not specify any nodes, this will just send to all your links the information about their setup. Each node will receive a status report for files and mail, and area list for all file areas and mail areas to each aka a node has, and a flow report for mail for each aka.

**mbfido** `roll` This command will only do something if a new week or month has begun. If this is true the statistic records in several databases are updated. You should run this command each midnight from cron to be sure that this when it is time to do so. This command is always executed before one of the scan, toss or tic commands so if you don't do this in cron, it will still happen.

**mbfido** `scan` Scan for new messages entered at the bbs or by other utilities. If the file ~/tmp/echomail.jam or ~/tmp/netmail.jam exists, mbfido will only scan the messages in areas which are pointed at in this file. This is a lot faster then a full mailscan. If it is not present, all messagebases are scanned to see if there is a new message. If you specify **-full** on the command line a full messagebase scan is forced. It is wise to do this sometimes, on my bbs I run this once a day.

**mbfido** `tag` The command will create tag- and areas files in the doc directory for each group of mail and files.

**mbfido** `test <node>` This is for testing of the mail routing. The node on the command line must be in the format f28.n280.z2 etc. The results are printed on the tty. If you have debug logging on in menu 1.5.16 items 17 and 18, then all needed debug information is written to the logfile. You can use this to debug your routing.

**mbfido** `tic` Process incoming files accompanied with .tic control files. Several actions can take place on the incoming file before they are imported in the BBS areas. Options are re-archiving, replacing banners (with your add), check for DOS viruses, running scripts for certain file name patterns, send these files to other nodes etc. All options can be defined for each area. If as a result from one of the actions there are new files hatched, for example after processing a nodelist difference file which created a new nodelist, the .tic processing will start again, until there is really nothing more to do.

**mbfido** `toss` Toss incoming fidonet netmail and or echomail. By default mail in the protected inbound directory will be processed, uncompressed .pkt files and compressed arcmail bundles are recognized, file name case doesn't matter.

**mbfido** `news` Scan all defined newsgroups for new news articles. New articles are fetched from the news server

and stored in your message base and send to your up- and downlinks. This is for use with an NNTP gateway.

**mbfido** `uucp` This will read a standard a newsbatch from stdin and gate the articles to Fidonet and the local message base. This is for use with an UUCP gateway, this mode should be called by uuxqt. The newsbatch may be compressed or uncompressed or a single news article.

**mbnews** This is an alternative to **mbfido uucp -quiet**.

**Options.**

**mbfido** `[command] -nocrc` Disable CRC checking of incoming TIC files. Only use this if you know what you are doing.

**mbfido** `scan -full` Force scanning of all message bases for new entered mail. You need this if mail in entered with other editors then from mbse. Also, running it once a day is advised to catch any missed messages.

**mbfido** `news -learn` Scan the news server for news articles, and update the news dupes database only. Use this switch if you start using mbfido to gate news articles for the first time. Articles will not be gated with this switch, mbfido will just learn which articles already exist. Normally you only need to use this switch once.

**mbfido** `[command] -nodupe` Disable checking for duplicates. Normally you should not use this switch. This switch doesn't work with the news command, only for echomail and tic files.

**mbfido** `[command] -quiet` Quiet mode, all output to the current tty is suppressed. Use this flag if you toss mail from a script (started by cron) after mail is received.

**mbfido** `toss -unsecure` Toss mail without checking if the echomail is for your own system and without checking if the sending node is connected to your system. Nodes who are excluded from a certain echo, can still not enter messages in that echo.

**mbfido** `[command] -unprotect` Toss from the unprotected inbound directory. The default is to toss from the protected inbound directory.

**Environment.**

In order to run the bbs you need to set one global environment variable **$MBSE_ROOT** This variable must point to the root of the bbs directoy structure.

# mbfile - File database maintenance program.

**Synopsys.**
**mbfile** `[commands] <options>`

**Description.**
**mbfile** is the file database maintenance program for mbsebbs. In order to run mbfile you must have started **mbsed**, this is the deamon which controls all bbs activities.
The main purpose of **mbfile** to do automatic maintenance on the downloadable files on the bbs, such as removing or moving old files, checking the database and packing the database. The best way to do the maintenance is to run **mbfile** from the crontab. example:
```
30 05 * * * export MBSE_ROOT=/opt/mbse; /opt/mbse/bin/mbfile kill pack index -quiet
```
Other functions in **mbfile** are meant to do manual maintenance such as adding, deleting or moving files.

**Environment.**
In order to run the bbs you need to set one global environment variable **$MBSE_ROOT** This variable must point to the root of the bbs directoy structure. The main configuration file **config.data** must exist in the subdirectory ~/etc.

**Commands.**
**mbfile** `adopt <area> <file>` "[description]" Adopt a file to the file base. The description is optional, but if present it should be enclosed in quotes. The optional description may be at most 255 characters. Default mbfile will try to use the FILE_ID.DIZ if this is present. The file to adopt needs to be in your current directory and will be copied to the new area. If you have virus scanners defined, the file will be checked for viruses.
**mbfile** `check [area]` Check the database integrity. All files in the file database must exist on disk and all files on disk must exist in the file data base. There are some exceptions, files.bbs, files.bak, 00index, index*.html, header, readme and files that start with a dot. Of all files the date and time is checked, the size and the crc value of the file. If there is something wrong, the error is corrected or the file is removed. If the area is a CD-rom area, the check that files on disk must exist in the file database is skipped. If no area number is given, all areas are checked.
**mbfile** `delete <area>` "<filemask>" Mark file(s) in an area for later deletion. You can use wild cards, and if you do surround them with quotes to prevent shell expansion. The file is not removed until you run **mbfile pack**.
**mbfile** `import <area>` Import files to the given area using files.bbs descriptions. There are two possibilities, import from CD-rom or just from a temporary directory.
   • Import from a temporary directory: Put the files to import there including the files.bbs file.
   • Import from CD-rom: Go into the directory on the CD where the files are. If the files.bbs is not in the same directory, you must enter the full file name and path where the files.bbs is with **mbsetup** 8.4.area.6.
To start the import, type `mbfile import area` where area is the number of the destination file area. The files are not checked for FILE_ID.DIZ inside the archives. If you have virus scanners defined, the files are checked for viruses before they are added the the bbs. If you don't want these fresh files to be announced, add the `-announce` option on the command line. This may be a good idea if you migrate from another bbs system.
**mbfile** `index` Create fast file request index for the **mbcico** file request processor. In each directory with the downloadable files is also a **files.bbs** file written in msdos cr/lf format. This includes a download counter. Also, if defined, in each available download area for ftp/http, **00index** and **index.html** files are created, starting at the ftp pub directory. If the area contains gif or jpg images, thumbnails are created using the **convert** program if you have that installed. The thumbnails have the same file names as the original images, only they start with a dot. The convert command is defined with **mbsetup** menu 1.19.8, if you don't have it or don't want thumbnails, make that entry empty. The index command only rebuilds area index files (files.bbs, index.html and 00index) when there was a change in the area.
**mbfile** `kill` Mark files to delete or move files in areas that have the **download age** set or the **filedate age** set. A setting of 0 is ignored. Areas on CD-rom are always skipped. If the Move to Area option is set the files are moved to the given area. The upload date and download date are reset to the current date and time. So if you set in the destination area aging of 14 days, files will stay there for 14 days after the move. This is good for automatic "last chance" areas.
**mbfile** `list [area]` List all defined file areas, the number of files, the total size of the files and the primary group. If and area is given on the command line, the files in that area are listed.

**mbfile** move <from> <to> <file> This command will move a file from one area to another. No actions on the file are performed.

**mbfile** pack This command will actualy remove the records of files that are marked for deletion. If the file is still on disk, it will be removed also. So when you delete files with mbsetup or with mbfile, they are still in your database and on disk until you run **mbfile pack**.

**mbfile** rearc <area> "<filemask>"; Re-archive file(s) in an file area to the archiver that must be set in the file area setup.

**mbfile** sort <area> Sort the files in area on the long file names. The sort is case insensitive. When the sort is completed, the command **mbfile index** will automatic run to rebuild all indexes.

**mbfile** toberep Shows which files are to be reported. This lists only the files that are received during .tic processing, new uploaded files are not displayed. This list is automatic removed after new files are announced. See also mbtoberep.

**mbfile** undelete <area> "<file mask>" Mark a file that is marked for deletion to be available again, ie. undo the deletion mark. You can use wild cards, and if you do surround them with quotes to prevent shell expansion.

**Options.**

**mbfile** [command] -quiet Quiet mode, no screen output. Use this switch if you run **mbfile** from the crontab.

**mbfile** [command] -announce Announce mode, use this switch with the adopt and import commands to prevent that the added file(s) will be announced. This can be useful if you add lots of files that you don't want to announce these as new files.

**mbfile** [command] -force With the adopt, import and move commands files that already exist in the destination area are not overwritten. With this switch you can force to overwrite the existing file.

**mbfile** [command] -virus Supress virus scanning of files during adopt or import. Use with care.

# mbindex - Nodelist Index Compiler.

**Synopsis.**
**mbindex** `<options>`


**Description.**
**mbindex** is the nodelist index compiler. It will create an index file containing the sorted fidonet addresses as index file to the raw nodelists in the defined nodelist directory. Several other programs use this index file for fast retrieval of data from the nodelists. Compiling new nodelist indexes can always be done, while compiling the result is stored in temporary index files and only after successful compilation the original indexes are renamed and the temporary files get the normal names. The renamed (old) indexes stay on disk including the previous version of the old raw nodelist. They stay there in case some program had the nodelist or index still open. So in the nodelist directory there are current and previous nodelists, current and previous indexes, and during compiling the temporary indexes. There is no need to manually remove (and not wise to do so) files from the nodelist directory.

The nodelists in the nodelist directory are the normal uncompressed nodelists in MS-DOS format (with CR/LF). The file name extensions must be two or 3 digits. So if you have a private pointlist named **bestbbs.pts** you will have to rename that to **bestbbs.999** to make it work.


**Environment.**
In order to run **mbindex** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.


**Options.**
**mbindex** `-quiet` Quiet mode, no screen output. Use the switch if you run **mbindex** from a shell script or from the crontab.

# mblang - Language Data Compiler.

**Synopsis.**
**mblang** [language data file] [language source text]


**Description.**
**mblang** compiles the source text file to language datafile which is used by the **mbsebbs** program. You only need to use this program if you install a new language file. When you build the complete mbse bbs package, this command is run automatic for you.

# mblogin - Unix login replacement for MBSE BBS.

**Synopsis.**
**mblogin** [-p] [-h hostname] [user]


**Description.**
**mblogin** replaces the standard Unix login program for MBSE BBS. You can use this program from **mgetty** for modem/ISDN access, or from **inetd** to allow telnet access. This program logs in the systems syslog facility because the normal login program does this too. It uses the file /opt/mbse/etc/login.defs for the behaviour you prefer. All options in this file are well commented. It uses the bbs user database to see who are allowed to login. Users can use their Fidonet name, Unix name or Handle. If the name is found in the user base, the name is replaced with the real Unix name to do the password check and to start the **mbsebbs** program. If the user is not found in the user base, the user is asked if he wants to register as a new user. This behaviour can be turned off. If the user uses the name **bbs** he will start the **mbnewusr** program directly. All other users are not allowed to login, not even root. If you change it in the file /opt/mbse/etc/login.defs you may allow user **mbse** to login. I advice against it, you should use **ssh** if you want remote access to do maintenance.


**How to use from inetd.**
If your system is connected to the internet you may want to let users login using telnet. The changes you need to make are different for each operating system mbse supports. The right lines for your distribution should already be present, but they are not set active, in other words, you still need to make changes to allow telnet login to work. First I will describe how to use it from inetd. When you make changes to your system to change the telnet login, make sure you are already logged into your system as root from another terminal. If you make a mistake and can't login any more you will be glad that you are still logged in on another terminal. Here are the tested set ups:

- GNU/Linux: edit the telnet line in /etc/inetd.conf:
  ```
  telnet   stream   tcp   nowait   root   /usr/sbin/tcpd   in.telnetd   -L   /opt/mbse/bin/mblogin
  ```
  After changing this file do a **kill -HUP pid** where pid is the pid of the inetd process.
- FreeBSD: edit the telnet line in /etc/inetd.conf:
  ```
  telnet   stream   tcp   nowait   root   /usr/libexec/telnetd   telnetd   -p   /opt/mbse/bin/mblogin
  ```
  After changing this file do a **kill -HUP pid** where pid is the pid of the inetd process.
- NetBSD: append a line in /etc/gettytab just below the line with the default entry:
  ```
  mbsebbs:cd:ck:np:lo=/opt/mbse/bin/mblogin:sp#38400:
  ```
  The speed entry 38400 doesn't seem to be important. Then edit the telnet line in /etc/inetd.conf:
  ```
  telnetd stream tcp nowait root /usr/libexec/telnetd telnetd -g mbsebbs
  ```
  After changing this file do a **kill -HUP pid** where pid is the pid of the inetd process.

Now you can test it with `telnet localhost` or from another machine with `telnet your.machine.com`.
Check if you can still do other logins such as ssh, rlogin and login on local consoles.


**How to use from xinetd.**
This is how to use xinetd, Read the section above also. Here is a tested set up: This has to be done as root. You have to create a service under /etc/xinetd.d named telnet if you dont have it yet installed, if it is already installed just modify the server_args line to match this:
```
server_args = -L /opt/mbse/bin/mblogin
```
Your /etc/xinetd.d/telnet file should look something like this:
```
service telnet

{
        protocol        = tcp
        instances       = 10
        flags           = IPv6
        log_on_failure += USERID
        socket_type     = stream
        user            = root
        server          = /usr/sbin/telnetd
        server_args     = -L /opt/mbse/bin/mblogin
        wait            = no
}
```
This will replace the standard login program on xinetd systems to the mblogin program, so that next time someone

logs into your bbs they will be presented the issue file on /opt/mbse/etc file and the login prompt for the BBS instead of the standard login prompt. Note that on some systems the server_args need -E instead of -L. If you don't have IPv6 enabled on your system, you must remove the flags = IPv6 line.

**How to use from mgetty**
This is described with the Setup for mgetty.

# mbmon - MBSE BBS Monitor.

**Sysnopsis.**
mbmon

**Description.**
**mbmon** is the monitor program so that you can see what is happening on your bbs. It can show all processes and actions of all programs, show system statistics, disk usage, and the last callers list. **mbmon** must run on the same system where the bbs is. When started it will also report that the sysop is present on the system for chat so that users of your bbs can page you for a chat. If a user pages you, this will be shown in every screen of **mbmon** and a beep will sound every few seconds.
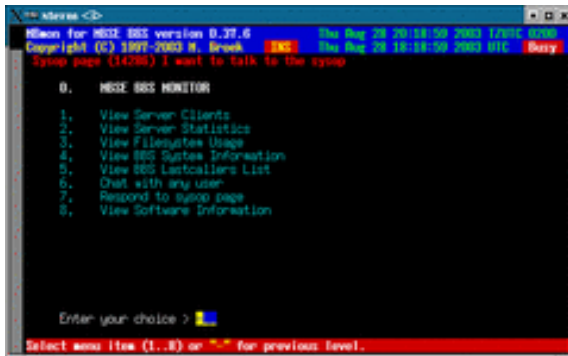

Main menu, a user is paging.
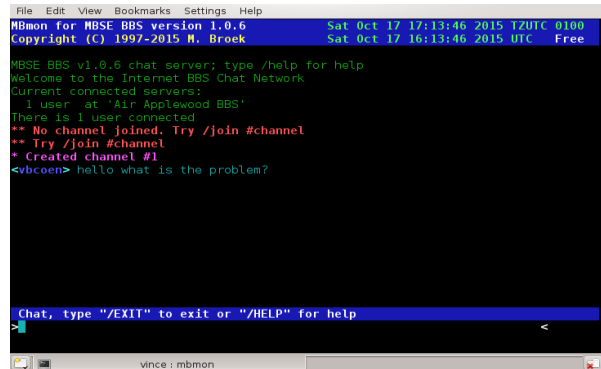

Chat with a user.

**Environment.**
In order to run **mbmon** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure.

Menu 1. - This screen shows the running bbs programs and what they are doing.

Menu 2. - This screen displays the actual mbtask server status and some internal variables. The left side of the screen are more statistics and counters, the right side displays the actual bbs open status, Zone Mail Hour status, if Internet is alive, if internet is needed, if mbtask is internal running and the system load average. Consult this screen if you think mbse doesn't run.

Menu 3. - This screen displays the file system usage. Removable file systems such as CD's are not displayed, only file systems that the bbs could use.

Menu 4. - This screen shows some bbs information.

Menu 5. - This menus shows the last callers of the bbs. It shows the user name, location, user level, the device they logged in from, the start time, the duration, the number of calls, the line speed and the actions they have done. The action fields mean: **H**idden, **D**ownload, **U**pload, **R**ead messages, **P**ost messages, **C**hat, **O**ffline reader, **E**xternal door.

Menu 6. - Chat with any user. Using this menu you enter the chat server and you can chat in any channel just as the users of your bbs can do. The chat server works just like an IRC server. Type /HELP for help.

Menu 7. - Respond to a sysop page. This is actually the same as chat with any user, the only difference is that you will join channel #sysop immediately. If the user is not yet connected after his page, he will also be connected to the channel as soon as possible.

Menu 8. - This shows the software information.

# mbmsg - Message Base Utility.

**Sysnopsis.**
**mbmsg** `[commands] <options>`

**Description.**
**mbmsg** is the message base utility program for mbsebbs. In order to run mbmsg you must have started **mbsed**, this is the deamon which controls all bbs activities.

The main purpose of **mbmsg** is to link messages after tossing mail, and to maintain the size of the message bases and the age of the messages. The best way to do the maintenance is to run **mbmsg** from the crontab. example:

`30 05 * * * export MBSE_ROOT=/bbs; /bbs/bin/mbmsg kill pack link -quiet`

Another purpose is to automatic post messages in message areas for.Echomail and netmail.

**Environment.**
In order to run **mbmsg** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

**Commands.**
**mbmsg** `link` Link all messages by subject ignoring Re: in the subject lines. You should run this after tossing or scanning mail.

**mbmsg** `kill` Kill messages in areas that have the **age** set or the **maximum** messages set. A setting of 0 is ignored. The messages are not removed from the message base, they are only marked as deleted.

**mbmsg** `pack` This command actually removes the messages who have the deleted flag set. The last read pointers are updated and the messages renumbered. After this command there is no way you can recover your messages, except from backups.

**mbmsg** `post <to> <#> <subj> <file> <>` This command posts a message in numbered area. If a field consists of more then one word it must be surrounded with quotes. The **to** field can be "Michiel Broek" for a full name or "Michiel_Broek@f16.n2801.z2.fidonet" for netmail addressing. Look out: you need underscore between the first name and last name, no spaces.  can be one or more of the characters "c", "i", "h" or "p" to set the Crash, Immediate, Hold or Private flags. If no  is needed, use the - (minus sign) as a place holder.

**Options.**
**mbmsg** `[command] -area <#>` Process only one area <#> number.

**mbmsg** `[command] -quiet` Quiet mode, no screen output. Use this switch if you run **mbmsg** from the crontab.

# mbnntp - News server.

**Synopsis.**
`mbnntp` You should *not* run this from the command line, instead it should be run by **inetd** or **xinetd**.

**Description.**
**mbnntp** is a RFC977 and RFC2980 compatible news server. The purpose is to make echomail available to news reader clients so that bbs users can use their favourite news reader to participate in echomail. It will not make your news groups available if you have any, the bbs user can get these directly from the internet. The bbs echomail areas are only made available if they have a valid newsgroup name set in the area set up. If this is not alread set, invent your own newsgroup name. For example the WWB echomail area MBSE could be set as fido.wwb.mbse, but you can use anything as long as it doesn't conflict with real news groups.
To use the server, the user must login in the news server with his bbs unix name and password, thus the user must exist on the bbs. The access to the areas is the same as if the user would be logged into the bbs.

**Installation.**
If installed from **inetd** add the following line to `/etc/inetd.conf`:
```
nntp    stream tcp    nowait mbse   /opt/mbse/bin/mbnntp    mbnntp
```

**Building.**
If you did run `./configure --enable-newsgate` when you compiled the mbse package, **mbnntp** will be build as a not working program. This is intended to prevent loops between real news and echomail. With normal builds without `--enable-newsgate` (recommended) **mbnntp** will function as news server.

# mbout - The Outbound Manager.

**Synopsis.**
**mbout** [command] <params> <options>

**Description.**
**mbout** is the outbound manager for MBSE BBS. It can ask information from the nodelists, create and remove polls, request and send files and display the outbound status. Most of the tasks such as create and remove polls should be done from the crontab.

**Environment.**
In order to run **mbout** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

**Commands.**
**mbout** att <node> <> <file> will attach the specified file to the specified node. The node should be in the format f2802.n280.z2, should be crash, immediate, normal or hold. Only the first letter of the parameter is needed. If the node is not in the nodelist, the status is Down or Hold, then this command fails. To non-CM nodes you must use the **Immediate** flavour if you want to send the file direct. The flavours Hold and Normal are still allowed. The file must be in the directory range from where file attaches are allowed.

**mbout** poll [node..node] creates poll requests in the outbound for one or more nodes. The node should be in the format f2802.n280.z2. The semaphore **scanout** is created so that the mailer will start calling. The mailer will handle the poll request as if it should deliver immediate mail, so the node will be called as long as the poll request exists, even to nodes which are not CM. The error counter for the node to poll will be reset to zero, so a node that was previous marked un-dialable will be called again. If a call to a node is successful, the poll file will be removed by **mbcico**. If a node is not in the nodelist or has the status Down or Hold, no poll will be created for that node.

**mbout** stop [node..node] removes poll requests that are leftover when polling nodes didn't succeed. There is no check if the node is in the nodelist or has the status Down or Hold, the poll is always removed.

**mbout** req <node> <file> [file..file] creates file requests to a node. One or more file names may be given including wild cards. It is not possible to do update or password protected uploads yet. If there is already a requestlist for that node, the new requests will be added. This command does not call a node, you need to create a poll request to make the actual call. This is also practical if you want some files from your uplink, just make the requests and the actual request is send when your normal scheduled poll to your uplink is processed.

**mbout** reset [node..node] resets the nodes "try" counter to zero. So if a node is undiable, the counter is 30 or more, then by resetting this counter that node will be called again. The outbound is rescanned after this command so that calling will start immediately.

**mbout** stat shows the status of the mailer outbound. This status is also written to the logfile.

**mbout** node <node> will show the nodelist information for a certain node.

**Options.**
**mbout** [commands] -quiet will suppress screen output. This is useful if you run **mbout** from the crontab or from background scripts.

**Examples.**
This is an example of crontab entries that writes the outbound status to the logfile and creates and stops polling of 2 nodes.

```
00 00 * * * export MBSE_ROOT=/opt/mbse; $MBSE_ROOT/bin/mbout stat -quiet
00 01 * * * export MBSE_ROOT=/opt/mbse; $MBSE_ROOT/bin/mbout poll f98.n100.z92 f0.n100.z92 -quiet
00 02 * * * export MBSE_ROOT=/opt/mbse; $MBSE_ROOT/bin/mbout stop f98.n100.z92 f0.n100.z92 -quiet
```

# mbpasswd - The password wrapper.

**Description.**

**mbpasswd** is the wrapper for the **passwd** program is present on all GNU/Linux systems. To use **passwd** to change the password of another user is only allowed by **root**. The mbpasswd program overcomes this limitation. The wrapper **mbpasswd** is run from the bbs by user **mbse** when a new user is being registered to set his password. It can also be started from the bbs when an existing user changes his password. If you as sysop use **mbsetup** to change a users password it will be used too. His password under Unix is then always the same as his password in the bbs program. This is necessary for the user to be able to get his email using the pop3 protocol.

You never need to run **mbpasswd** by hand, in fact it is protected so that it can only be started by the bbs or mbsetup.

**Environment.**

**mbpasswd** must be installed setuid root and setgid root, ls -la looks like this:

```
-rws--s--x   1 root     root           6644 Jun 26 21:23 /opt/mbse/bin/mbpasswd*
```

**Commands.**

Not mentioned here because **mbpasswd** is only called by other programs, it can not be run manually.

# mbsebbs - The main BBS program.

**Synopsis.**
**mbsebbs**

**Description.**
**mbsebbs** is the main bbs program for the users. To be able to use the bbs, the bbs must be set open with **mbstat open** or users may not login. This is normally done at system boot. Also Zone Mail Hour is honoured if the user logs in on a tty that has the **honor ZMH** flag set to true. If it is ZMH and the ZMH flag is true, the user will be kicked out of the bbs. You should only set this flag on the modem lines where you need to be available for ZMH if your "Fidonet" network requires that.

The first visible action is to show the logo.ans file. Because it is not known who is trying to login, you only need to place this file in the default language directory.

The next check is to see if the user is allowed to login on the tty he currently is on. If this tty is not available or is not in your set up, the user is kicked out. If he is allowed to login, a message is shown at which port he is on, unless you have turned this feature off in the setup.

If the user login is successful, his favourite language will be loaded. Then it is checked if the user is the Sysop, if so, the Sysop flag is set. If the users Date of Birth is invalid, he is forced to enter the right Date of Birth. The next check is to see if the user has passed the expiry date, this is a useful feature for systems with donating users. Finally the access limits are set for the user and time remaining for today and download limits are set.

After successfull login the user can be presented with a bunch of advertising screens. I will only name the screens without file name extension, as these screens are searched for in the following order;

1. file name.ans in the users favourite language directory.
2. file name.ans in the default language directory.
3. file name.asc in the users favourite language directory.
4. file name.asc in the default language directory.
5. If nothing is found, nothing is displayed.

Here is the list of displayed screens:

1. **mainlogo**. Here you can put a logo or something.
2. **welcome**. This screen can contain information about the session the user has, his download limits, time left etc.
3. **welcome1** is shown if the user has show bulletins set to true.
4. **welcome2** is shown if the user has show bulletins set to true.
5. **welcome3** is shown if the user has show bulletins set to true.
6. **welcome4** is shown if the user has show bulletins set to true.
7. **welcome5** is shown if the user has show bulletins set to true.
8. **welcome6** is shown if the user has show bulletins set to true.
9. **welcome7** is shown if the user has show bulletins set to true.
10. **welcome8** is shown if the user has show bulletins set to true.
11. **welcome9** is shown if the user has show bulletins set to true.
12. **birthday** is shown if the user logs in at his birthday and if he has show bullentins set to true.
13. **dd-mm** is shown if **dd** is the date of today and **mm** is the current month and if the user has show bulletins set to true.
14. **sec20** is shown if the user has the security level in the file name, level 20 in this example. Als the display bulletins must be set to true.
15. **news** is shown if the user has the display bulletins set to true.
16. **onceonly** is shown only if the user has never seen this screen, the test is to compare the users last login date against the date of this file.

After all these screens the users Off line Reader areas are checked to see if you as sysop didn't change the message areas. If you made changes, the users set up is adjusted and he will be notified. This means he sees a list with deleted areas and new areas.

Next if the user has newmail scan set to true all message bases are checked to see if the user has new mail. This includes a check to see if he has Unix mail. If there is Unix mail, those mails are retrieved from the POP3 server

and stored in his private mailbox. If there is new mail for the user, the user sees a list of areas with the messages in it and he is offered to read and reply these messages.

Then if the user has 'show newfiles' set to true and he is not a new bbs user he will see a list of new files you have on your bbs. During this display he can tag files for later download.

The final setup is to set the users "do not disturb" flag and then the menu system is started. The menu system will run forever, until the user chooses to logoff, the connection is lost or his daily time limit is reached. For the possibilities and set up of the menus see MBSE BBS Menu System

**Environment.**

In order to run **mbsebbs** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

If the environment variable **CONNECT** is present, a log entry will be made with the connect speed.

If the environment variable **CALLER_ID** is present, a log entry will be made with the caller id.

The environment variable **LOGNAME** must contain the unix user name. This is set by the **mblogin** program.

If the environment variable **REMOTEHOST** is present, a log entry will be made with the name or IP address of the remote host.

If the environment variable **TERM** is present, a log entry will be made with the terminal type of the remote user.

# mbseq - Sequence number creator.

**Synopsis.**
**mbseq**

**Description.**
**mbseq** writes a eight character hexadecimal unique sequence number to the stdout. This number is received from **mbsed** which keeps track of the generated sequence numbers. This written number can be used in shell scripts to create unique file names for Fidonet .pkt files, for example:
cp temp.pkt `mbseq`.pkt

# mbsetup - The Setup Program.

**Synopsis.**
`mbsetup [init]`

**Description.**
**mbsetup** is the set up program for MBSE BBS. It should be run only by user **mbse** and it fails to start if you are not. If it is started with the optional parameter **init** only the databases are initialized. This is automatically done by **mbtask** when you start the system for the very first time. You should never need to use this switch. When **mbsetup** is started without arguments the databases will be checked and initialized before the main screen is displayed. Both methods are being used to make sure all needed databases exist. For a detailed description of all setup options see BBS Setup Guide

**mbsetup** uses locking to protect the system databases. Some setup menus can only be entered if the bbs is free, ie. no users logged on, no mailer sessions, not tossing mail etc. If the bbs is free, then these menus can be entered and the bbs will be closed. No users can login, tossers do not run etc. Only mailer calls are still accepted. So be careful not to stay too long in these menus, you are blocking normal bbs use. In the top status bar this situation is displayed.

**Environment.**
In order to run **mbsetup** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory. If it is not present, ie you run **mbsetup** for the first time, a default **config.data** will be created. This will also happen with several other databases.

# mbstat - MBSE BBS Status Changer.

**Synopsis.**
**mbstat** [commands] <options>

**Description.**
**mbstat** changes the bbs status between open and close, can wait for all users to log off and wait for critical utilities to stop their actions.

**Environment.**
In order to run **mbstat** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

**Commands.**
**mbstat** semaphore scanout will set the internal semaphore *scanout* in the **mbtask** daemon.
The following semaphore's are valid: scanout, mailout, mailin, mbindex, reqindex, msglink.
**mbstat** close will close the bbs for users. Users that are just logging in to the bbs will be thrown out after a short message. Users already logged in will be thrown out when they pass by a menu prompt. So users who are doing file transfers can finish their transfers before being disconnected.
**mbstat** open opens the bbs for users. This should be run from one of the system start up scripts right after you started **mbsed**. If you installed everything as it should this command is already executed at system startup.
**mbstat** wait will wait for the bbs to become free. This includes a check for utilities that do critical actions so they can finish their job without corrupting the bbs databases. The default is to wait 60 minutes. If the semaphore **upsdown** exists it will wait only 30 seconds.
You should run **mbstat close wait** in your system shutdown script so that the system shutdown will wait for a clean shut down of the bbs before the rest of your system goes down. If you installed everything as it should be then these commands are already installed in your system shut down scripts.

**Options.**
**mbstat** [command] -quiet will suppress screen output. This is good for using mbstat in scripts.

# mbtask - MBSE BBS Taskmanager.

**Sysopsis.**
`mbtask` [-nd]


**Description.**
**mbtask** is the task manager for the whole MBSE BBS system. This deamon keeps track of all client actions, does the logging for the clients, does database locking, authorizes clients, set/resets users "do not disturb flags", sends and receives chat messages, keeps track of Zone Mail Hour, the status of the mail and files in the outbound, and the BBS open/close status. Communication between **mbsed** and the client programs is done via Unix Datagram sockets. The protocol used to communicate between **mbtask** and the clients is explained later. This daemon also watches the semaphore directory for some special files. It also starts programs when they are needed. The very first time **mbtask** is started it creates a default config.data and task.data, the main configuration and task configuration files. Then it calls **mbsetup init** to build the default databases. **mbtask** should be started at system boot so the bbs system will start working. The init script that is installed on your system will do that.

After start up and initialization **mbtask** runs internally once per second forever. If there is nothing to do then this time will slowly increase up to 5 seconds. This time will be reset to one second as soon as there is work to be done. The actual work is to check a number of external and internal semaphore's and act on these. But before any program is started a number of things are checked:

1. Check the system's load average. If it is too busy the processing of background tasks is suspended until your system load drops. The default setup is set at 1.50 but you can change that with mbsetup. Experience will learn what the best value will be and I need some feedback on that.
2. The UPS semaphore **upsalarm** will be checked. This means that the system is running on battery power and no new jobs are started.
3. The UPS semaphore **upsdown** will be checked. This is the fatal one, if this one exists **mbtask** will try to stop all current running jobs. If there are no jobs left running then **mbtask** will stop itself. The upsdown semaphore means that the system will shut down and power off, that's why it's fatal and there is no way back.
4. The status of the bbs will be checked, is it open or closed. If it is closed, no jobs will be started.
5. The Zone Mail Hour is checked. If ZMH begins the semaphore's **zmh** is created. If ZMH ends the semaphore **zmh** is removed.
6. Each twenty seconds a ping is send to the IP addresses defined with **mbsetup** to check if the internet can be reached. If both ping addresses fail, it is assumed that the internet can't be reached. After a status change, the outbound will be scanned.
7. Scan the mailer outbound for work. This builds a list of nodes with mail in the outbound and sets the necessary flags on nodes who may be called. If a node needs to be called, **mbtask** will spawn **mbcico** to call this node. The number of free modem and ISDN ports and the maximum number of TCP/IP sessions and already registered sessions, determine how many sessions will be started. The sessions will be started at intervals of 20 seconds. It will also set a time when something will change for a node, ie. a zone mail hour is reached, or a mail window for a node with Txx flags is reached. Internally this scheduler runs at the UTC clock because Fidonet has all times defined in UTC.

Each new minute the timestamp of semaphore **mbtask.last** is updated so that you can check that **mbtask** is running. Also each minute is checked if the system configuration files are changed, is so they are reloaded. There is no need to stop and start **mbtask** if you made changes to the system configuration. Then all kind of internal semaphore's will be checked. The commands that are executed have default values, but they can be changed wit mbsetup. The commands can be scripts as well. The checks and actions are:

| semaphore | Speed | Tasktype | Depends on | Job to run |
|-----------|-------|----------|-----------------|------------------------------------|
| mailout   | Fast  | mbfido   | Max. 1 mbfido task | mbfido scan web -quiet |
| mailin    | Fast  | mbfido   | Max. 1 mbfido task | mbfido tic toss web -quiet |
| newnews   | Fast  | mbfido   | Max. 1 mbfido task | mbfido news web -quiet |
| mbindex   | Fast  | mbindex  | No other tasks  | mbindex -quiet and if exist: goldnode |
| msglink   | Fast  | mbfido   | No other tasks  | mbmsg link -quiet |
| reqindex  | Fast  | mbfile   | No other tasks  | mbfile index -quiet |
| scanout   | Slow  | call     | Only 1 call task | mbcico -r1 |

The Fast and Slow values mean: Fast is each second, Slow is check each 20 seconds. As you can see, the system will not do too much at the same time. Jobs like compiling new nodelists or create file request indexes have a very low priority. Because this daemon checks the semaphore's each second it responds much better that the old scripts running on the cron daemon. The system will be expanded so that more outgoing calls will be done at the same time, ie. ISDN and analogue calls, and if they are present internet calls, will be made at the same time.

The **mbtask** program keeps also track of a unique number generator, this is just a simple counter that is increased each time it is asked for a new number. It will take years for the numbers to repeat. Even if the status file is lost the chance that numbers are repeated on your system are almost zero. The first time the counter is initialized it is set to the current unix time in seconds since 1 January 1970. This counter is used by several programs to create unique .pkt file names, msgid numbers etc.

The command line option **-nd** is only for debugging, it allows to start without becoming a daemon, **mbtask** will run in the foreground. This option is only useful for developers.

### Environment.

In order to run **mbtask** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure.

### Security.

**mbtask** is installed setuid root. This is needed to initialize a raw socket for the ping function. After that is done the privilege drops to user **mbse** before the child process is created and the rest of the initialisation is done. The child process can never get root privileges because it is spawned by user mbse.

### Communications.

Communication between the server and the clients is established by Unix datagram sockets. There can be only 1 server running. The server will accept connections from clients on your local machine only. The limit for the amount of clients that can connect to the server is set to 100.

The server creates a Unix datagram socket at startup and waits for connections. The name of this this socket is /opt/mbse/tmp/mbtask. When a client connects it creates a Unix datagram socket in /opt/mbse/tmp, the name is the name of the program, added with the pid of the program. So if **mbcico** is started with pid 2312 the socket will be /opt/mbse/tmp/mbcico2312.

All commands are 4 capital letters followed by a colon, a number indicating how much data fields will follow. If that number is higher than zero, the data fields are separated with commas. The command is terminated with a ; character. Examples are:

```
GCLO:0;               Zero datafields command.
DOPE:1,dbname;        One datafield command.
```

All commands will receive a reply as soon as possible. If a resource is temporary not available, a reply will follow too, telling this condition. Replies can also contain optional data. Examples:

```
100:0;                Response 100, no data.
200:1,Syntax error;   One datafield.
```

Because data fields in commands and replies are separated and terminated by , and ; characters these characters need to be escaped. Also characters outside the us-ascii range must be escaped. This is done by translating the character to a backslash followed by the hexadecimal number of the character. The server has a 10 minute time out for receiving data when a connection is established. The clients need to "ping" the server at regular intervals to prevent a disconnect. All official MBSE BBS programs do that. The pid send with most commands is the pid of the calling program. Since this number is unique, it is used to keep track of the connected clients.

The commands are divided in 26 categories, most unused at this time.

### Categories:

```
Cat.    Description
----    -----------------------------------------
Axxx    Accounting, system monitor info etc.
Cxxx    Chatting
Dxxx    Disk watch
Gxxx    Global commands.
Mxxx    MIB commands.
Sxxx    Status commands.
```

**Group A, Accounting.**

Command: AINI:5,pid,tty,uid,prg,city;   Initialize connection, and who am I.
Reply:   100:1,line;              Ok, linen is a logical linennumber for the bbs.
     200:1,Syntax Error;         Error.


Command: ADOI:2,pid,doing;           What am I doing right now.
Reply:   100:0;                  Ok.
     200:1,Syntax Error;        Error.


Command: ACLO:1,pid;               Close my connection.
Reply:   107:0;                 Connection closed.
     200:1,Syntax Error;        Error, connection is still open.


Command: ALOG:5,fil,prg,pid,grade,txt;  Write a line of text in logfile with grade.
Reply:   100:0;                  Ok.
     201:1,errno;              Error, number in errno.


Command: ATCP:1,pid;                Registrate this session as TCP/IP session.
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.


Command: AUSR:3,pid,uid,city;        Set user name and city
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.


Command: ADIS:2,pid,flag;            Set Do Not Disturb flag.
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.


Command: ATIM:1,time;               Set new Client/Server timer in seconds.
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.


Command: ADEF:0;                  Set Client/Server timer to default (10 minutes).
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.


Command: ATTY:2,pid,tty;            Set new tty name.
Reply:   100:0;                 Ok.
     200:1,Syntax Error;        Error.



**Group C, Chatting**

Command: CIPM:1,pid;                Is Personal Message present.
Reply:   100:2,fromname,message;       Yes, from .. with message text.
     100:0;                  No.


Command: CSPM:3,fromuser,touser,txt;    Send personal message to user.
Reply:   100:1,n;                n: 0=Ok, 1=Do not disturb, 2=Buffer full, 3=Error.
     100:0;                 Impossible.

Command: CSYS:2,pid,1;        Sysop available for chat (from mbmon).
     CSYS:2,pid,0;        Sysop goes away (from mbmon).
Reply:  100:0;        Always Ok.

Command: CPAG:2,pid,reason;     Page sysop for a chat.
Reply:  100:1,n;        1=busy, 2=sysop not available, 3=error.
     100:0;        Ok

Command: CCAN:1,pid;        Cancel sysop page.
Reply:  100:0;        Always Ok.

Command: CCKP:0;        Check sysop page (from mbmon).
Reply:  100:3,pid,1,reason;     Page is active.
     100:3,pid,0,reason;     Page is cancelled, user still online.
     100:0;        No page active.

Command: CISC:1,pid;        Check sysop in chatmode.
Reply:  100:1,1;        Yes and drop into chatmode.
     100:1,0;        No.

Command: CCON:4,pid,user name,unixname,n; Connect to chatserver with user name. n=1 user is sysop.
Reply:  100:1,error;        Error with message.
     100:0;        Ok.

Command: CCLO:1,pid;        Close chat session.
Reply:  100:1,error;        Error.
     100:0;        Ok.

Command: CPUT:2,pid,message;     Put message on chatserver.
Reply:  100:2,0,error;        Error, not fatal and continue.
     100:2,1,error;        Error, fatal and disconnect chat.
     100:0;        Ok.

Command: CGET:1,pid;        Get message from chatserver.
     100:2,0,message;        If message present.
     100:2,1,error;        Error and disconnect chat.
     100:0;        No new message.


**Group D, Disk watch command.**

Command: DRES:0;        Reset and reread disk tables.
Reply:  100:0;        Always Ok.

Command: DSPC:0;        Enough free diskspace.
Reply:  100:1;n;        0=No, 1=Yes, 2=Unknown, 3=Error.

Command: DGFS:0;        Get file system status (see note below).
     100:n,data1, ..., data10;     Maximum 10 file systems datalines.


**Group G, Global commands.**

Command: GNOP:0;        No Operations.
Reply:  100:0;        Ok.

Command: GPNG:1,data;                Ping, echo data.
Reply:   100:1,data;          Ping reply.

Command: GVER:0;                 Give server version.
Reply:   100:1,Version ....;     Version reply.

Command: GSTA:0;               Get complete mbsed status record. (13 fields)
Reply:   100:19,start,laststart,daily,startups,clients,tot_clients,tot_peak,syntax_errs,
       com_errs,today_clients,today_peak,today_syntax,today_comerr,bbsopen,
       is_zmh,do_inet,processing,system_load,sequence;

Command: GMON:1,n;            Get registration info line, 1=First, 0=Next line.
Reply:   100:7,pid,tty,user,program,city,isdoing,starttime;
      100:0;            No more lines.

Command: GSYS:0;             Get bbs statistics.
     100:7,calls,pots_calls,isdn_calls,network_calls,local_calls,startdate,lastcaller;

Command: GLCC:0;             Get Lastcallers count
     100,1,n;           Return counter value.

Command: GLCR:1,recno;        Get Lastcaller record
     100:9,user,location,level,tty,time,minsmcalls,speed,cations;
     201:1,16;          Not available.


**Group M, MIB commands.**

Command: MSMS:6,kbrcvd,kbsent,direction,state,itype,freqs; Set Mailer Session
Reply:   100,0;

Command: MGMS:0;             Get Mailer Session
Reply:   100:12,kbrcvd,kbsent,sessin,sessout,sess_sec,sess_unseq,sess_bad,ftsc,yoohoo,emsi,binkp,freqs;

Command: MSTN:3,in,out,bad;      Set Tosser Netmail
Reply:   100,0;

Command: MGTN:0;             Get Tosser Netmail
Reply:   100:3,in,out,bad;

Command: MSTI:3,in,out,bad;       Set Tosser Internet-email
Reply:   100,0;

Command: MGTI:0;             Get Tosser Internet-email
Reply:   100:3,in,out,bad;

Command: MSTE:4,in,out,bad,dupe;    Set Tosser Echomail
Reply:   100:0;

Command: MGTE:0;             Get Tosser Echomail
Reply:   100:4,in,out,bad,dupe;

Command: MSTR:4,in,out,bad,dupe;    Set Tosser RFC-news
Reply:   100:0;

Command: MGTR:0;               Get Tosser RFC-news
Reply:   100:4,in,out,bad,dupe;

Command: MGTT:0;               Get Tosser Totals
Reply:   100:4,in,out,bad,dupe;

Command: MSFF:6,in,out,bad,dupe,magics,hatched; Set Tosser Files
Reply:   100:0;

Command: MGFF:0;               Get Tosser Files
Reply:   100:6,in,out,bad,dupe,magics,hatched;

Command:   MSBB:9,sessions,minutes,posted,uploads,kbupload,downloads,kbdownload,chats,chatminutes;   Set
BBS
Reply:   100:0;

Command: MGBB:0;               Get BBS
Reply:   100:9,sessions,minutes,posted,uploads,kbupload,downloads,kbdownload,chats,chatminutes

Command: MGOB:0;               Get Outbound Size
Reply:   100:1,size;


### Group S, Status commands.

Command: SBBS:0;               Get BBS Status (open, zmh, shutdown).
Reply:   100:2,0,The system is open for use;
         100:2,1,The system is closed right now!;
         100:2,2,The system is closed for Zone Mail Hour!;

Command: SOPE:0;               Open the BBS.
Reply:   100:0;            Ok.

Command: SCLO:1,mesage;        Close the BBS with reason.
Reply:   100:0;            Ok.

Command: SFRE:0;               Is the BBS Free.
Reply:   100:1,Running utilities: n  Active users: n;
         100:0;            It's free.

Command: SSEQ:0;               Get next unique sequence number.
Reply:   100:1,number;         Next unique sequence number.

Command: SEST:1,semaphore;     Get status of internal semaphore.
Reply:   100:1,n;              1 = set, 0 = not set.
         200:1,16;            semaphore not known.

Command: SECR:1,semaphore;     Set semaphore
Reply:   100:0;            Ok.
         200:1,16;            Error.

Command: SERM:1,semaphore;     Remove semaphore
Reply:   100:0;            Ok (also if there was no semaphore).
         200:1,16;            semaphore not known.

Note: in reply of DGFS the reply is 100:n,size free mount point fstype,.....
where n = 1 for 1 file system, and 10 for a total of 10 file systems.
There will never be a reply for more then 10 file systems. The reported file systems are collected by a thread process of mbtask that only includes the file systems actually used by mbse. This is used by the **mbmon** program to get a "live" view of your file systems. The GSPC command is used by utilities to check if enough space is available to continue to work safely.

# mbtoberep - List newfiles to Report.

**Synopsis.**
`mbtoberep`

**Description.**
**mbtoberep** is a small utility to list the file ~/etc/toberep.data which contains the newfiles found on your system before **mbaff announce** is run. This program is intended for system development but I decided to leave it in the distribution. If you pipe the output through more or less you are able to inspect the records.

**Environment.**
In order to run **mbtoberep** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory.

# mbuser - User Database Maintenance.

**Sysnopsis.**
**mbuser** `[commands] <options>`

**Description.**
**mbuser** is the user database maintenance program. It can delete users upto a certain level who have not called for a number of days. It can also pack the user database. This is not really a pack of the database, the deleted records are zeroed but the database is never shrunk. Every user once in this database will keep his record forever. This is to be sure that all LastRead Pointers will be correct. Records that are zeroed can be reused for new users. **mbuser** must run setuid root and setgid root because it executes /usr/sbin/userdel to delete the Unix account of the user that is removed from the bbs.

**Environment.**
In order to run **mbuser** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory. **mbuser** must be installed setuid root and setgid root, ls -la looks like this:
```
-rws--s--x  1 root     root        23560 Jun 19 19:50 /opt/mbse/bin/mbuser*
```

**Commands.**
**mbuser** `kill [n] [l]` will mark users to delete who have not called in **n** days up to and including level **l**.
**mbuser** `pack` will delete (zero) the users marked for deletion. You should also run this command if you marked users to delete with **mbsetup**.

**Options.**
**mbuser** `[command] -quiet` will suppress screen output, this is for running **mbuser** in the background or from the crontab.

# mbuseradd - The useradd wrapper.

**Sysnopsis.**
**mbuseradd** [gid] [user name] [comment] [userdir]

**Description.**
**mbuseradd** is the wrapper for the **useradd** program that should be present on most GNU/Linux systems. **useradd** may only be executed by **root** and there are some other minor things that need to be done as **root** to create a new Unix account that can be used with MBSE BBS. The solution for these problems is **mbuseradd**, this little program runs setuid root and setgid root. If it fails to do that it aborts. **mbuseradd** is called by **mbnewusr** and checks if it's called like that. If it is successful the user will have an entry in /etc/passwd, the comment is his Fidonet name, and his shell is $MBSE_ROOT/bin/mbsebbs.

If all this is successful until now, the home directory for this user is created and the right ownership and permissions are set. In his home directory the empty file **.hushlogin** is placed to prevent check for new mail when he logs into your system. This is the Unix mail check that is skipped and has nothing to do with the check for new mail in the bbs. All other directories that are needed for the bbs are created by **mbsebbs**.

**Environment.**
In order to run **mbuseradd** you must set the global variable **$MBSE_ROOT**. This variable must point to the root directory of the bbs structure. The main configuration file **config.data** must be present in the ~/etc directory. **mbuseradd** must be installed setuid root and setgid root, ls -la looks like this:

```
-rws--s--x  1 root    root        6644 Jun 26 21:23 /opt/mbse/bin/mbuseradd*
```

**Commands.**
Not mentioned here because **mbuseradd** is *only* called by mbnewusr. It cannot be run manually.

# Appendixes

# Appendix 1 - MBSE BBS FAQ and Howto.

**Introduction**

This FAQ is a work in progress. It is not complete, nor does any one claim that it is complete. This FAQ only answers questions pertaining to using MBSE BBS on your GNU/Linux box. It is a compilation of previous questions that Michiel Broek has received and answered from us newbies in getting MBSE up & running.

Caution many of the answers are related to older versions of mbse down to 0.33 so may not be correct for versions v1.0.n or later.

1. Installation
    1. Why do I have to install MBSE in the /opt directory?
    2. What other operating systems will MBSE run on?
    3. Will MBSE ever be ported to DOS/Windows or OS/2?
    4. What does the quest switch in the bbs user setup mean?
    5. How do I monitor as the things happen?
    6. How do I start mbsebbs from inetd or xinetd?
    7. I just installed mbse, now I get a Socket send failed error 2
2. Fidonet Setup
    1. The nodelist compiler crashes, why?
    2. What is the purpose of the "Comment" line in the tic area setup?
    3. How do I poll a node?
    4. I created a filerequest but mbcico doesn't call out.
    5. Does MBSE support fidonet via ftp feeds?
    6. The tosser and mbmsg program are slow, why?
    7. How can I use an external netmail tracker with MBSE?
    8. I am having problems with dates when unarchiving .arc files.
3. Internet Setup
    1. Posting newsmessages to the newsserver fails.
    2. FTP fidonet feeds how-to.
    3. Polling a internet node fails
    4. Dialup script examples for MBSE
    5. BBS users cannot get their email on Debian
4. BBS Setup
    1. How do I to set up MBSE for ISDN?
    2. How do I use GoldED together with MBSE?
    3. How do I use MsgEd together with MBSE?
    4. How do I adopt a directory full of files to the bbs?
5. Doors Setup
    1. How do I run DOS doors under MBSE BBS?
    2. Dosemu compile tips
    3. Setting up for InterBBS games.
    4. Door Maintenance.

**1. Installation.**

| Q 1.1 | Why do I have to install MBSE in the /opt directory? |
|---|---|
| A 1.1 | The /opt is for extra applications, when mbse is installed there it is easier to upgrade the OS. If you insist on installing in /usr/local the choose /usr/local/mbse. |

| Q 1.2 | What other operating systems will MBSE run on? |
|---|---|
| A 1.2 | Currently only on most GNU/Linux distributions, FreeBSD (& OSX) and NetBSD. The author, Michiel Broek, has made a valiant effort ensuring that MBSE will recognize and install on the major distributions of GNU/Linux. It also runs under IBM OS/390 & Z/OS in Linux mode. Could also work in Unix mode if all of the required packages are installed. |

| Q 1.3 | Will MBSE BBS ever be ported to DOS/Windows or OS/2? |
|---|---|
| A 1.3 | No. |

| Q 1.4 | What does the guest switch in BBS user setup mean? |
|---|---|
| A 1.4 | Nothing yet. |

| Q 1.5 | How do I monitor as the things happen? |
|---|---|
| A 1.5 | tail -f /opt/mbse/log/system.log or start mbmon. |

| Q 1.6 | How to start mbsebbs from inetd or xinetd? |
|---|---|
| A 1.6 | This is described in the documentation of the mblogin program. |

Q 1.7.   I just installed mbse, now I get a Socket send failed error 2.
A 1.7.   With a normal system boot the **mbtask** daemon will be started by one of the systems init scripts. If you just installed mbse you need to do this for the first time manually. As user **mbse** issue the following commands:
```
mbtask
mbstat open
```
This will start the daemon, the daemon creates the default databases.
Then you can use **mbsetup** to configure your new system.

## 2. Fidonet Setup.

Q 2.1.   The nodelist compiler crashes, why?
A 2.1.   Note: From previous experience, it is vital that you follow the docs closely. Also, ensure that you have a copy of a valid nodelist and nodediff file. I have ran into problems where either the nodediffs don't match the nodelist, the crc checks fail, or I have files from two different zones. I have even had to back up several months before finding a valid nodediff that would match the nodelist.

Q 2.2.   What is the purpose of the "Comment" line in the tic area setup?
A 2.2.   The downlinks will see this when receiving FileMgr replies. It is meant only as a description for the filearea.

Q 2.3.   How do I poll a node?
A 2.3.   mbout poll f2802.n280.z2

Q 2.4.   I created a file request but mbcico doesn't call out.
A 2..4.   You need to add a poll for that node to do the call.

Q 2.5.   Does MBSE support fidonet via ftp transfers?
A 2.5.   Yes, but only the passive side. Refer to 3.2

A 2.6.   The tosser and mbmsg programs are slow, why?
A 2.6.   Tossing mail and linking message areas are background tasks. When mbse was first developed the hardware was not fast enough to run these programs at full speed. The mailer could give CRC errors and the bbs seemed to stop now and then. To overcome this problem all programs started with the -quiet switch (background jobs) are stopping for short periods so that they would use less system resources. Systems with SCSI disks or other fast and well designed machines don't have a need for that slowdown. In mbsetup menu 1.5.17 is the setting for the speed of all utility programs, the default is slow. You might try this set to fast and see what happens on your system.

Q 2.7.   How can I use an external netmail tracker with MBSE?
A 2.7.   If the netmail tracker can work on .pkt files you could try to install it into mbfido. It can call a external program to process incoming .pkt files before mbfido does something with it. That option was built a few years ago for extra Y2K checks. (The pktdate program of Tobias Ernest).

Q 2.8.   I am having problems with dates when unarchiving .arc files.
A 2.8.   If you are experiencing a problem with the dates on arc files showing as January 1970 or similar when extracted using Arc v5.31 for linux then obtain "nomarch". MBSE will detect nomarch during configuration/install and set your archiver.data file accordingly. Nomarch extracts .arc files and keeps the original file date.

**3. Internet Setup.**

Q 3.1.   Posting news messages to the news server fails.
A 3.1.   Check if you need the "mode reader" command for the news server. Set this flag in mbsetup 1.15.5.


Q 3.2.   How do I configure MBSE for my ftp filenet feeds?
A 3.2.   1. Follow the next steps to setup your system for FTP clients.
1.   As root, add a new group to your system: **groupadd bbsftp**.
2.   Make sure there is a directory /opt/mbse/var/bbsftp. This directory should be owned by mbse, group bbs and have mode 0775.
3.   Add an invalid shell to /etc/shells, /usr/bin/false is good.
4.   Install an FTP server, the examples here are for ProFTPD.
5.   In /etc/proftpd.conf make sure there are the following lines:
# Anonymous ftp and members of group bbsftp have a chroot environment.
# DefaultRoot      ~ bbsftp
This will make sure that users who are member of the bbsftp group have a chrooted directory structure for their ftp directories. Also make sure the line Umask 022 is changed to Umask 002.
6.   Add a user name for the node you whish to give an FTP feed. Use the following as root:
useradd -g bbs -G bbsftp -d /opt/mbse/var/bbsftp/user name
   -s /usr/bin/false -c "FTP Account for user name" -m user name
Note: this is one line! There will now be a user added and have a home directory of /opt/mbse/var/bbsftp/user name. Make sure that directory has permission 0775 or 0770. Create in the user name directory two other directories, inbound and outbound. They must be owned by the user and be a member of group bbs and have mode 0775. The modes 077x are needed because the user and mbse must be able to read and write to these directories.
7.   Try to login with ftp from some other system with the choosen user name and his password and check the changes until now. You should not be able to escape from the directories of this user and you should be able to upload files, download and delete files. Check also if you uploaded a file if user mbse can delete that file and also put a file in the users ftp directory as user mbse, and check that you can download and delete is as ftp user.
8.   Make sure for the node you want to change there is no mail left in the outbound. Now start mbsetup, menu 7, open the setup of the node that will use ftp. In screen 3, items 7 and 8, set these to Directory. Now enter screen 8, here you will setup the directory session.

```
           7.8  EDIT NODE DIRECTORY SESSION


              Outbound settings
      1.   Files path      /opt/mbse/var/bbsftp/user name/outbound
      2.   Check for lock  Yes              3.   Wait clear lock Yes
      4.   Check lockfile  /opt/mbse/var/bbsftp/user name/lock.bsy
      5.   Create lock     Yes
      6.   Create lockfile /opt/mbse/var/bbsftp/user name/lock.bsy
              Inbound settings
      7.   Files path      /opt/mbse/var/bbsftp/user name/inbound
      8.   Check for lock  Yes              9.   Wait clear lock Yes
      10. Check lockfile   /opt/mbse/var/bbsftp/user name/lock.bsy
      11. Create lock      Yes
      12. Create lockfile  /opt/mbse/var/bbsftp/user name/lock.bsy
```
Save this and you are ready.
A final note, since this user is in group bbs and not in the bbs user database he can never telnet to your system with this account and get a shell. The user doesn't even have a valid shell.


Q 3.3.   Polling a internet node fails.
A 3.3.   There are several reasons why mbcico refuses to call an internet node, most problems are nodelist related. A internet node with binkp protocol should be listed with the flags CM,IBN before mbcico will make a call. Also, in the nodelist system name field there should be the hosts full qualified domain name so that mbcico can get that node's IP address. If one of these items are missing, the node will not be called. To correct this problem there are two options, make sure the node is listed with a dns name and proper flags in the nodelist. The second option is to

add a setup record for that node and fill in the nodelist override fields. You can find these in menu 7.3, item 5 can be filled with the nodelist flags, ie. CM,IBN and item 6 can be used to give that node's fdn or IP address.

If the call still fails there are several things that must be right before mbcico makes any internet call. In mbmon, screen 2 (SERVER STATS) there are flags that indicate if the system is running, internet is available and the bbs is open. If one of these is No, then no call will be made. (For POTS the Internet flag does not matter of course).

In mbsetup, screen 18 (taskmanager), you need to set the maximum TCP sessions to allow. This is inbound and outbound together. So if you set this to 3 and 3 or more systems are already calling in via internet, your system will not call anymore until the amount of TCP sessions drops below 3.

If you set this to high on an average cable/adsl connection, you can get into trouble because you use up all upstream bandwidth. (We have seen a lot of errors at a busy when the upstream speed was 64K and down speed 512K and about 4 sessions together). A good rule seems 1 TCP session for each 32K bandwidth.

Then turn in that same screen debug on. After a minute mbtask rereads his configuration.

Make a poll with mbout poll <node>.

Watch the mbtask logfile and see what it does for that node. It should be added to the calllist. If not, it might be still a nodelist problem.

## Q 3.4.   Dialup script examples for MBSE
A 3.4.   Dialup Script using mclient:

```sh
#!/bin/sh
#
# Dialup script for MBSE BBS.
#
#==========================================================================
# Copyright (C) 1997-2002
#
# Michiel Broek              FIDO:           2:280/2802
# Beekmansbos 10
# 1971 BV IJmuiden
# the Netherlands
#
# This file is part of MBSE BBS.
#
# This BBS is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2, or (at your option) any
# later version.
#
# MBSE BBS is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with MBSE BBS; see the file COPYING.  If not, write to the Free
# Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#==========================================================================
#
#
# Enter commands to make the dialup connection and to hangup the connection.
#
DIALCMD="mclient -d -c DDS"
HANGUPCMD="mclient -k"
#
# Log to syslog
#
LOGGER="logger -p user.notice -t basename $0$$]"

if [ -z "$DIALCMD" ]; then
    echo "This script is not configured, edit this script to use it."
    exit 1
fi

echo "Dial internet start" | $LOGGER
#
# Do 3 attempts to connect
```

```
#
connect=0; try=0;
while let 'connect == 0'
do
    # Count connect attepts
    #
    let try=try+1
    if [ $try == 4 ]; then
        echo "No internet connection, giving up" | $LOGGER
        $HANGUPCMD >>/dev/null
        exit 1
    fi

    echo "Call attempt $try"    | $LOGGER
    $DIALCMD >>/dev/null
    i=60;
    while let 'i > 0' && let 'connect == 0'
    do
        if [ -f $MBSE_ROOT/var/sema/is_inet ]; then
            connect=1
            echo "mbtask detected internet"     | $LOGGER
        fi
        let i=i-1
        sleep 1
    done
    #
    # Check if we have a connection
    #
    if [ ! -f $MBSE_ROOT/var/sema/is_inet ]; then
        echo "Failed, 10 seconds pause ..."     | $LOGGER
        sleep 10
    fi
done

#
# Here we have a connection, now make all needed polls
#
$MBSE_ROOT/bin/mbout poll f5003.n280.z2 f19.n280.z2 -quiet

#
# Now we wait until the do_inet semaphore appears, we
# timeout at 25 seconds, must be enough.
#
i=25;
while let 'i > 0'
do
    if [ -f $MBSE_ROOT/var/sema/do_inet ]; then
        i=0;
    else
        let i=i-1;
        sleep 1
        echo "wait do_inet $i"
    fi
done

#
# Now test until the do_inet semaphore is gone.
# Timeout is one hour.
#
i=3600;
while let 'i > 0'
do
    if [ ! -f $MBSE_ROOT/var/sema/do_inet ]; then
        i=0;
    else
        let i=i-1;
        sleep 1
        echo "wait ready $i"
    fi
```

```
done

echo "Internet sessions done, hangup"    | $LOGGER
$HANGUPCMD >>/dev/null
echo "Script ready"                       | $LOGGER
exit 0


===============================================
Dialup Script using diald:
===============================================
#!/bin/sh

PATH=/opt/mbse:/opt/mbse/bin:/usr/bin:/usr/local/bin:/bin:/usr/sbin:/usr/local
/ sbin:$PATH
cd /opt/mbse
if [ -f /opt/mbse/tmp/ip.up ] ; then
        exit
fi
echo "up" > /var/run/diald.ctl
sleep 15

ping -c 10 -w 10 mail.iol.ie
if [ -e /var/run/ppp0.pid ]  ; then
        touch /opt/mbse/tmp/ip.up
   cd /opt/mbse
   echo "Fetching News"
   /opt/mbse/fetchnews
   echo "Synching clock"
   telnet 192.168.0.2 9049
   bin/mbfido news
   echo "Fetching Mail"
   /usr/bin/fetchmail -f /opt/mbse/.fetchmailrc
   bin/soup
        if [ ! 'cat /opt/mbse/irex/s/rexbusy.0 | xargs ps --no-header' ] ;
        then
                rm /etc/ftn/irex/s/rexbusy.0
        fi

   ./rexl
   fetchmail -f /opt/mbse/.fetchmailrc
   ./rexl -sf m
   /opt/mbse/bin/bink

  echo down > /var/run/diald.ctl

  bin/bres.sh  # >> /dev/null 2>&1
fi
rm /opt/mbse/tmp/ip.up
=================================
```

Q 3.5.  BBS users cannot get their email on Debian

A 3.5.  Debian Linux uses for normal users a security model with private user groups. The users of the bbs are all in group bbs. Some pop3 servers do not handle that situation well and don't show the email for bbs users that is delivered in /var/mail. If you choose to use qpopper it will work.

**4. BBS Setup.**

Q 4.1.   How to set up MBSE for ISDN.
A 4.1    Contributed by: J. Beekhuizen, 2:280/1018
Of course, you have to configure and build your kernel for ISDN support and the correct drivers for your card. It is beyond the scope of this document to explain how to do that.
To your /etc/inittab file add the following lines for your ISDN devices ttyI0 and ttyI1.
# ISDN lines
I1:23:respawn:/usr/local/sbin/mgetty -x 4 -i /opt/mbse/etc/issue ttyI0

I2:23:respawn:/usr/local/sbin/mgetty -x 4 -i /opt/mbse/etc/issue ttyI1

To your mgetty.config file [/usr/local/etc/mgetty_sendfax/mgetty.config on my system] set up the ttyI0 and ttyI1 ports. With AT&E you set the MSN/EAZ the device should listen to. This should for an MSN be your telephone number **with** the area code but **without** the leading 0. For an EAZ the last digit of your EAZ.
The ISDN ports
port ttyI0
modem-type data
init-chat "" ATZ OK
AT&E714015437&W0 OK
AT&B512 OK


port ttyI1
modem-type data
init-chat "" ATZ OK
AT&E714017198&W0 OK
AT&B512 OK
From the mbsetup menu #5 "Edit Modem types" configure the ISDN modem type.

```
        Modem type   ISDN Linux
        Init string  ATZ\r
        Init string
        Init string
        OK string    OK
        Hangup       \d\p\p\p+++\d\p\p\pATH0\r
        Info command ATI2\r
        Dial command ATD\T\r
        Connect      CONNECT 64000
        Connect      CONNECT
        [...]
        Reset cmd    ATZ\r
        Error string BUSY
        Error string NO CARRIER
        Error string NO DIALTONE
        Error string NO ANSWER
        Error string RING\r
        Error string ERROR
        Error string
        Error string
        Error string
        Error string
        Cost offset  0
        EMSI speed   28800                  ! not relevant for ISDN
        Strip dashes No
        Available    Yes
```
And from #6 "Edit TTY lines info" the ISDN lines
```
        TTY name      ISDN Line 1
        Device name   ttyI0
        Phone or DNS  31-71-4015437
        Line speed    64 kbits
        Fido flags    CM,XA,MO,X75
        Equipment     ISDN
        Available     Yes
        Auth. log     No
        Honor ZMH     Yes
```

```
        Callout      Yes
        Modem type   ISDN Linux
        Locked speed 0                             ! not relevant for ISDN
        EMSI name    The Wizard's ISDN line 1

        TTY name     ISDN Line 2
        Device name  ttyI1
        Phone or DNS 31-71-4017198
        Line speed   64 kbits
        Fido flags   CM,XA,MO,X75
        Equipment    ISDN
        Available    Yes
        Auth. log    No
        Honor ZMH    Yes
        Callout      Yes
        Modem type   ISDN Linux
        Locked speed 0
        EMSI name    The Wizard's ISDN line 2
```

That's all folks :))

Q 4.2.   How do I use GoldED together with MBSE?

A 4.2.   Since MBSE BBS version 0.33.12 GoldED and MBSE BBS can be used together without problems as long as you use it to read the sysop mail. The mbsetup program can export a file called /opt/mbse/etc/golded.inc which will contain your main Aka's, Aka matching, sysop name and all your mail areas. This file is only (re)created if you change the global settings or one of the mail areas. The first time you must force this by making a change somewhere.

Now create /opt/mbse/etc/golded.cfg, here is what I wrote:

```
; GoldED.cfg
;
; Internet Addressing
;
INTERNETADDRESS Michiel_Broek@f2802.n280.z2.fidonet.org
INTERNETGATE UUCP 2:292/875
;
;
OUTBOUNDPATH /opt/mbse/var/bso/outbound
REPLYLINK chain
STYLECODES yes
;
;
; MESSAGE READER
;
DISPMSGSIZE KBYTES
DISPATTACHSIZE KBYTES
DISPLOCALHIGH YES
DISPPAGEBAR YES
VIEWHIDDEN YES
VIEWKLUDGE NO
VIEWQUOTE YES
;
INCLUDE /opt/mbse/etc/golded.inc
;
; The end.
```

Put in /opt/mbse/.profile the following line: export GOLDED=$HOME/etc

When you now start GoldED you use it as the sysop. Make sure that the sysop's user record is the first user in the MBSE BBS userbase. If not, the lastread pointers are not right. The GoldED nodelist compiler can be added to the setup in menu 18.

Q 4.3.   How do I use MsgEd together with MBSE?

A 4.3.   Since MBSE BBS version 0.35.05 MsgEd and MBSE BBS can be used together without problems as long as you use it to read the sysop mail. The mbsetup program can export a file called /opt/mbse/etc/msg.txt which will contain your mail areas setup. This file is only (re)created if you change the global settings or one of the mail areas. The first time you must force this by making a change somewhere.

Q 4.4.   How do I adopt (batch upload) a directory full of files to my BBS's file base?

A 4.4.   Written by: Russell Tiedt.

Easy using the script below, provided the file archives (*.zip, *.rar, *.arc, *.lha, etc.) all contain "file_id.diz" files.

Sort the files in temporary directories that correspond to the filebase area, ie. /dos4 (DOS files - filebase area no 4) Then copy the script below to the directory, edit the number of the filebase that the files must be batch uploaded to, to the number that corresponds to the required filebase. (On my BBS system the DOS file section corresponds to the number 4, hence the name above for the temporary directory used above) I copy all files to be batch uploaded to the DOS file section, to the /dos4 directory, copy the script below to the /dos4 directory, edit the script so that it uploads to the DOS filebase (4) and run it.

```
for file in `ls` ; do
  mbfile adopt 4 $file "Dummy description"
done
```

NOTE: the 4 in the script above, indicates the filebase/filearea that the files will be batch uploaded to. For archives that have/contain a file_id.diz file, "Dummy description" will be replaced with the content of the "file_id.diz file of that archive, if no file_id.diz exists in the archive, then "Dummy description" will appear in the description field of the files.bbs file for the file area it gets uploaded to.

If the files don't have/contain file_id.diz files, then create and insert in the archive, a file_id.diz file having a suitable description of the content of the archive, then follow the instructions above.

For *.exe and *.com files, as well as other files that are not archived ie. in *.zip/*.rar/*.arc/*.lha/*.lhz/*.zoo etc. Create a file_id.diz file for the file, then make an archive using the archiver of your choice containing the file and the file_id.diz you created for it, then follow the instructions above.

**5. Doors Setup.**

Q 5.1.   How do I run DOS doors under MBSE BBS?
A 5.1.   Written by: Redy Rodriguez, sysop of Parolas BBS 2:348/609
Fidonet:    Redy rodriguez 2:348/609
Email:      redy@users.sourceforge.net
          Redy_Rodriguez@f609.n348.z2.fidospain.org
*Note: the next method is obsolete, with newer versions of dosemu it is possible to run the doors without sudo but with user privileges. See the doors page for details.* **The information below is for old versions of dosemu.**
**NO WARANTEE**
The information gathered here, works for the indicated intention. It is proven and working in PAROLAS BBS, however, I do not guarantee that it works correctly in all the cases. The author is not responsible for the damage this can cause. If you follow the instructions, and you get not the expected result, or as a result of it you get undesired results, you loose data or destroy your system, you you will be the only person in charge.
**INTRODUCTION**
This document describes how I have been able to execute MS/DOS doors in MBSEBBS. Probably it is not the best way to do it, but this is the one that works for me. If somebody finds one more more effective, or simpler or safer a solution, then I like to hear this solution.
**WHAT IS NEEDED?**
MBSEBBS is correctly compiled and working, in my case I have the version 0.33.19, I suppose that this method can be valid for newer versions. DOSEMU 1.0.2 I have tried previous versions, but these do not work correctly with virtual ports. You can download dosemu.1.02.tgz from [www.dosemu.org](www.dosemu.org) , you may also need to download the freedos archive dosemu-freedos-bin-1.02.tgz. A partition with MS/DOS, I have on my disc a partition with installed MS/DOS, and that is the one that I use, but I'm sure that many doors could work correctly with the freedos. You also need **sudo**. Sudo is a utility that let's ordinary users execute certain programs as superuser, this is available with almost all GNU/Linux distributions.
**COMPILING DOSEMU**
Unpack dosemu in some directory, for example /usr/src, then changed into that directory and execute 'make'. When this is ready we have dosemu compiled. If we already had a previous version of dosemu it is advisable to uninstall it before continuing. Copy dosemu-freedos-bin-1.02.tgz to the toplevel directory of the dosemu source and execute:
./install_systemwide -fd dosemu-freedos-bin-1.02.tgz
In the directory etc, within the directory where we have decompressed the source code of dosemu we have configuration examples. Create the directory /var/lib/dosemu if it does not exist the directory, and copy global.conf to that directory. Also copy dosemu.conf to directory /etc and copy dosemu.users.secure as dosemu.users to /etc.
**CONFIGURE THE DOS PARTITION**
Now we must configure the dos partition, in my case this exists as a mounted dos partition in /dos/c. If this is not the case then you could work with an image of freedos. I have created in my dos partition a directory c:\doors where I install all doors that I want to use, and a file called c:\doors.bat with the following contents:
```
@echo off
c:
cd \doors
if exist %1.bat call %1 %2 %3 %4 %5 %6 %7 %8 %9
c:\dosemu\exitemu
```
Change to the directory /var/lib/dosemu and create directory called for example 'c':
mkdir c.
A link to the mounted partition could be made but I prefer to create a new directory and in that directory make links to dos program which are only of interest for us. Many programs and directories of the dos partition are not needed in dosemu and to run doors. Change to the created directory and make links to everything what we have in our dos partition (we will erase soon what we do not need). Autoexec.bat and config.sys must copied instead of making links so we can modify them, because surely we will want that they are different in dosemu, then in real dos.
```
cd c
ln -s /dos/c/* .
rm autoexec.bat
rm config.sys
cp /dos/c/autoexec.bat config.sys .
```
Create a directory dosemu and copy the typical utilities of dosemu.
```
mkdir dosemu
```

```
cp /usr/src/dosemu-1.02/1.0.2.0/commands/* dosemu
```
Now we modify the copy of config.sys and autoexec.bat to erase everything what is not necessary to execute our games as doors, like drivers for cdrom, mouse etc... We will clear the keyboard driver 'KEYB XX' and make sure that in the path in autoexec.bat to include the directory c:\dosemu. Also load the fossil emulator, for example add these two lines:
```
path %PATH%:c:\dosemu
c:\dosemu\fossil
```
If there is another fossil, such as bnu or x00 in config.sys or autoexec.bat we clear all references to these, we will load them if that is necessary in the file that it executes the door when needed, this can be necessary for some doors and for other not. Now we can erase all the links to the directories who are not necessary, and that we do not want that they are visible.

Then edit the file /etc/dosemu.conf and look for the line that begins with "$_hdimage =" and change that to:
```
$_hdimage = "c"
```
Now login as root and you should be able to execute our dos emulator by executing '/usr/bin/dosemu.bin'
(to leave dosemu type in exitemu)

**CONFIGURING SUDO**

Normally the dos partition is not accessible to the users of the BBS, except from within dosemu, and to be able to have access without activating the setuid bit of dosemu, we use sudo so that any user of the BBS can execute dosemu.bin as root. As root execute visudo to edit the configuration file of sudo add the lines:
```
%bbs    ALL=NOPASSWD:/opt/mbse/bin/bbsdoor.sh
%bbs    ALL=NOPASSWD:/opt/dosemu/bin/dosemu.bin
```
The first command is a script that takes care of copying the file door.sys to the dos partition, the second line is to allow to execute dosemu.bin. Any user of group BBS can execute these commandos like root without password is asked.

In order to test that this works login as user mbse and execute:

sudo dosemu.bin

**INSTALLING A DOOR**

First we need a pair of scripts. These scripts are installed when you did install MBSE. These scripts are needed to execute the doors and are installed in /opt/mbse/bin:
```
--- rundoor.sh ---

#!/bin/bash
#
# rundoor.sh - Never call this script directly, create a symlink
#              to thisfile with the name of the door. For example
#              tu run the door ilord do:
#              cd /opt/mbse/bin
#              ln -s rundoor.sh ilord
#
# by Redy Rodriguez and Michiel Broek.
#
DOOR=`basename $0`
COMMANDO="\" doors $DOOR $*\r\""

/usr/bin/sudo /opt/mbse/bin/bbsdoor.sh $DOOR $1
/usr/bin/sudo /opt/dosemu/bin/dosemu.bin \
-F /var/lib/dosemu/global.conf \
-I "`echo -e serial { com 1 virtual }"\n" keystroke $COMMANDO`"
reset
tput reset
stty sane


--- bbsdoor.sh ---


#
#  Initialize DOS environment before starting a bbs door.
#  Parameters: $1 = name of the door
#              $2 = the nodenumber for this session
#
#  by Redy Rodriguez, 22-Oct-2001
if [ "$1" != "" ]; then
    if [ "$2" != "" ]; then
```

```
        mkdir -p /dos/c/doors/node$2 >/dev/null 2>&1
        # Copy door.sys to dos partition
        cat ~/door.sys >/dos/c/doors/node$2/door.sys
        # Create .dosemu/disclaimer in user home to avoid warning
        if [ ! -d $HOME/.dosemu ]; then
            mkdir $HOME/.dosemu
        fi
        if [ ! -f $HOME/.dosemu/disclaimer ]; then
            touch $HOME/.dosemu/disclaimer
        fi
    fi
fi
```

The script rundoor.sh is never called directly, simply create a link with the name of the door to this file, this name must be the same as the one of the .bat file that starts the door in the doors directory of the dos partition. Let me explain this with an example:

I have chosen the door Virtual Sysop, but it could be any other door. My experience says to me that not all doors work correctly with a serial port in virtual mode. This one is working at Parolas BBS.

Unpack Virtual Sysop in the dos partition, in the directory c:\doors\vsysop. Execute as user mbse 'sudo dosemu.bin' and make sure that the door works in local mode (these commands are for this door):
```
cd doors
cd vsysop
vsysop -local
```
If everything is correct then create the file c:\doors\vsysop.bat with the following contents:
```
--- vsysop.bat ---

@echo off
C:\COMUNIC\X00\X00 E B,0,57600
cd vsysop
vsysop -D c:\doors\node%1\door.sys -BBSNAME PAROLAS
deltree /y c:\doors\node%1
C:\COMUNIC\X00\X00
cd ..
```
Before passing the control to dosemu, the file door.sys will be copied to c:\doors\node???? where the ???? is the node number from which the door is invoked (to allow that several users can execute the door at the same time). This nodenumber also is passed as the first parameters to the .bat file, this is to indicate to which directory to copy the door.sys file.

The Virtual Sysop door supports door.sys and that is the only dropfile that is generated by mbsebbs and that is supported by most of the doors. If you need another format you must use a converter to change the format of the dropfile.

I use the fossil X00, I have it installed in C:\COMUNIC\X00 and this door works for me correctly with these parameters, it is alos possible that it works without X00 or that you can load another fossil.

Now create a link to rundoor.sh in /opt/mbse/bin called vsysop. The name must be the same one that the name of the .bat file created in the dos partition, but without the extension bat.

ln -s /opt/mbse/bin/rundoor.sh /opt/mbse/bin/vsysop

Now we execute mbsetup and enter the menu setup to add an entrance like this (It puts option 16 to Yes):
```
    8.3. EDIT MENU ITEM

    1.  Sel. key  V
    2.  Type nr.  7          Run external program in shell
    3.  Opt. data /opt/mbse/bin/vsysop /N
                  12345678901234567890123456789012345678901234567890123456789012345
    4.  Display
    5.  Security  0       -------------------------------
    6.  Min. age  0
    7.  Max. lvl  0
    8.  Password  <null>                 13. No door.sys No
    9.  Credit    0                      13. Y2K style   No
    10. Lo-colors Normal display colour   14. Use Comport Yes
    11. Hi-colors Bright display colour   15. Run nosuid  Yes
    12. Autoexec  No                      16. No Prompt   Yes
```
And that is everything...

Q 5.2.   Dosemu compile tips

A 5.2.   Written by: Rick Van Ruth, sysop ofVampyre's Heaven BBS 3:640/954

Dosemu still exhibits some file locking behaviour and this needs to be overcome, especially if you are running multinode door games. The following information was supplied by Bob Newell of Chung Kuo BBS (telnet chungkuo.org) and I reiterate it here.

There is an issue in that GNU/Linux file locking does not map precisely to DOS file locking. You need to patch DOSEMU and build a special version to deal with this problem. Find the file "dosext/mfs/mfs.c" in your DOSEMU source tree. Locate the following lines:

```
case DENY_ANY:
fl.l_type = F_RDLCK;
```

Change the second line to

```
fl.l_type = !writing ? F_RDLCK : F_WRLCK;
```

and build a new version of DOSEMU.

To solve a problem of slow screen painting, locate "base/async/int.c" in your source tree. Locate

```
case 0x2C:
```

and just below, comment out or delete the line

```
usleep(INT2F_IDLE_USECS);
```

and rebuild your DOSEMU.

It is recommended to use both tips above as it will not harm any dos door games.


Q 5.3   Setting up for InterBBS games.

A 5.3.   Written by: Rick Van Ruth, sysop ofVampyre's Heaven BBS 3:640/954

There are 2 methods for providing InterBBS setup. If your system merely connects to one host for your games packets you need only perform a few simple steps to have it working. If your system is a Host for InterBBS games you will also need to add a script to separate different packets for different downlink systems.


**Common to Both Methods:**


With most BBS games you will need 2 directories in your dos file system area. One for the inbound packets and one for netmail messages door games produce to send outbound packets.

The easiest way to solve this is by creating symlinks from your MBSE directories into the directory where your dos doors live (c:\doors). The 2 directories you wish to link are /opt/mbse/var/inbound and /opt/mbse/var/msgs - link them into your dos directories doors directory with the same names ie: "inbound" and "msgs". Now according to dosemu you will have a c:\doors\inbound and a c:\doors\msgs. We do not actually use the "msgs" directory for anything other than a place to tell games where to write its netmails to. These "netmails" are subsequently deleted from the system, but the games need to be able to write them there.


1. **The Easy Way - Single Host System:**

1A. First of all you will realise that door games running under dosemu usually write files as owner root, chmod 0644. This is not good for MBSE to transfer these files as it does not have permission to act on the files. In order to force dosemu to write games files so that MBSE can act upon them we need to alter /etc/sudoers and change the umask for root running under sudo. To do this open /etc/sudoers (you must be root to edit this file) and add the following line at the top of the file before the specifications:

Defaults        umask=0000

This will allow dosemu to write files chmod 0666 (read/write everybody) and MBSE will be able to manipulate them.

For my example I will use the InterBBS dos door game called Barren Realms Elite (BRE for short).

1B. Install Bre in c:\doors\bre and setup the necessary files for its execution as a dos door from MBSE as per the dos doors faq.

1C. Tell Bre (in its BBS.CFG) that the path for inbound packets is c:\doors\inbound and the netmail path is c:\doors\msgs Bre will now automatically process all inbound packets it finds in the inbound directory for itself. It needs no assistance as it merely searches for the correct file name in the directory.

1D. We need to link Bre's outbound path to our host nodes MBSE mailbox for automatic transfer of games packets. First make sure you have added an "outbox" for your uplink node in the nodes setup in MBSE - Section 7, Select

Uplink, Menu 1, Section 2 "outbox dir". eg: if your uplink is 3:640/954 then MBSE will default to creating /opt/mbse/var/boxes/node3_640_954 as the outbox (I will use this as an example). Note the flavour of your uplinks node connection will apply to the outbox as well. Thus if that particular node is set crashmail then anything you put in the outbox will also be sent crashmail.

Next go to your Bre game directory. You will notice Bre has an OUTBOUND directory, this is where it places all its game packets that are to be sent to the uplink. We need to delete this directory and instead create a symlink to the node outbox instead. If Bre is in c:\doors\bre then the symlink command will be:

ln -s /opt/mbse/var/boxes/node3_640_954 /dos/c/doors/bre/OUTBOUND

Alter the above depending on where your linux path to your dos is, also remember linux is case sensitive! Note you can also make as many symlinks as you like to the nodes outbox, so if you have 2 or 3 different games you send to that node you can link the outbox to each games outbound directory. Also if you have different games to different host nodes just link the game outbound directories to the outboxes of each node.

Thats it. The system will now process inbound and outbound packets automatically. You will however need to delete the netmails in the msgs directory - I have included this in the Door game maintenance section further on.

If you experience problems with your system polling nodes and sending the game mail packets while a user is online within the game then you might wish to use the dos "share.exe" so packet files are not being accessed by both the game and the mailer at the same time. You can load share.exe within your doors.bat in dos root directory.

**2. The Hard Way - Multiple Host System**
Follow steps 1A, 1B, 1C above, the change occurs for 1D (below).

1D. We need a script to process the Bre outbound packets and add them to MBSE's node outboxes path so they are sent. A script on how to do this follows (with comments ). First remember to create your outboxes for each node that you transfer games packets to. I will use 3:640/954 as above and 1:123/456 for this example of 2 systems linked to the one Bre game.

This script should be owned by user "mbse" and have the permissions (chmod 755) -rwxr-xr-x. It is best kept in /opt/mbse/bin with your dos game executing scripts. I call this script "dos_poll"

```
-----------------------------------------------------------------------
#!/usr/bin/perl -U
use File::Basename;
use File::Copy;

# Define the variables for the outboxes for our links
$node1 = "/opt/mbse/var/boxes/node3_640_954";
$node2 = "/opt/mbse/var/boxes/node1_123_456";
# Define the linux path to Bre's outbound directory
$bre_path = "/opt/mbse/dos/doors/bre/OUTBOUND";

opendir(DIR, "$bre_path") || die;
@dirlist = readdir(DIR);
closedir(DIR);

foreach $file name (@dirlist){
    $brefile = "$bre_path/$file name";
    ($filenameOnly, $pathname, $fileExtension) = fileparse($file name, '\..*');

    if($filenameOnly eq "048b0201") {
        move("$brefile","$node1");
    }
    if($filenameOnly eq "048b0203") {
        move("$brefile","$node2");
    }
}

# '048b0201' is the packet name Bre creates. You can find out what
# your packet name is by running BRE PLANETARY and then checking inside
# the Bre OUTBOUND directory - note no ext on name. You should also check
# your nodes dat for which is which 048b0201 means from system 2 to system 1
# so you would be system 2 in the nodes dat and 1 would be 3:640/954
# 048b0203 is obviously to system 3 in the nodes dat - 1:/123/456 in our
# example here
```

That's it. You can run this 'dos_poll' script from the maintenance scripts described below.

Q 5.4.   Door Maintenance.
A 5.4.   Written by: Rick Van Ruth, sysop ofVampyre's Heaven BBS 3:640/954
Ok, so you can setup doors, run inbound and outbound for interbbs games. So how do you do maintenance?

Again by scripts. Again I will use BRE as an example. First I create a maintenance batch file in my
C: root directory. I have called it maint.bat and for the bre game it looks like this:

```
cd \doors\bre
BRE INBOUND
BRE SCORES
BRE REQUEST
BRE PLANETARY
exitemu
```

If you are familiar with Bre you will know this to be a standard sort of maintenance except for the "exitemu" at the end. That is there because we are going to use a script to start dosemu and call the maint.bat - so afterwards we need it to exit dosemu.
Now a maintenance script to be called from your cron or whatever else you are using to execute events on your system. I call this script "dos_maint" and it also lives in /opt/mbse/bin

```
#!/bin/sh
#
COMMANDO="\" maint \r\""
/usr/bin/sudo dosemu -dumb -E dir > /opt/mbse/log/dos.log -quiet -I "`echo keystroke $COMMANDO`"
/bin/rm /opt/mbse/var/msgs/*.msg
#/opt/mbse/bin/dos_poll # Uncomment this lineif you are using the Multiple host script
```

You will notice this script starts dosemu and calls maint.bat, and then when its finished there it deletes all those games netmails and if you run the script for multiple hosts for your games it can call that as well. Please note the second line is all ONE line, it has most probably wrapped here.
You will notice that the script calls dosemu and not dosemu.bin, this is because we need to tell dosemu to open a dumb terminal and not display to a console or screen. By use of the redirection we redirect the normal console output to a file, here I have used /opt/mbse/log/dos.log as that is where all of MBSE's other logs reside.
Once "dos_maint" is run via the cron it will run maintenance on your dos doors, process inbound/outbound interBBS game packets and have your outbound packets ready to send. That's about all there is to do :)

Note: You can make variants on these scripts for running daily maintenance or just packet processing
at certain times of day. They are just variations on the information above.
Have fun!

# Appendix 2 - Fidonet Technical Standards.

**Introduction**
This is an overview of documents used for the development of the MBSE BBS package. Note that there are more documents, but only the relevant and valid documents are shown here. The documents are not available in this distribution any more, you can get these from the FTSC website.

**FSC Documents**
- FSC-0035 Transparent Gateways to and from FidoNet, Michael Shiels
- FSC-0039 A type-2 packet extension proposal M.Howard
- FSC-0046 A Product Identifier for FidoNet Message Handlers, J.Homrighausen
- FSC-0048 A Proposed type-2 packet extension, J.Vroonhof
- FSC-0049 A proposal for passing domain information during FTS-0006 sessions, B.Hartman
- FSC-0050 A character set identifier for FidoNet message editors, T.Sundblom
- FSC-0053 Specifications for the ^aFLAGS field, J.Homrighausen
- FSC-0056 EMSI/IEMSI Protocol Definition, J.Homrighausen
- FSC-0057 Conference Managers - Specifications For Requests, F.Fabris, J.Homrighausen
- FSC-0059 Newsgroup Interchange within FidoNet, J.Decker
- FSC-0062 Nodelist Flag indicating Online Times, D.Thomas
- FSC-0070 Improving FidoNet/UseNet Gating and Dupe checking, F.Arnoud
- FSC-0072 The HYDRA file transfer protocol, J.Homrighausen, A.Lenz
- FSC-0087 File forwarding in FidoNet technology networks, R.Williamson
- FSC-0088 Compatibility and Link Qualifier Extensions for EMSI Sessions, R.Williamson
- FSC-0091 ISDN nodelist flags (rev.002), A.Lenz
- FSC-0093 Reduced seen-by lines, F.Ellermann

FSP Documents
- FSP-1011 BinkP - a protocol for transferring Fidonet mail over reliable connections, Dima Maloff
- FSP-1013 Character set definition in Fidonet messages
- FSP-1018 Binkp/1.0 Protocol specification
- FSP-1019 Binkp 1.0 optional protocol extension CRAM
- FSP-1026 Translation of a Fidonet Address into an Internet Domain Name
- FSP-1030 Unicode character set in FidoNet messages
- FSP-1032 Binkp optional protocol extension Dataframe Compression
- FSP-1033 The ICM Flag

FTA Documents
- FTA-1005 FTSC Product ID
- FTSC Product codes list

FTS Documents
- FTS-0001 A basic FidoNet(r) technical standard, R.Bush
- FTS-0004 Echomail specification, B.Hartman
- FTS-0006 YOOHOO and YOOHOO/2U2, V.Perriello
- FTS-0007 SEAlink protocol extension, P.Becker
- FTS-0008 Bark file-request protocol extension, P.Becker
- FTS-0009 Message identification and reply linkage, J.Nutt
- FTS-1024 Raw ifcico mail transfer protocol
- FTS-4001 Addressing Control Paragraphs, Goran Eriksson
- FTS-4008 Time zone information (TZUTC)
- FTS-4009 Netmail tracking (Via)
- FTS-5000 The distribution nodelist, David Hallford
- FTS-5001 Nodelist flags and user flags, David Hallford

# Appendix 3 - Miscellaneous Documents.

**Introduction**

This is an overview of used unofficial documents for the development of the MBSE BBS package.

**Documents**

- BBS doors dropfiles, M. Broek
- Implementation and Usage of Filefind Utilities, R.Williamson
- FILE_ID.DIZ Information, R.Moller
- Integration of IP-Nodes in the nodelist, L.Behet
- How to setup a FTP server with MBSE BBS, M.Broek
- JAM Message Base Proposal, J.Homrighausen
- Binkley style mailer outbound for MBSE BBS, M.Broek
- semaphore files for MBSE BBS, M.Broek
- System load and usleep() code, M.Broek
- How to setup a web server with MBSE BBS. M.Broek

# BBS doors Dropfiles.

## Dropfiles for Unix BBS systems.

Not all options that are available under DOS or OS/2 can be used with Unix BBS systems and must be faked. The doorfiles are written in the users homedirectory when the user starts a door.

## DOOR.SYS format.

The door.sys format is a 52 lines ascii textfile, each line is terminated with a cr/lf pair. In the setup it is possible to force the creation of MM-DD-YYYY dates instead of the MM-DD-YY style. Newer doors sometimes need that.

| Line | Description |
| --- | --- |
| 1 | Port, 5 characters in DOS format, COM1: or COM0: |
| 2 | Effective Baudrate, 19200 or 0 for COM0: |
| 3 | Databits, always 8 |
| 4 | Nodenumber, 1..9999 |
| 5 | Locked baudrate, 19200 or tty portspeed for COM0: |
| 6 | Screen display, Y=snoop on, N=snoop off, on Unix allways Y. |
| 7 | Printer Y=on N=off, on Unix always N |
| 8 | Page Bell Y=on N=off, on Unix allways Y |
| 9 | Caller alarm Y=on N=off, on Unix allways Y |
| 10 | Users first name and lastname |
| 11 | Users location |
| 12 | Voice/Home phone |
| 13 | Work/Dataphone |
| 14 | Users password. |
| 15 | Security level, 0..32768 |
| 16 | Users number of calls |
| 17 | Users last call date MM-DD-YY or MM-DD-YYYY |
| 18 | Seconds remaining this call |
| 19 | Time left in minutes |
| 20 | ANSI, "GR" is yes, otherwise "NG" |
| 21 | Screen length |
| 22 | User mode, always N |
| 23 | Always blank |
| 24 | Always blank |
| 25 | Subscription expire date MM-DD-YY or MM-DD-YYYY |
| 26 | Users record number |
| 27 | Default protocol |
| 28 | Users total number of uploads |
| 29 | Users total number of downloads |
| 30 | Users daily download kilobytes total |
| 31 | Daily download kilobyte limit |
| 32 | Users date of birth MM-DD-YY or MM-DD-YYYY |
| 33 | Path to users database files, cannot be used on Unix |
| 34 | Path to message database files, cannot be used on Unix |
| 35 | Sysop first and last name |
| 36 | Users handle |
| 37 | Next event starting time, "none" on Unix |
| 38 | Error-free connection Y=Yes or N=No, always Y |
| 39 | Always set to N |
| 40 | Always set to Y |
| 41 | Text colour as defined in setup 7 = gray. |
| 42 | Always 0 |

| 43 | Last new files scan date MM-DD-YY or MM-DD-YYYY |
|----|--------------------------------------------------|
| 44 | Time of this call HH:MM |
| 45 | Time of last call HH:MM |
| 46 | Always set to 32768 |
| 47 | Number of files downloaded today |
| 48 | Total kilobytes uploaded |
| 49 | Total kilobytes downloaded |
| 50 | Comment stored in users record |
| 51 | Always set to 0 |
| 52 | Total number of messages posted |

## DOOR.SYS format.

The door32.sys format is a 11 lines ascii textfile, each line is terminated with a cr/lf pair.

Line    Description
-----   --------------------------------------------------------------
1       Com type: 1=serial, 2=telnet, with mbse always 1
2       Com port: 1 or 0 for local mode
3       Effective baudrate: 19200 or 0
4       BBS name
5       User record number
6       User's real name
7       User's handle
8       User's security level
9       Time left in minutes
10      Graphics mode: 1 or 0
11      Node (line) number

# fsc-00xx Implementation and Usage of FileFind Utilities

Version:   0.6
Date       Aug 30, 1995
Title:     Implementation and Usage of FileFind Utilities
Authors:   Robert Williamson FidoNet#1:167/104.0  robert@ecs.mtlnet.org


Intro

A portion of the document is derived from information in AllFix.DOC by Harald Harms @ 2:281/910 with  additional sections  from FQuery.DOC by Robert Williamson @ 1:167/104

The  MSdos program ALLFIX by Harald Harms first introduced the idea of searching for files via echomail.  The term applied to this function is  'FileFind'.   A FileFind system allows sysops, points and BBS users to  search  for files  by  placing  a  message  to 'ALLFIX' in an echo designated  for  the purpose of finding files. All FTN sites running a FileFind  processor which is configured to scan that echo will reply to that  user if there any files matching his query.  This system provides a  method  for  searching  many  FTN sites throughout the world, with a single message.

FileFind  programs  work by either scanning through defined message bases or scanning packets for defined AREA tag names for messages to the default  name  ALLFIX.   All FileFind programs MUST respond to the name ALLFIX, but  may also respond to the name FILEFIND and the name of the particular  FileFind  program  in  use  or  defined for the echo.  The FileFind  program  will  process  these  messages, examining the Subject field  for  search queries.  If any valid query is found, the FileFind program  will search  the  sites files database for files matching the users's query.

If the FileFind program finds any matches, it will generate a reply containing  a list of the files found, and some basic information ABOUT the  system  posting  the reply.  When the user who initially wrote the request  reads  the reply, he will then be able to decide if any of the reported  files  meet  his  needs,  and  from the ABOUT included in the reply, learn where and how he may get those files.

FileFind Query Message Structure

To: name_of_FileFind program

The  message  must be addressed to ALLFIX so that all FileFind programs can  respond.   To  use  features specific  to  a  particular FileFind program,  or  to  limit  the  responses  to  a particular platform, the message should  be  addressed  to  that program's name.  Some FileFind programs  will  respond  to more than two names.

Subject:
A space-separated  list  of  file  specifications,  keywords or quoted strings.

keyword    - single word preceded by a '/' with no intervening spaces, must be at least 3 characters, not including the '/'. a keyword search is in actually a substring search of the site's filelist.

description - string enclosed in double-quotes, if a single word, must be more than 3 characters.

filespec   - single word, no spaces, no double-quotes or preceding /, must be at least 3 characters, not including any wildcard  or  pattern  matching  characters,  such  as '*'. Messages addressed to ALLFIX must not have any embedded pattern matching characters.


The  minimum  number  of  characters  for  description, keyword and
filespec  queries  is an implementation detai of the FileFind program.

These values should be configurable, but should never be settable to values of less than 3.

Each implementation should allow the operator the ability to configure a list of disallowed keywords.

## NetMail Queries

Some FileFind programs may also have the ability to process file search queries received as netmail and addressed to the name of the particular FileFInd program with this capability. In this case, all replies are via netmail also.

## NetMail Commands

FileFind Netmail commands are identifed by a leading '%'. Implementation of netmail commands is optional. If implemented, compliant FileFind utilities should be able to process the following minimum NetMail command set.

```
%HELP       - netmail only, returns an extended help text for the
              FileFind program, the ABOUT of the the site and a list
              of MAGIC freqable names.
%ABOUT      - netmail only, returns the ABOUT of the site and a full
or %MAGIC     list of MAGIC names.

%NEWFILES   - netmail only, returns the NEWFILES list of the site
or %NEW       via netmail.
```

### Extended NetMail Commands:

Implementation of the following netmail commands is optional and not required for compliance with the FileFind NetMail Command set.

```
%REPORT <tagname>
        - sends a configuration report for echo <tagname>
          this allows an echo moderator to check if a site running
          a FileFind utility is compliant with the rules of the
          filefind echo.

%REQUEST <file name>
        - if found, will place requested file on hold for remote
          site

%UUREQUEST <file name>
        - if found, and the filesize after uuencoding is less
          than 60K, it will be sent as multiple netmail messages
```

## The Site ABOUT

Obviously, a system that neither accepts file requests nor allows users to download on their first call should not be responding to FileFind messages. If there are any limitations for the caller to acquire any of the files that the site has advertised as being available in it's FileFind response, these limitations MUST be listed in the reply. This information should be included in the ABOUT file that the FileFind program user creates.

The site ABOUT should contain the following information. The FileFind program implementor should instruct his users on these requirements.

- sitename
- site operator's name
- complete phonenumber
- baud rate
- hours during which filerequests are accepted, if at all
- hours during which users can download
- conditions for file requests and user downloads
NOTE: the above information should be within the first 14 lines.
optional:
- a list a MAGIC names
- an indication if magic names are also available to terminal users.

Searching for Files and Creating Replies

The method used by the FileFind program to search for requests is up to the implementor. However, if searching a list, the FileFind program should confirm the actual existance of all files that match the query specification.

The FileFind program should only process description strings, filespecs or keywords that contain more than 3 valid characters and should have configuration options to define greater minimum lengths on a per-echo basis.

For filespecs, the wildcard character '*' IS considered a valid specification as well as the '?' wildcard, but only the '?' is to be counted as a character when determining the length of query. File extensions are not necessary and any characters AFTER a '*' are to be ignored. The FileFind program should be configurable so as to allow replacement all of the file extensions with '.*' or '#?' dependant on platform. This results in queries being independant of the various archivers in use.

Replies

Replies created by FileFind utilities are expected to be in compliance with the following FTN specifications:
    FTS-0001   - packed message format
    FTS-0009   - MSGID/REPLY
    FSC-0046   - PID and tear line

In addition, a FileFind utility may use the FID: control line for any information needed that cannot be put in a PID: without violating that specification.

    ^AFID: ascii text CR

Must be less than 80 characters including ^A and terminating CR.

There are three ways in which the FileFind program can create replies:

- write the replies in the echo in which the query appeared.
- write the replies in an echo that has been specifically
  designated for that purpose in the particular FTN or for
  a gorup of echos in that FTN.
- reply via routed netmail.

    Since each FTN site connected to a particular FileFind program area
is  capable  of creating an information reply, there is much concern as
to  the  amount  of  traffic  that  can  be generated, FileFind program
developers  must  be sensitive to these concerns by providing the means
to  their users to limit the traffic on a per-echo basis.  For example,
various  FileFind  echos  have  rules  limiting  the  size or number of
replies,  or  the length of the system information that may be included
in a reply.

Limiting replies

It is strongly suggested that some default limitations be built-in.

Limiting Site Header (ABOUT):

    If the site's ABOUT, (the text that has been configured in order to
add  the  system's  information  and Magic names list to the reply), is
greater  than  14  lines,  the  remainder should NOT be posted.  A line
should  be  added  to  the response indicated this, and the user may be
invited to either Freq or download the MAGIC name's ABOUT or MAGIC, for
a  full  list of magic names.  The FileFind program may optionally send
the full system information and magic name list via routed netmail.

Limiting Match List due to ambiguity of query:

    If  the list of matches (note:  not the size of the message itself)
is  greater than 32K, the FileFind program should post a message to the
user to indicate that his query may have been too ambiguous and perhaps
invite him to freq or download the MAGIC name FILES for a full list.

Splitting Match List into Multiple Messages:

    If the list of matches is greater than 10K, it should be split into
multiple  messages  of  no more than 8K.  Although the backbone permits
messages  up  to  16K  in length, 8K is a more readable size.  Only the
first  split  message  may  contain  the ABOUT information of the site.
Each  message must be given both a unique Subject field (eg:  prepended
by  "Part n/n") and a unique MSGID:.  This because some tossers may use
either or both for dupe detection.

Limiting Number of Split Messages:

    If  the  number of messages is greater than the preset limit of the
echo,  and  the FileFind Program does not have an option to forward the
replies  via  netmail,  the  replies  should  be discarded and the user
informed that his request may have been too amibiguous.


NetMail Reply:

The  FileFind program may have an option to forward all replies via
routed netmail, or to do so under certain conditions as outlined above.
Obviously, if the FileFind program can process netmail queries, it MUST
respond via netmail.

User NetMail Reply Request:

   Alternativly the user can request a netmail reply for his echomail
query by preceeding the query with either "%" or "!".
 eg;
   Subject:  % /fsc /fts

   If  the  FileFind  program  does  not support this feature, it must
ignore  any  echomail  query message that has a "%" or "!" as the first
WORD of the Subject field.

Second Reply or Extended Response Request:

   The  FileFind  site  indicates  availablility  of  Second  Reply by
placing the string 'program_name 4d_address' in the From:  field of the
message.
   eg: FROM: FQUERY 1:167/104.0

   When a user replies to a FileFind reply, the message will be to the
FileFind  program  @  {network  address}.  When processing the FileFind
conferences, the FileFind program will treat any message to itself that
includes the site address as a Second Reply Request.

   If  this feature is available, the FileFind program will include up
to  a maximum of 15 files (maximum 12K match list) in it's replies.  If
the  user  wants  a  more  detailed  listing,  he simply replies to the
FileFind  program's  reply.   Only  the system that posted the original
reply  will  respond to that new request.  This second, specific reply,
will  contain  up  to  50  files (32K of matchlist) either including or
SKIPPING the first 15.  These numbers may be replaced by byte limits in
some implementations.

No Second Reply in Designated Reply Echo:

   The Designated Reply Echo method does not allow replies to be made,
because  the FileFind program may not be permitted to scan a Designated
Reply  Echo.  The FileFind program should automatically report up to 50
files  for any requests.  Therefore, the traffic limitation features may
be  disabled for networks that require the FileFind program to reply in
a Designated Reply Echo, and disallow Second Reply in that echo.

Disable Local Messages:

   The  FileFind  program must be able to to disable the processing of
local  messages.  What this means is that the FileFind program will not
process  any messages generated on that FTN site, including messages by
the  sysop  using  an  offline  reader,  or by a site's BBS or off-line
reader users.  This should NOT exclude messages from a site's points.

Limit by Age:

   The  FileFind program must be configurable so that the operator can limit  the  age  of an query message that is acceptable for processing. This  should  be  in  number  of  days.  The  FileFind  program may be configured  to  process all the FileFind requests regardless of how old they are.  Age should never be greater than 365 days.

LinkMGR Support:
   Implementers  may choose to support the LinkMGR proposal for netmail queries  and  commands.   In this proposal, the queries and commands do not  appear  in  the  subject  field but rather, in the the BODY of the message.  The subject field wil contain the LinkMGR password.
   Use of the LinkMGR method allows the user to send multiple commands to the flleFind program.

FILEID.TXT v1.8 by Richard Holler Last Revision 05/05/94

This text file was prepared at the request of the ASP (Association of Shareware Professionals), but the information contained in it may be of value to any shareware author.

**FILE_ID.DIZ INFORMATION**
Basically, the FILE_ID.DIZ file is a straight ASCII text file, distributed inside your distribution archive file along with your program files, which contains a description of your program. This file will be used by most BBS (Bulletin Board System) software for the online file description of your file. We recommend that the FILE_ID.DIZ file be used in all of your distribution archives.

This text file contains a description of the FILE_ID.DIZ file, as well as a description of the recommended distribution archive format.

**WHY SHOULD YOU USE FILE_ID.DIZ?**
The use of this file will insure that the online description of your program will be in your own words (and who better to describe your program than yourself?), and that it will remain the same no matter how many different people upload your file to various BBS systems.

As more and more BBS software makes use of this file, you can be assured that your own description will replace such online descriptions as "Cool Program" or "OK utility, but needs better ..."

Please note that the ASP Hub Network, the Author Direct FDN (File Distribution Network), and the majority of other electronic distribution services *REQUIRE* that a valid FILE_ID.DIZ file be contained in your submitted distribution archive. If your file doesn't contain a valid FILE_ID.DIZ file, then it simply won't be distributed by these services. Furthermore, most BBS sysops will not accept uploads of files which do not contain a valid FILE_ID.DIZ file, so you automatically lose out on that distribution as well.

**DESCRIPTION:**
FILE_ID.DIZ was created by Clark Development for use with their PCBDescribe utility, as a means for BBS callers to upload a file without having to manually type in a file description. It also ensures that the online description is always the same regardless of the number of different BBS systems the file is posted on. It has since been accepted by the BBS industry more-or-less as the "standard" file description source. (The extension of "DIZ" actually stands for "Description In Zip").

NOTE: The FILE_ID.DIZ file *MUST* be named exactly that, and *NOT* something like <file name>.DIZ. It will *ONLY* be used if it is named FILE_ID.DIZ!

The FILE_ID.DIZ file is nothing more than a straight ASCII text file which contains the full description of the archived file containing it. It is used by most popular BBS software to describe your program, rather than using the description supplied by the person that uploaded your file. It should be placed *INSIDE* your distribution archive file.

The BBS software will "look" inside the archive file. If a FILE_ID.DIZ file is found, it will replace any existing online file description with the text contained in FILE_ID.DIZ. It is an excellent method for making sure that your program files are described the way that "you" want them described. Even sysops who's software can't automatically make use of the FILE_ID.DIZ file have found it to be an excellent source for their manually added file descriptions.

**STRUCTURE:**
The file consists of straight ASCII text, up to 10 lines of text, each line being no more than 45 characters long. It should *NOT* contain any blank lines, any form of centering or formatting, or any Hi-ASCII or ANSI characters. (i.e. it should ONLY contain alpha & numeric characters).

We recommended that it consist of 5 basic parts:

1. the proper name of your program
2. the version number
3. the "ASP" identifier (optional, for ASP members)
4. the description separator
4. the description

All of the above parts should be separated by a single "space".

PROGRAM NAME: To set it apart from the rest, it is recommended that you use ALL CAPS for the program name.

VERSION NUMBER: The version number should be in the form of "v12.34".

ASP IDENTIFIER: If you are an ASP author, we recommend that an "<ASP>" identifying mark be added after the version number, to identify your product as an ASP-authored product.

DESCRIPTION SEPARATOR: To separate the actual description text, insert a simple "-" (dash/minus) character after the ASP identifier (or version number, if not using the ASP identifier), and in front of the description text.

DESCRIPTION: You should attempt to FULLY describe your product, including its most important functions and features. Be sure to include anything which will separate your program from it's competition, and make the BBS user want to download your file. Also try to include any hardware or software requirements that your product may have.

You should try to use the first 2 lines of the text to give a basic description of your program. This is helpful for sysops who's BBS software limits them to less than 10 lines, 45 characters. Sysops who are limited to using shorter descriptions can simply use the 1st two lines and truncate the rest. Thus, you can basically still supply your own description for BBS software which does not actually utilize the FILE_ID.DIZ feature.

The remaining lines of text can be used to elaborate on the programs features, enhancements from the prior version, information concerning multi-file sets. Please note that older versions of some BBS software can only use 8 lines of text. It is advisable that you create your FILE_ID.DIZ file so that the file can be truncated to various line lengths without destroying it's usefulness.

**EXAMPLE**
```
MY PROGRAM v1.23 <ASP> - A program which will
do anything for anybody. Will run in only 2k
of memory. Can be run from the command line,
or installed as a TSR. Completely menu-
driven. Version 1.23 reduces the previous 4k
memory requirements, and adds an enhanced
graphical user interface. Also, MY PROGRAM
now contains Windows and DESQview support.
Coming soon - an OS/2 version.
From Do-It-All Software, Inc. $15.00
```

**MULTIPLE DISK INFO**
Please note that if your distribution archive requires multiple archive files, you should create a separate, specific FILE_ID.DIZ file for each archive. This can be utilized to describe the various contents of each archive, and to identify each disk in the set. For example, the FILE_ID.DIZ file for disk #1 could contain:

```
 "MY PROGRAM v1.23 <ASP> Program Executable
  Files - Disk 1 of 2"
  [followed by detailed description text]
```

while the FILE_ID.DIZ file for disk #2 could contain:

```
 "MY PROGRAM v1.23 <ASP> Documentation Files -
  Disk 2 of 2"
```

[followed by more detailed description text]

Optionally, you could also create a "complete" FILE_ID.DIZ file for the first disk, which would fully describe the program in detail, and identify it as Disk 1 of x. Then, for each remaining file in the set, simply include the Program Name, version number, ASP identifier, and the disk number (i.e. "MY PROGRAM v1.23 <ASP> Disk 2 of x").

## ADDITIONAL INFO

Please don't be tempted to use fancy graphic or ANSI sequences in the FILE_ID.DIZ file, as most BBS software will not allow this, and will render your FILE_ID.DIZ file useless. Also, don't be tempted to simply copy your program description file to FILE_ID.DIZ. Attempting to "format" your FILE_ID.DIZ file (i.e line centering, right & left justification, etc) will also cause unexpected results, especially for BBS software which re-formats descriptions to other than 10line/45char.

```
Fred Hill <ASP> has written a freeware utility which interactively creates
a valid FILE_ID.DIZ file. The file is called DIZGEN.ZIP and can be found on
CompuServe (GO IBMBBS, Library 2) as well as on many fine BBS systems. I
highly recommend that you download a copy of this wonderful utility for
creating your FILE_ID.DIZ files.
```

The following is a recommendation for the structure and contents of distribution archives prepared for use on BBS systems.

## DISTRIBUTION DISK RECOMMENDATIONS

The following are recommendations for preparing your program files for distribution to Bulletin Board Systems (BBSs) via the ASP's distribution services, as well as other methods.

Two varieties of program files are defined here:

1) Program files which utilize an "install" utility and self-extracting program archives (later referred to as "Author-Installed Programs").

2) Programs files which do not use install utilities or self-extracting archives (later referred to as "User-Installed Programs").

## AUTHOR-INSTALLED PROGRAMS:

These programs require a bit more work from the author, but will eliminate many user mistakes, especially in programs which require complicated setups.

Most "installation" utility programs will make use of program files which have been "archived" into Self-Extracting (SFX) archives. We will attempt to define which files should be contained in the Self-Extracting archives, and which files should not.

1. Files which should be contained in the self-extracting program file archive:

   a. All program-specific executable files.
   b. Any required configuration and/or data files required by the program.
   c. Program documentation files. Optionally, these may be left
      outside of the self-extracting archive, in order to allow
      them to be viewed/read by the various archive viewing utlities.
   d. Any other program-specific files that are required for the
      operation of the program.

2. The files described above should be compiled into a self-extracting archive file, which will then be extracted by the install utility.

NOTE: the author is required to abide by any distribution requirements specified by the archive utility author, and to

obtain any required distribution rights necessary. Please check to see if distribution rights are required for your archive utility choice.

3. Files which should NOT be contained in the self-extracting program file archive:

    a. The install utility itself (obviously).
    b. The FILE_ID.DIZ file. (described in detail in the section
       preceding this one)
    c. Any distribution/information files, such as VENDOR.TXT,
       SYSOP.TXT, etc.
    d. Any description or information file, such as DESCRIBE.TXT.
    e. A user file (such as README.1ST), which should explain how
       to use the install utility, what the user should expect
       during the installation, and any preparation that the user
       should make prior to the installation. This file might also
       contain a brief description of your program, in case the user
       is able to read the documentation files in the distribution
       archive prior to downloading (many BBS systems offer this
       ability to the user).

4. The actual distribution archive file (described below) should then contain the install utility, the self-extracting program archive, and the files described in #3 above.

**USER-INSTALLED PROGRAMS:**
This type of distribution archive is much simpler than the Author-Installed variety. It should simply be an archive file, containing all of the files for the program described above.

Since this type of program requires the user to do all of the installation manually, it should contain very specific and detailed information regarding the installation requirements (such as INSTALL.TXT).

**THE DISTRIBUTION ARCHIVE FILE:**
The actual distribution archive file should merely be an archive file containing the files described above. For BBS distribution, this archive should be of the standard archive format, and -NOT- a self-extracting archive.  Many sysops will not allow self-extracting archives, and most BBS software will not allow self-extracting archives to be uploaded.

There are many popular archive utilities available, such as PKZIP, LHA, LHARC, ARJ, etc. Most BBS systems are capable of handling archives in virtually any format. However, you should be aware that most BBS systems will convert your archive format to the format of choice by the sysop. By following the methods described above, this conversion process should not affect your program, or any self-extracting files which are contained within your distribution archive file.

You should also retain the default archive file extension defined by the archive utility. For example, PKZIP uses a ".ZIP", LHARC uses "LZH", etc. Changing the file extension may cause the BBS software to delete your file because it doesn't recognize the format.

For the actual file name for your distribution archive, it is recommended that the program file name be limited to 6 characters to represent the program's name (i.e. MYPROG could represent "My Program"). This should be followed by 2 numeric digits which will represent the version number of your release. Even if this is your initial release it should include the version number in the file name (i.e. MYPROG10.ZIP would indicate the program called "My Program" version 1.0).

Please note that CompuServe limits file names to only 6 characters. By limiting the file "name" to 6 characters, you will easily be able to rename the archive for CompuServe uploading by simply removing the 2-digit version identifier, to make the file compatible with CompuServe libraries.

By including the 2-digit version number in the archive file name, it will be very easy for both the user and the sysop (and yourself) to identify older versions of your program.

**MULTIPLE DISTRIBUTION ARCHIVES**

At one time, it was recommended that your final distribution archive not be larger than 350k, so that it would fit on a single 360k floppy disk and still leave room for any distribution files necessary for Disk Vendors. (i.e. Disk Vendors will often include their own GO.BAT file, or other various small files to help their customers install the software). This limitation is slowly falling by the wayside as more and more computer systems have 3.5" floppy disk drives as standard.

If your program is large enough to require more than one distribution archive, it is recommended that your file name be limited to 5 characters rather than 6 as described above. Following the 5-character name should be the same 2-digit version number. Then, append a single "letter" to identify the disk (i.e. MYPGM10A.ZIP, MYPGM10B.ZIP, etc.). For uploading to CompuServe, these file names may then be shortened to 6 characters by removing the version identifiers (i.e. MYPGMA.ZIP, MYPGMB.ZIP). However, for CompuServe it is recommended that you simply create a single distibution file, and eliminate the multi-part file set.

If your program requires multiple distribution archives, -BE SURE- to create separate FILE_ID.DIZ files for each distribution archive. Also, each FILE_ID.DIZ file should contain disk number information pertaining to each individual archive (i.e. Disk 1 of 3, Disk 2 of 3, etc.).

**THE DISTRIBUTION DISK**

It is recommended that your distribution disk simply contain a ZIPd version of your product. However, If you choose to supply "unarchived" files on a distribution disk for Disk Vendor use, it is _VERY_ important that you specify in your documentation a suggested archive file name, so that BBS sysops can create archived files with the proper author-specified file names. This information should be contained in your SYSOP.TXT (or VENDOR.TXT) file. If you don't supply a suggested archive file name, the sysops will be forced to create the name themselves, thus you may end up with thousands of versions of your products on BBS systems all over the world, but all with different file names.

Please note that the ASP Hub Network, and nearly every other electronic distribution service *REQUIRE* that your files be submitted as an archived file, using the ZIP format. Also note that many BBS sysops will not go to the trouble of ZIPing your unarchived files for you. If you don't supply them with an archived distribution version of your product, it might not get distributed by BBSs.

If you supply your own disk labels, it is recommended that the ASP logo, or at least the initials "ASP" be included on the label, so that anyone can immediately identify your disk as an ASP member's software.

**SUMMARY**

Your distribution disk should now be ready to submit to the various BBSs, distribution services, and Disk Vendors.

You may choose to create a separate distribution disk for use by BBSs and Disk Vendors. However, if you follow the above steps in preparing your distribution archive file, a separate "Disk Vendor" disk is probably not necessary. The majority of disk vendors will be able to accept your distribution file/disk if it is prepared in the above described format.

# How to setup an FTP server to work with MBSE BBS.

In order to let MBSE BBS and your FTP server to both function together you must organize a special file structure. Note that even if you don't setup an FTP server you must still create a structure like this for the fidonet mailer, if you don't, **mail and files will get lost!** This description is written for ProFTPD, on your distribution there may be another ftpd installed. Read the section for the webserver as well.

The file structure I used is as follows:

```
/opt/mbse/ftp/pub/dos_util/dos_4dos      - Public download areas
            |   |   |           /dos_disk
            |   |   |           /dos_file
            |   |   /virnet/mcafee
            |   |   |       /win16
            |   |   |       /win32
            |   |   /css/files.css        - Stylesheet for http download
            |   /incoming                 - FTP public upload.
            /var/bso/outbound             - Your default outbound
                |   /outbound.009         - Outbound Zone 9
                /inbound                  - Inbound directory
                /private/upload           - Non-public download areas
                |       /sysop
                |       /logfiles
                /tic_queue                - Queue for .tic files.
```

To let this work you need to change the default anonymous directory of the ftp server to /opt/mbse/pub. As root, edit /etc/passwd with the **vipw** command. Look for the entry of the ftp user and change his homedirectory to /opt/mbse/ftp. You may also use the command **usermod -d /opt/mbse/ftp ftp** to do the same. For a start the default configuration file for ProFTPD will do fine. If you want to let users upload into the incoming directory, you need to change /etc/proftpd.conf file to allow that.

If another DOS/Windows style mailer has access to your MBSE outbound you must set the DOS path and Unix path in **mbsetup** (1.4.12 and 1.4.13) to **"m:"** and **"/opt/mbse"**. Note that to get forwarding of .tic files to work the **tic_queue** must be a subdirectory of "/opt/mbse" too. You could actually use any drive letter for the DOS path.

This means that a fidonet file attach from the dos_4dos public download directory shall get the subject "M:\FTP\PUB\DOS_UTIL\DOS_4DOS\COMMAND.ZIP". Only use this if you need it!

As you can see, anonymous ftp users can't get to the mail, non-public downloads etc. Normally, your BBS users have unix accounts and will be able to do a ftp login and access any directory on your system. Because the bbs users have **mbsebbs** as their shell and this shell is not in the file **/etc/shells** the ftp daemon will not let the bbs users in. So even your own bbs users must login as anonymous to get files from the ftp server.

The following is a list of file permissions when using ProFTPD:

```
Directory                       owner group mode perms
------------------------------- ----- ----- ---- ----------
/opt/mbse                       mbse  bbs   0775 drwxrxxr-x
/opt/mbse/ftp                   root  root  0755 drxxr-xr-x
/opt/mbse/ftp/pub               mbse  bbs   0755 drwxr-xr-x
/opt/mbse/ftp/incoming          ftp   ftp   0755 drwxr-xr-x
```

Note that all subdirectories under ../pub also must be owned by **mbse** and group **bbs** and have at least mode 755 as long as it are real bbs subdirectories. The bbs will maintain these directories automatic and must have the rights to do so.

In the /opt/mbse/ftp/etc/group file, add the group bbs so that your directory listings give the proper group name instead of a number.

If you want to increase the download counters when files are downloaded via ftp you must make sure that the ftp daemon logs the downloads to a **xferlog** file, for example /var/log/xferlog. This logfile must be readable by user mbse. Then in **mbsetup** menu 1.13.5 enter full file name and path to this log file. The **mball** program will parse this file and increase the download counters for the files that are downloaded from the bbs.

# FSP-???? R1 - Integration of nodelist IP-Nodes (FTS-0005)

Author:        Lothar Behet, 2:2446/301
Revision Date:  25 October 1998
Expiry Date:

Contents:
1. Required fields according to FTS-0005, basic flags for ip-nodes
2. Optional extensions
3. Addendum
-------------------------------------------------------------------

## 1.  Description of the nodelist format

Every node entry contains the following 8 fields:

keyword,node_number,node_name,location,sysop_name,
phone_number,baud_rate,flags

Certain fields have defined values according to FTS-0005.

1.1.    Implementation for IP-connectivity
    Because of the limited characterset in the phone_field and
    to avoid any misinterpretion by conventional dialing, the
    ip-specific address-information is entered in another field
    and there are additional flags required.

1.1.1.  Field #1 (keyword) is PVT for an ip-only node without
    conventional phone number related connectivity. In this
    case, the phone field contains "-Unpublished-" according
    to FTS-0005.

1.1.2.  Field #2 (node_number) contains the node number within his
    net and zone.

1.1.3.  Field #3 (node_name) is used for the FQDN (Fully Qualified
    Domain Name) or the ip-address.

1.1.4.  Field #4 (location) contains the geographical location of
    the node. While some nets/regions cannot supply their
    ip-only nodes with a adequate link, these nodes may be
    collected in a seperate net or region, until their original
    net/region support additional ip-connectivity. This special
    net/region is definitely a temporary solution for routing
    within a region or zone!

1.1.5.  Field #5 (sysop_name) represants the name of the system
    operator.

1.1.6.  Field #6 (phone_number) contains the phone_number for
    conventional connectivity. In case of an ip-only node
    it must contain "-Unpublished-".

1.1.7.  Field #7 (baud_rate) contains the maximum baud rate for
    conventional connectivity or 300 in case of an ip_only node.

1.1.8.  Field #8 (flags) represents operational definitions for the
        node.
        Note that these are user flags.
        The ip-flags consist of two parts:
        A basic transport and an optional non-standard port,
        seperated by a colon.
        The default port may be omitted, but is listed as optional
        parameter in this document. In some cases, two flag names
        are mentioned:
        The second one is supported by some software nowadays, but
        these values may conflict with other programs, which not
        completely decode the length of each individual flag (i.e.
        TELN conflicts with the T-flag for online-time)
        Additional flags for ip-nodes are:

1.1.8.1.  IBN[:24554] (Argus: BND[:24554])
          BinkP protocol

1.1.8.2.  IFC[:60179]
          Raw protocol as used by ifcico

1.1.8.3.  ITN[:23] (Argus: TEL[:23])
          Telnet protocol. Some variants of ifcico support Telnet
          on port 60177, which should be added as additional flag
          ITN:60177.

1.1.8.4.  IVM[:3141]
          Vmodem protocol

1.1.8.5.  IP
          General flag for special protocol specifications, if the
          flags conforming to 1.1.8.1. to 1.1.8.4. are not relevant.

1.1.9.  Comments on the proposed nodelist flags
        The additional flagnames in () are supported at this moment
        by Argus, based on the use in z2r50. But the TEL[NET]-flag
        stays in conflict with the generally in all zones and
        regions used T-flag (online time according to FSC-0062).

## 2.  Optional extensions for future use

While the above mentioned flags (1.1.8.1 to 1.1.8.4) define a
minimum set of operational flags for ip-nodes, several additions
are already foreseeable at this moment.

2.1.   Additional sessions_handshake parameters
       There is at least one program, which supports several
       transport protocols according to chapter 1.1.8. on a
       single port. If other programs should imitate this habit,
       then the following extension to the flag suite 1.1.8.
       (transport[:port[:handshake]])is advised:

2.1.1.  FTS-0001 session handshake:   1
2.1.2.  Yoohoo session handshake  :   Y

2.1.3.  EMSI sessions handshake  :  E
2.1.4.  BinkP sessions handshake  :  B


2.2.    Non-handshaking protocols
        While the definitions until this chapter describe direct
        handshaking sessions with optional password authentification,
        there are several other methods for the tunneling of fidonet
        data via the internet available.
        The setup of these connections does not rely on the nodelist
        (at this moment of writing), but we can think of standard
        setup procedures to use the nodelist for configuration of
        this additional transport methods.
        Therefore the following flags 2.2.1. to 2.2.4. are advised
        for at least informational purpose.


2.2.1.  IFT
        FTP (File Transfer Protocol)


2.2.2.  ITX
        TransX, an Email based variant


2.2.3.  IUC
        Uuencoded packet (one packet per message)


2.2.4.  IEM
        Email based (generally, without exact specification at
        this moment)


## 3. Addendum


This proposal is based on a maximum compatibility to generally used definitions and standards within the Fidonet
community. Future developments might make additions necessary, if they can not be expressed with the existing
set of flags as defined by this FSP.

file name....: JAM-001
Rev.........: 001
Dated.......: 93-07-01
Status .....: Released
Subject.....: JAM message base proposal
Author......: Joaquim Homrighausen
Co-Authors..: Andrew Milner, Mats Birch, Mats Wallin

```
            --------------------------------------------------------------------
                                 JAM(mbp)
                 The Joaquim-Andrew-Mats Message Base Proposal
            --------------------------------------------------------------------
                 Copyright 1993 Joaquim Homrighausen, Andrew Milner,
                          Mats Birch, Mats Wallin.
                            ALL RIGHTS RESERVED.
            --------------------------------------------------------------------
```

==================================================================
Definitions and general notes
--------------------------------------------------------------------
CURRENTREV              1

JAM             The JAM message base format.

CRC             Cyclic Redundancy Check. All CRC values
                calculated on strings must assume that the
                data within the string has been converted
                to lowercase (A-Z = a-z).

CRC-32          32-bit CRC (as used in the Zmodem file
                transfer protocol) value. The polynom for
                a CRC-32 is edb88320H and the CRC-32 seed
                is -1L (ffffffffH).

uchar           Unsigned 8-bit value

ushort          Unsigned 16-bit value

ulong           Unsigned 32-bit value

UNIX date       An ulong representing the number of seconds
                since midnight, January 1, 1970. UNIX-style
                dates is the only form of time stamps used
                in JAM (1).

Message #       The physical record number within the index
                file is used as a message number. The
                lowest message number is one (1) and the
                highest message number is 4294967295
                (ffffffffH).

FTN             FidoNet Technology Network

FTS             FidoNet Technical Standard

(1) All timestamps created locally (i.e. those not imported from
    other systems) are stored in local time.


=======================================================================
Files
---------------------------------------------------------------------
Each conference is made up from four files. How and where these files
are stored and named is implementation dependant. The only file with
a fixed minimum size is the .JHR (header data) file. It has a 1024-
byte block used to hold information about a specific message area as
described later.

file name.JHR - Message header data
file name.JDT - Message text data
file name.JDX - Message index
file name.JLR - Lastread information

A future revision of JAM may also include a file that holds the
following three items:

  - The highest assigned user number
  - The last generated message ID

- A global conference list with the conference name, description, and physical location of the message base.

==================================================================
.JHR file header
------------------------------------------------------------------
Below is the format of the 1024-byte record at the beginning of all
.JHR files. The first actual message header starts at offset 1024 in
the .JHR file.

```
FixedHeaderInfoStruct:
    ulong   Signature;       // <J><A><M> followed by <NUL>
    ulong   datecreated;     // Creation date
    ulong   modcounter;      // Update counter
    ulong   activemsgs;      // Number of active (not deleted) msgs
    ulong   passwordcrc;     // CRC-32 of password to access
    ulong   basemsgnum;      // Lowest message number in index file
    uchar   RESERVED[1000];  // Reserved space
end;
```

MODCOUNTER must be incremented and updated on disk each time an
application modifies the contents of the message base. When it
reaches ffffffffH, it wraps to zero.

------------------------------------------------------------------
BaseMsgNum                 Lowest message number in index file
------------------------------------------------------------------
This field determines the lowest message number in the index file.
The value for this field is one (1) when a message area is first
created. By using this field, a message area can be packed (deleted
messages are removed) without renumbering it. If BaseMsgNum contains
500, the first index record points to message number 500.

BaseMsgNum has to be taken into account when an application
calculates the next available message number (for creating new
messages) as well as the highest and lowest message number in a
message area.

------------------------------------------------------------------
????????.JHR                              Message headers
------------------------------------------------------------------
The .JHR file contains none or more Header records. Each record
define one message and contains information about the message and its
text (if any). The Header record is of variable length. The layout of
the Header record follows.

```
MessageHeader:
    MessageFixedHeader:
        ulong  Signature;     // <J><A><M> followed by <NUL>
        ushort Revision;      // Revision level of header         (1)
        ushort ReservedWord;  // Reserved for future use
        ulong  SubfieldLen;   // Length of subfields              (2)
        ulong  TimesRead;     // Number of times message read
        ulong  MSGIDcrc;      // CRC-32 of MSGID line             (3)
        ulong  REPLYcrc;      // CRC-32 of REPLY line             (3)
        ulong  ReplyTo;       // This msg is a reply to..
        ulong  Reply1st;      // First reply to this msg
        ulong  Replynext;     // Next msg in reply chain
        ulong  DateWritten;   // When msg was written
        ulong  DateReceived;  // When msg was read by recipient
```

```
        ulong  DateProcessed;// When msg was processed by tosser/
                             // scanner
        ulong  MessageNumber;// Message number (1-based)
        ulong  Attribute;    // Msg attribute, see "Msg Attributes"
        ulong  Attribute2;   // Reserved for future use
        ulong  Offset;       // Offset of text in ????????.JDT file
        ulong  TxtLen;       // Length of message text
        ulong  PasswordCRC;  // CRC-32 of password to access message
        ulong  Cost;         // Cost of message
      end;
      SubField1              // Extra fields as defined below
      .
      .
      SubFieldXX
  end;
```

(1) This field is intended for future revisions of the specifications
    to allow the use of a different fixed-length binary message
    header. The current revision level is one (1).

(2) The SubfieldLen field is set to zero (0) if the header does not
    have any subfield data. I.e. the length of the binary header is
    not included in this field.

(3) When calculating the CRC-32 of the MSGID and REPLY lines, the
    text ^aMSGID: and ^aREPLY: should be removed as well as all
    leading and trailing white space characters.

The SubField structure is made up of an ID, a length specifier, and
a block of data. Zero or more subfields may follow the fixed-length
binary header. SubFields are not stored in any specific order and
are not terminated by any specific character unless otherwise
specified.

```
  SubField:
      ushort  LoID;          // Field ID, 0-65535
      ushort  HiID;          // Reserved for future use
      ulong   datlen;        // Length of buffer that follows
      uchar   Buffer[datlen]; // DATLEN bytes of data
  end;
```

------------------------------------------------------------------
Defined LoID codes
------------------------------------------------------------------

ID=0, Name=OADDRESS

A network address. This is used to specify the originating address.
More than one OADDRESS field may exist. DATLEN must not exceed 100
characters. For a FidoNet-style address, this field must follow the
ZONE:NET/NODE.POINT@DOMAIN format where .POINT is excluded if zero
and @DOMAIN is excluded if unknown.

ID=1, Name=DADDRESS

A network address. This is used to specify the destination address.
More than one DADDRESS field may exist (e.g. carbon copies). DATLEN

must not exceed 100 characters. For a FidoNet-style address, this
field must follow the ZONE:NET/NODE.POINT@DOMAIN format where .POINT
is excluded if zero and @DOMAIN is excluded if unknown.


ID=2, Name=SENDERNAME

The sender (author) of the message. DATLEN must not exceed 100
characters.


ID=3, Name=RECEIVERNAME

The recipient of the message. DATLEN must not exceed 100 characters.


ID=4, Name=MSGID

Used to store the message identification data. All data not relevant
to the actual ID string, including leading and trailing white space
characters should be removed. DATLEN must not exceed 100 characters.


ID=5, Name=REPLYID

Used to store the message reply data. All data not relevant to the
actual reply string, including leading and trailing white space
characters should be removed. DATLEN must not exceed 100 characters.


ID=6, Name=SUBJECT

The subject of the message. DATLEN must not exceed 100 characters.
Note that this field may not be used for FidoNet-style file attaches
or file requests.


ID=7, Name=PID

Used to store the FTN PID kludge line. Only the actual PID data is
stored and ^aPID: is stripped along with any leading and trailing
white space characters. DATLEN must not exceed 40 characters.


ID=8, Name=TRACE

This is also referred to as ^aVia information in FTNs. It contains
information about a system which the message has travelled through.
The format of the field is <YYYYMMDDHHMMSS><Network address> where:

  YYYY is the year (1992-9999)
    MM is the month (01-12)
    DD is the day (01-31)
    HH is the hour (00-23)
    MM is the minute (00-59)

SS is the second (00-59)

The timestamp is stored in ASCII (0-9) characters. The network address is the address of the system. It is expressed in ASCII notation in the native format of the forwarding system.

ID=9, Name=ENCLOSEDFILE

A file attached to the message. Only one file name may be specified per subfield. No wildcard characters are allowed. If this subfield is present in a message header, the ATTRIBUTE must include the MSG_FILEATTACH bit.

ID=10, Name=ENCLOSEDFILEWALIAS

Identical to ENCLOSEDFILE with the exception that the file name is followed by a <NUL> (00H) and an alias file name to be transmited to the remote system in place of the local name of the file.

ID=11, Name=ENCLOSEDFREQ

A request for one or more files. Only one filemask may be specified per subfield. If the filemask contains a complete path, it is to be regarded as an update file request. If this subfield is present in a message header, the ATTRIBUTE must include the MSG_FILEREQUEST bit. To indicate that a password is to be transmitted along with the request, a <NUL> (00H) character followed by the password is appended. E.g. SECRET*.*<NUL>MYPASSWORD.

ID=12, Name=ENCLOSEDFILEWCARD

One or more files attached to the message. Only one file name may be specified per subfield. Wildcard characters are allowed. If this subfield is present in a message header, the ATTRIBUTE must include the MSG_FILEATTACH bit.

ID=13, Name=ENCLOSEDINDIRECTFILE

One or more files attached to the message. The file name points to an ASCII file with one file name entry per line. If alias file names are to be used, they are specified after the actual file name and separated by a <NUL> (00H) character, e.g. C:\MYFILE.LZH<NUL>NEWS. Wildcard characters are not allowed.

ID=1000, Name=EMBINDAT

Reserved for future use.

ID=2000, Name=FTSKLUDGE

An FTS-compliant "kludge" line not otherwise represented here. All
data not relevant to the actual kludge line, including leading and
trailing white space and ^A (01H) characters should be removed.
DATLEN must not exceed 255 characters. The FTS kludges INTL, TOPT,
and FMPT must never be stored as separate SubFields. Their data must
be extracted and used for the address SubFields.

ID=2001, Name=SEENBY2D

Used to store two-dimensional (net/node) SEEN-BY information often
used in FTN conference environments. Only the actual SEEN-BY data is
stored and ^aSEEN-BY: or SEEN-BY: is stripped along with any leading
and trailing white space characters.

ID=2002, Name=PATH2D

Used to store two-dimensional (net/node) PATH information often used
in FTN conference environments. Only the actual PATH data is stored
and ^aPATH: is stripped along with any leading and trailing white
space characters.

ID=2003, Name=FLAGS

Used to store the FTN FLAGS kludge information. Note that all FLAG
options that have binary representation in the JAM message header
must be removed from the FLAGS string prior to storing it. Only
the actual flags option string is stored and ^aFLAGS is stripped
along with any leading and trailing white space characters.

ID=2004, Name=TZUTCINFO

Time zone information. This subfield consists of four mandatory
bytes and one optional. The first character may be a plus (+) or a
minus (-) character to indicate a location east (plus) or west
(minus) of UTC 0000. The plus character is implied unless the first
character is a minus character. The following four bytes must be
digits in the range zero through nine and indicates the offset in
hours and minutes. E.g. 0100 indicates an offset of one hour east of
UTC.

```
--------------------------------------------------------------------
Message attributes
--------------------------------------------------------------------
MSG_LOCAL       (0x00000001L)   // Msg created locally
MSG_INTRANSIT   (0x00000002L)   // Msg is in-transit
MSG_PRIVATE     (0x00000004L)   // Private
MSG_READ        (0x00000008L)   // Read by addressee
MSG_SENT        (0x00000010L)   // Sent to remote
MSG_KILLSENT    (0x00000020L)   // Kill when sent
MSG_ARCHIVESENT (0x00000040L)   // Archive when sent
MSG_HOLD        (0x00000080L)   // Hold for pick-up
```

```
MSG_CRASH        (0x00000100L)   // Crash
MSG_IMMEDIATE    (0x00000200L)   // Send Msg now, ignore restrictions
MSG_DIRECT       (0x00000400L)   // Send directly to destination
MSG_GATE         (0x00000800L)   // Send via gateway
MSG_FILEREQUEST  (0x00001000L)   // File request
MSG_FILEATTACH   (0x00002000L)   // File(s) attached to Msg
MSG_TRUNCFILE    (0x00004000L)   // Truncate file(s) when sent
MSG_KILLFILE     (0x00008000L)   // Delete file(s) when sent
MSG_RECEIPTREQ   (0x00010000L)   // Return receipt requested
MSG_CONFIRMREQ   (0x00020000L)   // Confirmation receipt requested
MSG_ORPHAN       (0x00040000L)   // Unknown destination
MSG_ENCRYPT      (0x00080000L)   // Msg text is encrypted        (1)
MSG_COMPRESS     (0x00100000L)   // Msg text is compressed       (1)
MSG_ESCAPED      (0x00200000L)   // Msg text is seven bit ASCII  (1)
MSG_FPU          (0x00400000L)   // Force pickup
MSG_TYPELOCAL    (0x00800000L)   // Msg is for local use only
MSG_TYPEECHO     (0x01000000L)   // Msg is for conference distribution
MSG_TYPENET      (0x02000000L)   // Msg is direct network mail
MSG_NODISP       (0x20000000L)   // Msg may not be displayed to user
MSG_LOCKED       (0x40000000L)   // Msg is locked, no editing possible
MSG_DELETED      (0x80000000L)   // Msg is deleted
```

(1) This revision of JAM does not include compression, encryption, or
    escaping. The bits are reserved for future use.

====================================================================
????????.JDT                        Message text
--------------------------------------------------------------------

The .JDT file contains the text of messages. The text is stored as an
stream of seven or eight bit ASCII data. Allowed characters in the
text are 00H through ffH unless the header ATTRIBUTE field has the
MSG_ESCAPED bit enabled, in which case the legal range of data is 20H
through 7eH.

An escaped character is stored as \<hex> where <hex> is the two digit
hexadecimal ASCII value of the character. A single \ is stored as \\
or \5C. The case of the hexadecimal ASCII value is irrelevant, i.e.
5c is treated as 5C.

====================================================================
????????.JDX                        Message index
--------------------------------------------------------------------

The .JDX file is used to quickly locate messages for any given user
name or to locate a message with a specific number. Each record in
the file consists of two ulongs. The first ulong holds the CRC-32 of
the recipient's name (lowercase), the second ulong holds the
physical offset of the message header in the .JHR (header) file.

The record number (+BaseMsgNum) within the .JDX file determines a
message's number.

If both ulongs are -1 (ffffffffH), there is no corresponding message
header.

====================================================================
????????.JLR                        Lastread storage
--------------------------------------------------------------------
The .JLR file is used to maintain a user's position within a message
area. The layout of the "lastread" record follows. One record per
```

user is required.

```
  LastRead:
     ulong   UserCRC;        // CRC-32 of user name (lowercase)   (1)
     ulong   UserID;         // Unique UserID
     ulong   LastReadMsg;    // Last read message number
     ulong   HighReadMsg;    // Highest read message number
  end;
```

(1) The functions to convert a string to lowercase characters that
    are provided in the API will only convert characters A-Z (into
    a-z). It is required that this convention is followed by all
    applications.

The UserID field is a unique number for each user. If the "lastread"
record is deleted, UserCRC and UserID are both set to -1
(ffffffffH). An application may not depend on any specific order in
the .JLR file. A user's "lastread" record may appear anywhere in the
file and must be searched for when retrieving it and when storing an
updated record.

===================================================================
Updating message headers
-------------------------------------------------------------------
If a header record grows after is has been retrieved from the .JHR
file, it must be appended to the end of the .JHR file since it would
overwrite the following header otherwise. The .JDX file must be
properly updated to indicate the new location of the header record.
The old header record must be changed to indicate that it has been
deleted by setting the MSG_DELETED bit in the Attribute field and the
TextLen field to zero (to prevent a maintenance program from removing
the message text that is now pointed to by another header).

===================================================================
Message base sharing and locking
-------------------------------------------------------------------
To allow several programs to access the message base at any given
time, region locking is used to protect the message base from being
corrupted during updates.

When an application needs to write to any of the message base files,
it must first attempt to lock the first byte of the .JHR (header)
file. If the lock call fails, the application must either fail or
attempt to lock the file again. The message base files may under no
circumstances be updated if the application cannot successfully lock
the .JHR file.

Note that data acquired (read) from the message base may not be used
when writing data to the message base, unless the application has
maintained a lock of the message base from the time the data was
acquired or the MODCOUNTER field is the same as when the data was
acquired.

The application must open the files in shareable (DENYNONE) read/
write or readonly mode. The only exception to this is an application
that requires exclusive access to the message base, such as a message

base maintenance utility, it should open the files in non-shareable (DENYALL) read/write mode.

==================================================================
Reply threads and linking
------------------------------------------------------------------
JAM introduces a new reply link pointer, not commonly used today. This section is an attempt to describe how reply threads, reply linking, and this new reply link pointer is implemented in JAM.

One of the major differences is that reply threads in JAM are not based on similar or identical subjects of messages since this method does not allow for proper reply threads.

The method used in JAM is based on the immediate relation between any given message and direct replies to it. This is supported by many message editors by using the MSGID and REPLY FTS kludge fields. These are common, although expressed differently, in messages not based on FidoNet technology, such as RFC-822. The obvious advantages include allowing a program to easily find the original message to a reply, and to find all replies to any given message.

The reply thread information consists of three fields: ReplyTo, Reply1st, and ReplyNext. The reason for three fields, as opposed to just two, is that with two fields, it is only possible to keep track of the original message of a reply (which is sufficient) and one reply to any given message (which is not sufficient). With three fields, it is possible to maintain a thread of any number of replies to any given message.

In the description of the different fields below, the following messages and message numbers will be referred to:

```
  1 -> 2 -> 4 -> 5
  :   :
  :    +--> 8
  :
  +--> 3 -> 7
  :
  +--> 6
```

Message number two, three, and six are replies to message number one.
Message number four and eight are replies to message number two.
Message number seven is a reply to message number three.
Message number five is a reply to message number four.

------------------------------------------------------------------
ReplyTo
------------------------------------------------------------------
This field holds the number of the message that this message is a reply to. In the example above, the ReplyTo field would contain the following values:

Message number one would contain zero; message number two, three, and six, would contain one; message number four and eight would contain

two; message number seven would contain three, and message number
five would contain four.


---------------------------------------------------------------------
Reply1st
---------------------------------------------------------------------
This field holds the number of the first message that is a reply to
this message. In the example above, the Reply1st field would contain
the following values:

Message number one would contain two, message number three would
contain seven, and message number four would contain five. All other
messages would contain zero.


---------------------------------------------------------------------
ReplyNext
---------------------------------------------------------------------
This field is used to create the actual message thread or chain. In
the event that there is more than one reply to any given message, it
is necessary to maintain a thread of all the replies; this is due to
the fact that the original message can only hold information about
the first reply (the Reply1st field) to it.

The first reply (which the original message's Reply1st field holds),
has its ReplyNext field pointing to the second reply, the second
reply's ReplyNext field poinst to the third reply, and so on.

In the example above, the ReplyNext field would contain the following
values:

Message number two would contain three, message number three would
contain six, and message number four would contain eight. All other
messages would contain zero.


====================================================================
Contacts
---------------------------------------------------------------------
Joaquim Homrighausen             Telefax: +352 316 702
389, route d'Arlon                Modem: +352 316 702
L-8011 Strassen                  eMail: 2:270/17@fidonet
Luxembourg                        joho@abs.lu

Andrew Milner                    Telefax: +352 251 621
9a, Boulevard Joseph II           Modem: +352 251 621
L-1840 Belair                    eMail: 2:270/18@fidonet
Luxembourg                        andrew@fido.lu

Mats Wallin                      Telefax: +46 8 6453285
F"rskottsv„gen 11                 Modem: +46 8 6453882
S-126 44 H„gersten               eMail: 2:201/329@fidonet
Sweden                                mw@fido.lu

# Binkly style outbound documentation for MBSE BBS.

The MBSE BBS outbound directory structure is BinkleyTerm compatible, with domains and point subdirectories (full 5d). There are separate "protected" and "unknown" inbound directories for incoming sessions. Files received during outbound sessions are always placed in the "protected" inbound directory. Only the "protected" inbound directory is processed automatic.

.pol    Poll flag, is handled as crash immediate, the length is always 0 bytes.

    Flow files are files with the full pathnames to the files to send
    on disk. Names are translated by MBSE BBS to full DOS file names and
    paths depending on your setup.
    If you use it then it is importand that you think about the directory
    structure to use. See also the documentation about the setup of the
    [ftp server](#)
    The file names may be prepended with a special character:
    #    = Truncate file after sent.
    - or ^ = Kill file after sent.
    @    = Leave file after sent, this is the default. Upto version
       0.61.1 this was empty.

.flo    Normal flow file (contains complete file names to send).
.clo    Crash flow file.
.hlo    Hold flow file.
.dlo    Direct flow file, overrides CM flag.

    The following are .pkt files, during the mail session they will be
    renamed to nnnnnnnn.pkt with an unique name and added to the spool
    file. Messages can allways be added to the outbound as long as the
    node isn't locked.

.out    Normal .pkt file.
.cut    Crash .pkt file.
.hut    Hold .pkt file.
.dut    Immediate .pkt file.

.req    Request file. Contains file names in ascii with <cr><lf>.

.su0    Arcmail bundles, the last digit may be any digit or letter.
.mo0
.tu0
.we0
.th0
.fr0
.sa0

.sts    Node status file created by mbcico. These are data files containing
    three values:
    1. 'time', this is the time when a new call to this node is allowed.
      (in time_t format).
    2. 'retries', is the number of retries to try to connect that node. This
      field is zeroed when the call succeeds or when that node calls in.
      It is also zeroed when a new poll is created. Currently, mbcico stops
      calling a node if the counter is higher then 30.

3. 'code', is the return code of the last attempt.

0    - Successfull call
1    - No dialout port available
2    - No CONNECT or TCP connect failed
3    - Could not reset the modem
4    - System is locked
5    - Retry time not reached?
6    - Fatal error in nodelist lookup
7    - Call prohibited by config options
8    - Phone number unavailable
9    - No free matching port
10    - Unused
11..29 - Session (handshake) errors.

This file is **not** compatible with the .sts files created by **ifcico**.

.spl    Spool file, created by mbcico.

.bsy    Busy file, for locking nodes. The 'pid' of the process who locked that
        node is inserted into this file. All programs of the MBSE BBS package
        (and ifcico package) check if the pid exists if a .bsy file is found.
        If there is no pid found, the lock is a stale lock and is removed.

# semaphore files with MBSE BBS.

The directory $MBSE_ROOT/var/sema is the hard coded semaphore directory where all semaphore's must be created, tested and removed. When the system is booting, the init script will erase all semaphore's just before the BBS is started. This description is valid from MBSE BBS v0.33.19 and newer.

zmh           Purpose: to mark the state of Zone Mail Hour.
              Created by "mbtask" at the start of Zone Mail Hour.
              Removed by "mbtask" at the end of Zone Mail Hour.

upsalarm      Purpose: Signal that the system is running on battery power.
              Created and removed by UPS software.
              Checked by mbtask to suspend processing.
              Checked by mbfido to stop processing.

upsdown       Purpose: Signal that the system will go down on low battery.
              Created and removed by UPS software.
              Checked by mbtask to go down.
              Checked by several scripts and "mbstat wait".

newnews       Purpose: Signal that there are new articles on the news server.
              Checked by mbtask to start news processing.
              Removed by mbtask as soon as it is detected.

mailout       Purpose: Signal that there is mail posted in the message base.
              Checked by mbtask to start scan the message base.
              Removed by mbtask as soon as it is detected.

mailin        Purpose: Signal that there is new mail in the inbound.
              Checked by mbtask to start the tosser.
              Removed by mbtask as soon as it is detected.

scanout       Purpose: Signal that the outbound must be rescanned.
              Checked by mbtask to check the outbound.
              Removed by mbtask as soon as it is detected.

do_inet       Purpose: Signal that there are node(s) to be called via the
              internet. Usefull for dialup systems to check whether a
              connection to the internet is needed.
              Created and removed by mbtask.

is_inet       Purpose: Signal that the internet is available. Usefull for
              dialup systems to check when the internet connection is up
              or down after starting or stopping ppp.
              Created and removed by mbtask.

mbtask.last   Purpose: A timestamp created and touched by "mbtask" every
              minute so you can check it is running.

# System load and the usleep() call.

At some time when developing MBSE BBS I decided that background utilities didn't need full speed to do their jobs. BBS utilities under DOS needed to run as fast as possible because you needed to bring the bbs down to run these programs and users couldn't login during that time.

Starting with mball, the allfiles creator, I inserted code that does usleep(1) after each 5 processed files. The 1 microsecond is not really the time the program pauses, it's probably a lot longer. I think this depends on the hardware type, (Intel, Sparc, Alpha etc) how long GNU/Linux will really suspends executing the utility.

The program speed downgrade at the development machine that mball needed was 3 times the original execution time, while system loading stayed under 30%. At that time the development machine is an 486DX2-66 with a Seagate ST32151N SCSI hard disk.

The extra usleep code is only active if you run these utils with the -quiet switch and when this is set in mbsetup. See menu 1->5. With this switch, the program is mostly run by cron. If you omit this switch, this is probably when you start the program manually, it will then always run at full speed, no matter what the setting in mbsetup is.

If you have a fast system or don't care that the performance of your system drops because of background processing, you can turn this future off with mbsetup in the global section. (menu 1->5).

Remember, if you have a PII-400 MMX or so with IDE disks, you may still have performance problems and need to set that switch to yes. There is only one way to find out if you need it.

Well, actually, I tested this on a Dell Latitude PII-266, setting the switch to yes gave better performance then no. Why? The CPU has more time for the slow IDE disk. With the slow switch on programs runs even faster then with the switch off. Modern hardware (PIII and later) may run fine with the slow switch turned off.

In January 2004 the usleep code is replaced by a milliseconds timer written around the nanosleep system call. This call conforms to POSIX.1b and is written so that if the timer is interrupted it will resume with the time left to do. The real minimum time the timer runs is 10 ms on Intel systems, see also **man nanosleep** All usleep(1) calls are now msleep(1) calls, in practice these are pauses of 10 ms.

Michiel.

# How to set up an web server to work with MBSE BBS.

**Introduction**
To let a web server work with MBSE BBS you must organize a special file structure. Note that even if you don't set up a web server you must still create a structure like this for the fidonet mailer, if you don't, **mail and files will get lost!** This description is written for Apache, this is installed on most GNU/Linux and xxxBSD distributions. For the directory structure, read the setup for the FTP server.

**Important settings in mbsetup.**
Screen 1.18:
1. Base path      /opt/mbse/ftp/pub
Screen 1.19:
1. Docs root      /var/www/htdocs
2. Link to ftp    files
3. URL name       http://www.mbse.ym

The base ftp path is the default which mbse bbs installs. The Docs root for the apache httpd server is different on each distribution, in this example it is set for Slackware 8.0. Because the directory /var/www/htdocs has nothing to do with /opt/mbse/ftp/pub we make an alias link named **files**. For the URL name you must fill in the real internet name how your bbs is reached. If someone types this in, he must get the index.html from the directory /var/www/htdocs, the docs root. I just assume you have your http server up and running.
Now you can run the command **mbfile index**, this will create a main index in the directory /opt/mbse/ftp/pub and additional index.html files in all your download areas.

**Change Apache server configuration.**
As root edit your httpd.conf file, it is possible that you find it in /etc/apache, but that depends on your distribution. In the section between <IfModule mod_alias.c> and </IfModule> insert the following lines:

```
#
# Alias for MBSE BBS download areas.
#
Alias /files/ /opt/mbse/ftp/pub/
<Directory "/opt/mbse/ftp/pub">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

#
# Alias to access the MBSE BBS documenatation
#
Alias /mbseman/ /opt/mbse/html/
<Directory "/opt/mbse/html">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Then restart Apache with the command **apachectl restart** Now if you access your webserver with for example http://www.mybbs.com/files/ you must be able to browse the file lists. If you use http://www.mybbs.com/mbseman/ you must be able to browse this documentation online.

**Example views**
This are some snapshots of how this will look like:

## File areas at Dreamland BBS – Konqueror

Locatie  Bewerken  Beeld  Ga naar  Bladwijzers  Hulpmiddelen  Instellingen  Venster  Help

Locatie: http://www.dreamlandbbs.com/files/

# File areas at Dreamland BBS

| Area | Description | Files | Total size | Last added |
|---|---|---|---|---|
| 1 | ERO: Dutch sexstories | 1382 | 17 Mb. | 22-01-2002 |
| 2 | ERO: Anal pictures | 117 | 8 Mb. | 05-01-2002 |
| 3 | ERO: Bizar pictures | 252 | 18 Mb. | 11-01-2002 |
| 4 | ERO: Cartoons/drawings | 33 | 2 Mb. | 02-01-2001 |
| 5 | ERO: All sorts | 117 | 7 Mb. | 20-10-2001 |
| 6 | ERO: Cum shots | 218 | 22 Mb. | 29-12-2001 |
| 7 | ERO: Nodelist files | 0 | 0 Kb. | |
| 8 | ERO: Gay pictures | 5 | 232 Kb. | 12-06-2000 |
| 9 | ERO: Groupsex | 253 | 17 Mb. | 01-02-2002 |
| 10 | ERO: Lesbian girls | 361 | 22 Mb. | 28-11-2001 |
| 11 | ERO: Mpeg sexmovies | 189 | 691 Mb. | 10-03-2002 |
| 12 | ERO: Oral pictures | 202 | 17 Mb. | 26-12-2001 |
| 13 | ERO: Oriental girls | 385 | 34 Mb. | 18-01-2002 |
| 14 | ERO: Pissing & shitting | 143 | 13 Mb. | 26-12-2001 |
| 15 | ERO: Private & amateur pictures | 337 | 27 Mb. | 04-04-2000 |
| 16 | ERO: Erotic games & programs | 0 | 0 Kb. | |
| 17 | ERO: Shaved | 0 | 0 Kb. | |
| 18 | ERO: Sm & bondage | 10 | 930 Kb. | 07-07-1999 |
| 19 | ERO: Teenagers | 941 | 59 Mb. | 01-02-2002 |
| 20 | ERO: Transsexual | 5 | 390 Kb. | 05-01-2002 |
| 21 | ERO: Various files | 39 | 3 Mb. | 24-10-2001 |
| 22 | ERO: Women posing | 1564 | 120 Mb. | 06-02-2002 |
| 23 | ERO: X-rated | 1006 | 74 Mb. | 26-01-2002 |
| 24 | ERO: English sexstories | 471 | 8 Mb. | 06-01-2002 |
| 25 | ERO: Toys | 31 | 3 Mb. | 01-04-1999 |

Laden voltooid

---

## FS: Pictures – Konqueror

Locatie  Bewerken  Beeld  Ga naar  Bladwijzers  Hulpmiddelen  Instellingen  Venster  Help

Locatie: http://www.dreamlandbbs.com/files/filegate/fs/pic/index.html

Home  Back  Next page

# File index of FS: Pictures

| Nr. | Filename | Date | Size | Downloads | Description |
|---|---|---|---|---|---|
| 1 |  | 26-01-2002 | 5 Kb. | 0 | FS PIC: BY737200 |
| 2 |  | 26-01-2002 | 2 Kb. | 0 | FS PIC: BY737200P |
| 3 |  | 26-01-2002 | 4 Kb. | 0 | FS PIC: C17_BHC |
| 4 |  | 26-01-2002 | 5 Kb. | 0 | FS PIC: C17_BHCP |

Laden voltooid

**Counting downloads**

If you want to increase the download counters when files are downloaded via www you must make sure that the web logs the downloads to a **access_log** file, for example /var/log/apache/access_log. This logfile must be readable by user mbse and must be in **combined** format. You can set this in your apache configuration file. Then in **mbsetup** menu 1.13.4 enter full file name and path to this logfile. The **mball** program will parse this file and increase the download counters for the files that are downloaded from the web server.

End of Mbse Manual.