

FBB software - Documentation

CONTENTS

- 1 ACKNOWLEDGMENTS
- 2 COPYRIGHT
- 3 FEATURES
- 4 FIRST INSTALLATION
 - 4.1 RUNFBB.BAT (only for WinFBB)
 - 4.2 XFBB.SH (only for LinFBB)
 - 4.3 APPEL.BAT (only for DosFBB)
 - 4.4 PORT.SYS
 - 4.5 INITTNCn.SYS
 - 4.6 MAINTn.SYS
 - 4.7 BBS.SYS
 - 4.8 FORWARD.SYS
 - 4.9 BEACONx.SYS
 - 4.10 EPURMESS.INI
 - 4.11 SWAPP.SYS
 - 4.12 INIT.SRV
 - 4.13 CRON.SYS
 - 4.14 REJECT.SYS
 - 4.15 PROTECT.SYS
 - 4.16 LANGUE.SYS
 - 4.17 THEMES.SYS
 - 4.18 REDIST.SYS
 - 4.19 MEMO.SYS
 - 4.20 PK-232
 - 4.21 BPQ
 - 4.22 KANTRONICS KAM
 - 4.23 DRSI
 - 4.24 TFPCX/TFPCR
- 5 BBS COMMANDS
- 6 SERVER COMMANDS
 - 6.1 Connections - Statistics
 - 6.2 Documentation
 - 6.3 QTH-locator
 - 6.4 "Call-book"
 - 6.5 Satellite orbits
- 7 FBBDOS COMMANDS
- 8 SYSOP COMMANDS
- 9 CONSOLE COMMANDS
- 10 FULL SCREEN EDITOR (DosFBB)
- 11 VARIABLES
- 12 MISC. TOOLS
 - 12.1 FBBSETUP.EXE (only DosFBB)
 - INSTWFBB.COM (only WinFBB)
 - INSTALL.SH (only LinFBB)
 - 12.2 EPURMESS.COM
 - 12.3 EPURWP.COM
 - 12.4 LOGSTAT.EXE

- 12.5 SATUPDAT.EXE
- 12.6 SLEEP.COM
- 12.7 MAKEPG.COM
- 12.8 SETUSER.COM
- 12.9 CLEANUP.COM
- 12.10 MAINTINF.COM
- 12.11 CUT.COM
- 12.12 FV.COM
- 12.13 ISDAY.COM
- 12.14 CLR_USER.COM
- 12.15 FBB_ZM (only LinFBB)
- 12.16 XFBB (only LinFBB)
- 12.17 FBB_VIEW (only LinFBB)

13 DRIVERS AND PHONE MODEM:

Only for WinFBB:

- 13.1 FBBCOMM.DRV

Only for LinFBB:

- AF_AX25 domain socket (See APPENDIX 23 !!)
- AF_NETROM domain socket (See APPENDIX 23 !!)
- AF_ROSE domain socket (See APPENDIX 23 !!)

Only for DosFBB:

- 13.2 ESS.COM
- 13.3 ESSKAM.COM
- 13.4 FBBIOS.COM
- TFPCX/TFPCR (See APPENDIX 25 !!)

For both WinFBB and DosFBB:

- BPQ (See APPENDIX 3 !!)

Installation of a TELEPHONE MODEM. (See APPENDIX 20 !!)

14 SERVERS

- 14.1 REQDIR : Request directory-listing from a BBS's files-area.
- 14.2 REQFIL : Request an ASCII-file from a remote BBS.
- 14.3 NEWDOC : Update DOC-area of the BBS.
- 14.4 REQCFG : Request a list of the BBS's configuration.
- 14.5 MULTI : Send multipel identical messages to different addressees.

15 WHITE PAGES

- DESCRIPTION.
- UPDATE REQUESTS.
- DATABASE DESCRIPTION.
- DATABASE MANAGER.
- EPURWP AND UPDATE MESSAGES.
- WP SERVER REQUESTS.

16 OTHER FILE FORMATS (see also section 4.1 - 4.24 for most-used files)

- 16.1 DSZ.BAT Start binary transfer
- 16.2 PASSWD.SYS Password-file for remote sysops
- 16.3 ERROR.SYS File for logging errors
- 16.4 ETAT.SYS Shows when the BBS was put in service
- 16.5 HEARD.BIN Stations heard on radio-ports
- 16.6 english.TXT Texts used by the BBS. One for each language
- english.HLP User-help. One for each language
- english.ENT Logon-text. One for each language
- english.NEW Logon-text. One for each language. Shows ONCE
- english.INF Texts used by the BBS. One for each language
- 16.7 FBBLOG.nn Logfile for the BBS. One for each week
- 16.8 nnnnn.SAT Characteristics of satellites

- 16.9 NEWDOC.SYS Authorize callsigns for NEWDOC-server
- 16.10 STATIS.DAT Statistical data of BBS-usage
- 16.11 OPTIONS.SYS Options for console-usage
- 16.12 DIRMES.SYS Records for every single message
- 16.13 TPSTAT.SYS
- 16.14 WFBID.SYS Records of BIDs
- 16.15 INF.SYS Records of users
- 16.16 YAPPLBL.DAT Information on binary files
- 16.17 SATEL.DAT Information on satellites
- 16.18 WP.SYS White Pages database
- 16.19 MESS.WP File from WP tp update BBS

17 GLOSSARY

18 APPENDIX

- 18.1 APPENDIX 1 : EDIT Commands
- 18.2 APPENDIX 2 : PK232 Host-Mode
- 18.3 APPENDIX 3 : G8BPQ interface
- 18.4 APPENDIX 4 : Kantronics KAM
- 18.5 APPENDIX 5 : DRSI-card
- 18.6 APPENDIX 6 : PROCESS NUMBERS AND HELP
- 18.7 APPENDIX 7 : RECORDING A MESSAGE
- 18.8 APPENDIX 8 : TRICKS AND TIPS
- 18.9 APPENDIX 9 : FBB FORWARD PROTOCOLE
- 18.10 APPENDIX 10: COMPRESSED FORWARDING
- 18.11 APPENDIX 11: FORMAT OF ACK MESSAGES
- 18.12 APPENDIX 12: REPLACEMENT CHARACTERS OR WILDCARDS
- 18.13 APPENDIX 13: PROGRAMMING TECHNIQS FOR SERVERS
- 18.14 APPENDIX 14: CONTENTS OF THE DIRECTORIES
- 18.15 APPENDIX 15: CONNECTION FILTERING
- 18.16 APPENDIX 16: MESSAGE FILTERING
- 18.17 APPENDIX 17: PG PROGRAMS DEVELOPMENT
- 18.18 APPENDIX 18: UNPROTO MESSAGE LISTS
- 18.19 APPENDIX 19: EXTENSIONS TO THE YAPP PROTOCOLE
- 18.20 APPENDIX 20: Telephone modem
- 18.21 APPENDIX 21: File filtering
- 18.22 APPENDIX 22: MAIL.IN (import messages directly)
- 18.23 APPENDIX 23: Installation-help for LinFBB
- 18.24 APPENDIX 24: TELNET installation for LinFBB and WinFBB
- 18.25 APPENDIX 25: TFPCX / TFPCR / TFKISS (Hostmode KISS-driver)
- 18.26 APPENDIX 26: EMS/XMS memory.
- 18.27 APPENDIX 27: Using FBB with PE1CHL NET and the BPQ Hostmode Emulator
- 18.28 APPENDIX 28: Conference-mode
- 18.29 APPENDIX 29: Gateway-mode
- 18.30 APPENDIX 30: TFWIN.DLL.

1 ACKNOWLEDGEMENTS:

I would like to particularly thank Patrick F1GHV who was my first "official" beta-tester.

I also must thank Per LA6CU, who helps me from many years to set up this software, debugging and documentation, as well as Claude F1CDC, the "dumb individual", who, with his advices and criticism (often vitriolic!), has given his support in this task.

I would like to thank, as well, all the beta-test team and all the programmers for the development of applications and complementary tools for this software. Such a support made me free of these tasks, very often ingrate.

I thank again all the SYSOPS and users who are confident in the product, and cooperate, thanks to their comments and suggestions, in improving this BBS software package.

I would like, at the end, to thank WA8DED for the excellent piece of software, the Host-Mode for the TNC, and also NORD<>LINK for the additions made to it.

2 COPYRIGHT:

This software is delivered at no cost and is free of all charges. It can be freely copied or installed for any HAM-RADIO application. Any other usage for commercial, professional, or industrial purposes is forbidden.

Despite the free sharing nature of this product, a participation to the investment and to the mailing cost at the level of 100 FF or 20 US Dollars (or more) would be most welcome.

The author can be contacted by email or at the following address:

Jean-Paul ROUBELAT
6, rue George SAND
31120 ROQUETTES
FRANCE

Email: f6fbb@f6fbb.org

3 FEATURES:

- This software is available for Dos, Windows and Linux. Most features and operations are identical to all versions. Where there are differences, they are clearly marked in the manual with DosFBB, WinFBB and LinFBB (for Linux). There are two LinFBB-versions for Linux:
a windowing version (using X11 windowing system and MOTIF toolkit).
a daemon version. The daemon version has no interface and works alone in background (like works an anti-virus, hi...). So the computer may be used for any application while the BBS is running.
- The usage of this software is very close to the one made by WA7MBL, whose commands have been kept. It has also a set of specific and original supplementary commands.
- The code does not work on XT.
Works on AT286 and UP with at least 2MB of memory.
Code uses up to 16MB of memory.
- Multiconnections: up to 50 simultaneous channels on 8 TNC (4 or 8 channels per TNC depending on the software used).
- Multiplexing: this software supports usage of an external MUX and supports also extension boards in the case of a hardware configuration having more than 2 ports. The MUX connects 4 TNC on 1 serial port, either com1 or com2. Printed board is available from the ATEPRA association, (ATEPRA, 23 rue de Provins, 77520 MONS EN MONTAIS - FRANCE).

The software operates:

- with any TNC2 or clone fitted with the special 27256 eprom containing the WA8DED firmware, or TF4/TF8, with a Z80 clock of 2.5 or, better, 4.9 MHz.
- with a PK232 in Host-Mode (but TFPK software, WA8DED hostmode for PK232 works better !).

- with a G8BPQ node.
- With a KAM hostmode.
- With TFPCX/TFPCR/TFKISS and tnc's in kiss-mode or a kisslink to a nodesystem.
- Standard BBS: It has many BBS-commands identical to the BBSs by WA7MBL or WORLI. It has also server functions (computation of satellite orbits, callbook, and operator customizable chapters, gateway to an other channel, the users being able to chat together in real time on the same or different channels, etc...).
- Forwarding: With other BBSs of WA7MBL type or compatible, BIDS management (up to 32000 bids, depending on a parameter, are saved in a separate file), a BID is automatically generated if the user does not give one. Private message works with the management of MID. The messages are suppressed automatically after a delay which can be user defined. This is true for bulletins and personal mail.
- The forwarding is optimized between BBSs using FBB protocol type. It is more efficient on a VHF/UHF network. This kind of forwarding can be disabled by configuration.
- The compressed forwarding allows the reduction by a factor of about 40-50% the quantity of data to exchange in big messages. The messages are protected by checksums, so the transfer is made error-free.
- X forwarding Protocol is also implemented. XForwarding now supports re-routing and swapping. Checksum is implemented in the X-forwarding.
- The forwarding is simultaneous on the various ports regardless if they are incoming or outgoing. There might be several forwardings outgoing per port which number is set by parameter. The number of incoming forwarding is a function of the available channels. The time and the period of forward can be set separately on each port.
- Hierarchical routing is supported.
- The ping-pong phenomenon is automatically detected and an information is given to the sysop by a system message.
- Messages and bulletins for SYSOP may be duplicated for one or more destination-call which can be defined by configuration.
- A detailed log of the BBS activity is maintained and a statistic analysis program, written by FC1MVP, is also available.
- Binary transfer is supported with the usage of the YAPP protocol of WA7MBL. An extension to this protocol has been made, including the automatic restart and the checksum, should a stop occur or a disconnection take place during the transfer. This extension to the protocol works with the TPK, packet terminal program written by FC1EBN.
- Gateway between connected stations or with another port.
- Conference within the limits of the available ports and channels.
- Language: At the first connection, the connect language is attributed to the user, depending of the callsign.
- Remote SYSOP operation is supported and the housekeeping of the BBS messages, mail and old mail is done each and every night at low activity period which can be parametered, and is done automatically.

- DosFBB works under DesqView.
- Unproto beacons may be sent on any port and via any path (more than one per port, if wanted) for each arrived message. Users may be validated to be allowed to ask for resync of unproto-beacons to update their internal messages-list without connecting the BBS. This feature is used by TPK and some other modern terminal-programs.
- Modem in WinFBB support XMODEM, YMODEM and ZMODEM upload and download with the DSZ shareware.
- Modem in LinFBB support XMODEM, YMODEM and ZMODEM upload and download with the utility 'fbb_zm' that is included in the LinFBB package.
- Modem in DosFBB support XMODEM, YMODEM and ZMODEM download. Needs a recent version of FBBIOS. Upload is supported only for XMODEM.
- langname.NEW works as langname.ENT, but is sent only ONCE to each user (at the first connection after the file has been updated).
- Max KB for download in modem and yapp port may be set in INIT.SRV
- Only one simultaneous forward for a station is allowed.
- ACK-messages show only the first R:line (if passed through FBBs).
- Too long callsigns in routing are rejected (asked by G1NNB)
- 3 extensions to the version 1 of FBB protocol:
 - 1: Reject messages from being forwarded to the BBS.
 - 2: Receive messages, but hold them for sysop before release.
 - 3: Reject message-proposals with wrong format.
 In the first 2 cases a message may be sent to sysop to advice him of the problem (this is parametrized in INIT.SRV file).
- No more than 1 message from one callsign in each proposal block. If more, the other messages are delayed to the next proposal blocks. This avoid long lists of message from the same sender. May be disabled with the letter D in the T line of FORWARD.SYS
- Oldmail has a format compatible with import.
- Information about the number of free buffers in the TNC in the status line. Maximum buffers per channel is divided by two if the number of free buffers is going too low.
- New feature for FORWARD.SYS for sending of 7+ messages:

New line J in FORWARD.SYS : (data means 7+)

J 0 : no data sent

J 1 : data is sent (default)

J 2 : personal data is sent

J 3 : only data is sent (exclusive)
- Forced disconnection of all links at housekeeping time+15 min.
- Binary transfer protocol Autobin (used in GP, SP TurboPacket, etc...)

Commands are BGET and BPUT.
- Full screen editor usable to send or reply messages.

In DosFBB validated with AltF6.

Always used in WinFBB.
- Command "WHERE file" in FBBDOS (to find one or more files in FBBDOS)

Wildcards are accepted.

- Command TH -> access to the thematic zone. The file THEMES.SYS is configuring this zone.
- FBBIOS40 (and up) should work with 16550 FIFO chips (for DosFBB).
- Server REDIST is included. Used to redirect personal message as bulletin.
- System files are checked after local editing.
- Implementation of the VXD interface of the BPQ code.
- All program run from FORWARD.SYS are synchronous.
- Filters accepted as DLL (much faster...). Both for WinFBB and DosFBB (!!).
- Ethernet/tcp-ip implementation. Needs the WINSOCK.DLL Microsoft library.
- Information on a connected station. Double click on the callsign line of the connected stations give the info (WinFBB).
- Uploaded file via modem is validated and set to the user.
- F_FILTER validated for filtering of uploaded files.
The F_FILTER may also be created as a DLL. Both for WinFBB and DosFBB (!!). The filter must be installed in the path (\FBB\BIN) of Dos.
- Every connected channel is bufferized. The connection is visible while double-clicking on the user in the list (WinFBB).
- BBS forwarding is not allowed on a GUEST port if the BBS is not declared.
- Two DLLs have been developed to be called from FORWARD.SYS (WinFBB):
ICOM.DLL uses the same parameters as ICOM.COM
PROGTX.DLL sends a binary file to a tx through a serial com
format : PROGTX [options] filename
options : /C2 = COM2 (default COM1)
 /A02F8 = Com address is 02F8 (excludes /C OPTION)
 /B9600 = Baud rate is 9600 Bauds (default 1200)
- A hold-message is not recorded in the list, only saved as a file.
So no unproto list or other is sent.
- New programs INSTWFBB for WinFBB and FBBSETUP for DosFBB to configure the software. In LinFBB there is a small INSTALL.SH script that does part of the job for LinFBB.
- Binary forwarding via telephone modem (FBB or XFWD)
- WinFBB: The configuration may be called from the Config menu. Updates are made without rebooting the software (after saving the config).
- Blocking DOS commands (EDIT, ...) is killed after 60 seconds.
- Messages or files are not accepted if disk free is less than 100K.
- Telnet interface is working as modem interface except that password is mandatory.
- WinFBB: Servers can be made as DLLs provided they are fast enough (no multitasking during a DLL activity).

- WinFBB: Selection of zone in the channel windows. Accept copy (=Ctrl Ins) to clipboard and print. Menu is activated with the right button of the mouse inside the text window.
- Selection of a word (left double click) or a list of characters (Ctrl left double click).
- Sysop-command HO puts a message in the hold list.
- ReadOnly access if there is no password defined (modem/telnet port)
- MD ("Message Digest") password system. This password is based on a public key. The 10 digits are never the same and the answer is also never the same. So it is not possible to find the original password from the answer.
- Both WinFBB and DosFBB use protected mode (DPMI).
- Swapping is now using hierarchy. For instance :
 @ F6FBB.FRA.EU @ F6FBB.FMLR.FRA.EU
 Only the first full address will be swapped to the other
 @ F6FBB @ F6FBB.FMLR.FRA.EU
 All F6FBB addresses will be swapped to the other
- WinFBB32 is a 32-bit version of WinFBB. For use with Windows NT and Windows95. Can NOT be used with Windows 3.x !
- W32 : TfWin32.DLL interface validated. Only WinFBB 32-bit!

4 FIRST INSTALLATION:

This part of the manual is intended for the sysop who has never used FBB software before. I hope this part of the manual will help you to get "on the air" with FBB without too many problems.

FBB is designed in 3 versions. One for Dos (referred to as DosFBB), one for Windows (referred to as WinFBB) and one for Linux (referred to as LinFBB). This manual covers all versions. Most of the contents is identical for all versions, but when there is a difference, it will be clearly stated. In either case, I presume that you have installed your operating-system (DOS, Windows or Linux) correct before you attempt to install FBB.

Normally you will install FBB from diskettes. Normally your diskette-drive will be called A, and normally you will install the FBB-program on hard-disk called C. But this is no absolute need, you may use other drives as well. More about this a little later.

Your hard-disk is organised in several directories. If you are not familiar with this, you should read in your operating-system's manual about how the hard-disk is organized.

FBB will normally be installed on your hard-disk in a directory named FBB. This directory should be placed directly under root (C:\FBB). This directory will be automatically created for you. The BBS will also need a lot of other directories, and they can be created directly from root, or they can be created under the FBB-directory. In the procedure I will describe here, only the FBB-directory will be at the root (C:\FBB) of the hard-disk, while all the other directories will be placed under FBB (like C:\FBB\USERS etc). But you can choose this yourself. More about this a little later.

However, this is the recommended organization for your directories. The directories will be created like this if you make no changes (a list of the files and directories is given in appendix 14).

You should now be ready to start the installation procedure. Put the first of the diskettes in your disk-drive.

For WinFBB:

Have Windows up-and-running. Install WinFBB from your diskette-drive the same way as you normally install programmes from diskette. In Windows 3.x I think it should be normal to go to the file-manager, select RUN, and type A:INSTALL. In Win95 you may press START, then RUN, then type A:INSTALL.

You will first be asked for the directory (and drive) that you want to have WinFBB installed to. If you accept the defaults, just press ENTER. If you want WinFBB installed in another directory or on another drive, just do the changes you want now, by typing your own drive/path.

The screen will now display a text with some more information on the installation. Read it carefully before you proceed (by closing that window).

The installation continues. You will be advised that SYSTEM.INI will be modified to replace the COMM-driver. This is the driver for the TNCs and telephone-modem. You will also be told to increase the environment to 2kb. AUTOEXEC.BAT will also be updated with the correct path to your FBB-directory.

Next, the installation-procedure continues by starting the program INSTWFBB. (If this is not started during installation, you should start it after the installation has completed, by clicking on the icon for WinFBB Configuration). This program lets you set most of the parametres you need (all parametres for PORT.SYS and INIT.SRV) for your installation. Please take great care here, and go through ALL the pages of this installation very carefully before you exit the program. Use the NEXT - PREV buttons to go through the entire installation. Be sure to insert your correct callsign, your correct H-route etc. INSTWFBB will update both INIT.SRV and PORT.SYS for you, you will not need to change them manually later, if you take great care to do a good job now.

When you are sure that your setup now is correct, you may exit the INSTWFBB. The PC will then reboot, to finish the setup. After this, you will have a new program-group with icons for WinFBB and for WinFBB Configuration. You may run WinFBB Configuration at any time later, to change your setup again. The configuration may also be changed while the BBS is running, by clicking on the word Config on the top of the screen.

Next, the same procedure for DosFBB and LinFBB follows. Skip them, and go directly to the section called "FIRST-TIME SETUP OF DIFFERENT FILES."

For DosFBB:

You should now be ready to start the installation procedure. Put the first of the diskettes in your disk-drive, and type A: if that drive is named A, or B: if that drive is named B. Press ENTER to execute the command (you must always press ENTER after a command to the PC). Your PC will now answer with something like A:\> or B:\>

If you now are ready to install FBB for the first time on your PC, you type INSTA700.

You will now see at your screen something like this:

----- QUOTE: -----

**** FIRST TIME INSTALLATION OF FBB 7.00 FROM DISK %disk% ****

```
- Main BBS-directory - C:\FBB
- System-files       - C:\FBB\SYSTEM
- Messages           - C:\FBB\MAIL
- Binary-messages    - C:\FBB\BINMAIL
- User's dir.        - C:\FBB\USERS
- Yapp files         - C:\FBB\USERS\YAPP
- Oldmail            - C:\FBB\OLDMAIL
- Statistics         - C:\FBB\STATS
- Documentation      - C:\FBB\DOCS
- FBB-manuals        - C:\FBB\DOC
- Fwd-include-files  - C:\FBB\SYSTEM\FWD
- Executable files   - C:\FBB\BIN
- Example-files      - C:\FBB\EXAMPLES.700
- Utilities          - C:\FBB\UTIL
(Fwd-include-files MUST stay under %system%.)
```

If this configuration does not suit your needs, you must stop now with CTRL-C and edit the SET-lines in the file A:SUI TE1.BAT. If a version of FBB 5.15c already exists, press Ctrl-C and use 515c700.BAT for upgrading.

----- UNQUOTE. -----

This screen may vary a bit from version to version, but the work you have to do, will still be the same. First of all, you will see that the PC says that you will install from disk A. Is this correct ? If you are installing from disk-drive B, you must stop the installation now. The same applies if you want to use other directory-names or paths for some of the other directories that are listed on your screen now.

If you DO want to change any of these things, you MUST do it now. The changes are described in this paragraph; if you do NOT want any changes, you can go on to the next paragraph. You stop the installation-batch by pressing CTRL-C. Now you will see the standard A:\> prompt again. Now you must use a normal text-editor and edit the file SUI TE1.BAT on your diskette. I presume you have such an editor... Make sure that it is able to save the file again in ASCII (DOS) format ! So if you use for example Word Perfect, you MUST remember to save the file again in DOS-format !

At the beginning of that file, you will see a line saying SET disk=A: If you want to install from B-drive, you edit this line like this:

SET disk=B:

You change only the letter after '=', nothing else ! You can also change all directory-names or directory-paths in this same way. If you do not want to use C:\FBB\SYSTEM, but want it to be C:\FICHIERS, you just change the line saying

```
SET system=C:\FBB\SYSTEM
to
SET system=C:\FICHIERS
```

and so on. You must ONLY change the name behind the '=' sign, you must NOT change anything before that in that line. You can do this with any (or all) the directory-names and paths. But remember that you will need to do some extra work with other config-files later if you do not use the default installation.

When you have finished doing the changes you want in the lines starting with SET, you must save the file again to disk. When that is done, you start over again from the beginning, by typing INSTA700 etc.

OK. That is about all. Now the BBS-program and all its files will be installed on your hard-disk. You only have to change diskettes each time you are asked to (if you install from more than one diskette).

Next, the installation-procedure continues by starting the program FBBSETUP. This program lets you set all the parametres in INIT.SRV and PORT.SYS that you need for your installation.
(FBBSETUP supports multiple language files, but for the first time installation you will probably have only the English language files.) Please take great care here, and go through ALL the pages of this installation very carefully before you exit the program. Be sure to put in your correct callsign, your correct H-route etc. FBBSETUP will update INIT.SRV and PORT.SYS for you, you will not need to change it manually later, if you take great care to do a good job now.

The path FBB\BIN must be included in your PATH in AUTOEXEC.BAT. Do this by adding to the PATH line in AUTOEXEC.BAT ";C:\FBB\BIN" like :

```
PATH C:\;C:\FBB\BIN
```

Next, the same procedure for LinFBB follows. Skip it, and go directly to the section called "FIRST-TIME SETUP OF DIFFERENT FILES."

For LinFBB:

There are two versions:

- a windowing version (using X11 windowing system and MOTIF toolkit).
- a daemon version. The daemon version has no interface and works alone in background (like works an anti-virus, hi...). So the computer may be used for any application while the BBS is running.

Tnx to SM6TKY:

If you need some help in setting up AF_AX25 and AF_NETROM for use with LinFBB, you may want to read the Appendix 23.

You should now be ready to start the installation procedure.
To install and run LinFBB you MUST have root access,
you may log in as root or you can go to superuser mode with: su root

(1)
Put the floppy disk into the diskdrive.

(2)
If you have MTOOLS installed.
To copy from floppy disk A:, type: mcopy a:/x700.tgz /tmp
To copy from floppy disk B:, type: mcopy b:/x700.tgz /tmp

If you have NOT MTOOLS installed.
To mount floppy drive A: type: mount -t msdos /dev/fd0 /mnt
To mount floppy drive B: type: mount -t msdos /dev/fd1 /mnt

Then type: cp /mnt/x700.tgz /tmp , to copy the contents of disk to /tmp,
then type: umount /mnt , to unmount the floppy disk.

(3)
Now you should create the base directory where you want LinFBB.
Example, if you want LinFBB in: /usr/local/fbb

Then type: mkdir /usr/local/fbb

(4)

If you now are ready to install LinFBB for the first time on your PC,

Type: cd /usr/local/fbb , to go to the LinFBB root directory.

Type: tar xvzf /tmp/x700.tgz , to unpack the LinFBB archive.

OK. That is about all.

Now the BBS-program and all its files will be installed on your hard-disk.

Next, the installation-procedure continues by starting the installation script named INSTALL.SH

This script lets you set most of the parametres in INIT.SRV (BUT NOT ALL !) that you need for your installation. Please take great care here, and go through ALL the questions of this installation very carefully before you exit the script. Be sure to put in your correct callsign, your correct H-route etc. The script INSTALL.SH will update INIT.SRV for you, but you will also need to check the rest of it manually, too, afterwards !

FIRST-TIME SETUP OF DIFFERENT FILES.

So far so good... Now we must modify some files, to make the system work. I will describe all the changes you MUST do, and some changes that you CAN do, if you like. Read this section very carefully. For most of the files it is VERY important that the number of non-comment lines must NOT be changed. Later in the sysop-manual you will find more details on some of these files, and also details on other files that are not described here in this startup-section.

Files you MUST check/change:

RUNFBB.BAT (ONLY for WinFBB)

XFBB.SH (ONLY for LinFBB)

APPEL.BAT (ONLY for DosFBB)

PORT.SYS (MUST be changed for LinFBB, no changes needed for WinFBB or DosFBB)

INITTNCx.SYS

MAINTx.SYS

Files you must change if you plan to do any forward:

BBS.SYS

FORWARD.SYS

Files you should check, too, but not SO important:

BEACONx.SYS

EPURMESS.INI

SWAPP.SYS

Files that are automatically configured by INSTWFBB and FBBSETUP, and should normally not be needed to change manually after (DosFBB and WinFBB).

INIT.SRV

PORT.SYS

Files you may have a look at later (not so important).

CRON.SYS

REJECT.SYS

PROTECT.SYS

LANGUE.SYS

THEMES.SYS

REDIST.SYS

MEMO.SYS

Start the BBS for the first time.

Now you should be ready to start the BBS for the first time. Before you do, make sure that the internal clock of the BBS is correct. If not, you must correct it now with the DOS-command DATE and TIME. In most newer PCs today, the clock is automatically set at each power-up. But in some older PCs you may need to use a special clock/calendar card in your PC, and use a special program to set the date/time at each startup. For this you must look in the manual for that clock/calendar card and include the appropriate command in your AUTOEXEC.BAT file.

For WinFBB:

Go to the FBB program-group and double-click the WinFBB icon.

For DosFBB:

Go to the FBB directory and type APPEL and press ENTER.

For LinFBB:

To start LinFBB in X-Windows:

Go to base directory and run "./xfbb.sh" in a xterm.

To start LinFBB daemon:

Go to base directory and run "./xfbb.sh -d &".

The first time you start the BBS like this, it may ask you if you want to create some files. Just answer Y (for Yes) every time. Then those files are automatically created for you.

The BBS should now be ready for use. Try a connect to the BBS. Do this by pressing F2. The BBS will ask you for your name, city, home-BBS and ZIP-code.

Only for LinFBB:

Daemon version of LinFBB:

Go to 'bin' directory, then use the utility 'xfbbC' to connect (eg: xfbbC -c).

If you have set up PORT.SYS to allow incoming Telnet connections to LinFBB, then you can also use this..

For example, if you defined LinFBB to listen on port 6300 (hex = 189C)

for incoming Telnet connections, then you type: telnet sm6tky.ampr.org 6300

X-Windows version of LinFBB:

Select the "Window" in menu, and then select "Console" to enter Console connection.

Next you must define all the BBSs you want to forward to. You do this with the command EU. If you want to have forward to for example LA2D, you type EU LA2D.

The BBS will say that this callsign is not known, and ask if you want to create it. You answer Y. Next you will see a line of different options for that user. You only have to type B. Now that callsign (LA2D) will be defined as a BBS. You must press ENTER once more to stop editing the user. Repeat this procedure with the callsign of all the BBSs that you will forward to. You terminate the keyboard-session with the command B (Bye).

Well, if all is ok, and you have not encountered any big problems so far, you can stop the BBS.

In DosFBB you do this with Alt-F10 or Alt-x.

In WinFBB you do this by pressing Alt-F4).

In LinFBB you do this:

Daemon version:

Do 'ps ax' , check for 'xfbbd' look after

the process number and do 'kill <process number>'
X-Windows version:
Select "File" in menu, and then select "Exit"
to shutdown the BBS.

You can connect the TNCs, and in INIT.SRV you must change from No to Ok in the line that asks 'Mode test ?', or
in WinFBB : Click on the WinFBB Configuration icon
in LinFBB : Use INSTALL.SH
in DosFBB : Use FBBSETUP
to change the field "TEST MODE" to "NO".

The TNC.

This software will work with different kinds of TNCs or TNC-interfaces:

- TNC-2 or close clones. The EPROM of type 27256 must be replaced by a special EPROM with the WA8DED host-mode software, or by the german TF4, TF8, TF18 or similar.
- PK-232 in host-mode.
- Kantronics KAM in host-mode. You need a special driver called ESSKAM.
- DRSI-card.
- G8BPQ-node (uses TNCs in KISS-mode).

The software for the TNC-2 host-mode EPROMs TF4 and TF8, may be included on the distribution-disks. If you cannot burn new EPROMs yourself, there are a lot of hams around who can. You can also send a new EPROM to F6FBB or FC1GHV (with return-postage and packing material, of course) and they will do the work for you.

The TNC should work on 4.9 MHz. The standard is 2.5 MHz, but a modification for this is normally quite simple. In some cases it might be necessary to replace the Z80 and the Z80 SIO by A-versions.

If you use standard TNC-2 with one of the mentioned host-EPROMs, you can start again now, and all should work fine. Make sure that the TNCs have the same baud-rate as you wrote in PORT.SYS, and that they really are connected to the COM-ports that you said in PORT.SYS.

If you do NOT use a standard TNC-2 with DED-type host, but PK-232, BPQ, Kantronics KAM or DRSI-card, you must read about your special TNC-setup in one (or more) of the chapters PK-232, BPQ, KAM, DRSI or TFPCX/TFPCR before you can start the BBS again.

If the TNCs don't work ok with the software, you will see this when you start the BBS. You will have a lot of error-messages on the screen, and the TNCs may start to resync. Stop the program and go through all the setup once more.

For WinFBB:

You may need to check the port-speed in Windows, make sure it is the same for the port, as you have set the TNC and PORT.SYS to.

4.1 RUNFBB.BAT (only for WinFBB):

RUNFBB.BAT (\FBB\BIN).

This is the very first file that WinFBB will use. You will find it in \FBB\BIN-directory. You don't really need to do any changes in this file, unless you did some changes during the installation-process. If you did, you must change the pathnames in RUNFBB.BAT exactly the same as you did during the installation.

This is very important ! So look for the lines that I have marked with an <--- here :

@echo off

```
rem -----
rem This file is run from RUNFBB.EXE. Only the relevant part is started.
rem It can be START, STOP, RERUN, MAINT or ERROR.
rem Do not forget to type "goto end" at the end of each section !
rem -----
```

break off

set fbb=C:\FBB

<----

goto %1

:start ***** START of START script *****

echo off

cd %fbb%

goto end

rem ***** END of START script *****

:rerun ***** START of RERUN script *****

cd %fbb%

goto end

rem ***** END of RERUN script *****

:maint ***** START of MAINT script *****

cd %fbb%

@echo off

cd %fbb%\system\sat

<----

if not exist amsat.txt goto amsatend

satupdat amsat.txt /n > satupdat.res

satupdat amsat.txt >> satupdat.res

del amsat.txt

:amsatend

echo Backup of important files

copy %fbb%\system\dirmes.sys %fbb%\backup\dirmes.s1

<----

copy %fbb%\system\inf.sys %fbb%\backup\inf.s1

<----

echo Cleaning up messages...

del epurmess.res

epurmess

rem EPURWP Parametre: Number of days before update, and delete

rem Default is 40 and 90

epurwp 45 95

del %fbb%\system\wp\wp.old

<----

cd %fbb%\system

clr_user>>%fbb%\clr_user.res

<----

MAINTINF 3 /a

COPY %FBB%\SYSTEM\INF.NEW %FBB%\SYSTEM\INF.SYS

<----

cd %fbb%

del %fbb%\system\dirmes.new

<----

del %fbb%\system\dirmes.old

<----

if not exist %fbb%\error.sys goto direct

<----

rem -----

rem --- If you want a message when errors have occurred:

rem --- PLEASE exchange LA6CU with your own callsign !!!!

rem echo SP LA6CU >> %FBB%\MAIL.IN

<----

rem echo Error-report

>> %FBB%\MAIL.IN

<----

```

    rem type %FBB%\ERROR.SYS      >> %FBB%\MAIL.IN      <----
    rem echo.                     >> %FBB%\MAIL.IN      <----
    rem ECHO /EX                  >> %FBB%\MAIL.IN      <----
    rem type %FBB%\ERROR.SYS      >> %FBB%\ERROR.LOG     <----
    rem del %FBB%\ERROR.SYS                               <----
    rem -----
:direct
    goto end
rem ***** END of MAINT script *****

:stop ***** START of STOP script *****
    goto end
rem ***** END av STOP script *****

:error ***** START av ERROR script *****
    echo WFBB exited with ERRORLEVEL %2
    sleep 5
    goto end
rem ***** END av ERROR script *****

:end
    echo %1 script is finished.
    sleep 2

```

4.2 XFBB.SH (only for LinFBB):

XFBB.SH (/usr/local/fbb)

This is the very first file that LinFBB will use. You will find it in /usr/local/fbb-directory. (or in that directory you chosen when you installed LinFBB). You don't really need to do any changes in this file, unless you did some changes during the installation-process. If you did, you must change the pathnames in LinFBB.SH exactly the same as you did during the installation. This is very important ! So look for the lines that I have marked with an <--- below.

This file is used to start both the daemon version and X-Windows version of LinFBB, to start the daemon version you should type: ./xfbb.sh -d & in X-Windows xterm you should start LinFBB with: ./xfbb.sh

You must go to the LinFBB root directory before executing this shell script.

```

#!/bin/sh
#
# Shell file to LinFBB software
#
# F6FBB 1996
#
#
# Base directory of LinFBB software
XFBBDIR=/usr/local/xfbb <---
#
# Add local PATH
PATH=$XFBBDIR/bin:$PATH
#
# Export resource file

```



```

XAPPLRESDIR=$XFBBDIR/res

export XFBBDIR PATH XAPPLRESDIR

end_session()
{
    echo "Session of LinFBB is ended"
    exit 0;
}

maintenance()
{
    echo "Running maintenance"
    sleep 2
    $XFBBDIR/bin/epurmess > /dev/null 2>&1 <---
    $XFBBDIR/bin/epurwp 40 90 > /dev/null 2>&1 <---
}

rerun()
{
    echo "Re-running LinFBB"
    sleep 2
}

error_return()
{
    echo "Unknown return value $1"
    sleep 10
}

if [ "$1" = "-d" ]
then
    XFBB=xfbbd
    shift
else
    XFBB=xfbb
fi

PARAM=$*

while [ 1 ]
do
    cd $XFBBDIR <---

    echo
    echo "Running LinFBB ^C to abort"
    sleep 3

    echo "Starting LinFBB ..."
    $XFBB $PARAM <---

    ret=$?
    echo
    case $ret in
        1) end_session ;;
        2) rerun;;
        3) maintenance;;
        *) error_return $ret;;
    esac
done

```

4.3 APPEL.BAT (only for DosFBB):

APPEL.BAT (\FBB\BIN).

This is the very first file that DosFBB will use. You will find it in \FBB\BIN-directory. You don't really need to do any changes in this file, unless you did some changes during the installation-process. If you did, you must change the pathnames in APPEL.BAT exactly the same as you did during the installation.

This is very important ! So look for the lines that I have marked with an <--- here :

```
set fbb=C:\FBB          <-- Defines the base of the FBB file system.
echo off
cd %fbb%                <---
break off
echo Program will start, use ^C to interrupt within 3 seconds
sleep 3
serv %1
if errorlevel 2 goto direct
if errorlevel 1 goto suite
goto fin
:suite
cls
echo
echo Sorting HROUTE.SYS...
copy %fbb%\system\hroute.sys temp.$$$          <---
type temp.$$$ | sort | find /v "# " > %fbb%\system\hroute.sys <---
del temp.$$$
echo
echo Backup of important files...
copy %fbb%\system\dirmes.sys %fbb%\backup\dirmes.s1          <---
copy %fbb%\system\inf.sys %fbb%\backup\inf.s1                <---
echo Cleaning up message-file...
del epurmess.res
epurmess
echo Cleaning up white pages database...
del epurwp.res
epurwp
cd %fbb%\system
maintinf 3 /a
copy %fbb%\system\INF.NEW %fbb%\system\INF.SYS
del dirmes.new
cd %fbb%
:direct
if not exist c:%fbb%\system\sat\amsat.txt goto appel          <---
cd %fbb%\system\sat
satupdat amsat.txt /n > satupdat.res
satupdat amsat.txt >> satupdat.res
del amsat.txt
:appel
cd %fbb%                <---
echo Program will start, use ^C to interrupt within 3 seconds
sleep 3
serv -t
if errorlevel 2 goto direct
if errorlevel 1 goto suite
:fin
echo Program is terminated..
break on
```

Echo on

4.4 PORT.SYS:

PORT.SYS (\FBB\SYSTEM).

First, be aware of this useful feature:

If PORT.SYS does not exist, FBB looks for

PORT_D.SYS (for DosFBB)

PORT_W.SYS (for WinFBB)

PORT_L.SYS (For LinFBB)

So it is possible to have for example DosFBB and WinFBB files in the PC at the same time. This is useful at sites where you want to change from WinFBB to DosFBB from time to time..

I have divided this section in two parts:

1 : Simple instruction (for first-time setup).

2 : Advanced setup, with more details.

2.1 : For BPQ.

2.2 : For DRSI.

1: Simple instruction (for first-time setup).

This file is found in the directory called SYSTEM (if you used my default directory-names). In this file we decide what kind of TNCs to use, how many available channels etc. You must do several changes here, and be very careful that you do all changes correctly. If you don't, the BBS cannot operate properly. Remember that a line starting with # is a comment-line, and the BBS ignores these lines.

```
# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
2      2
#
```

Next you must type one complete line of parameters for each COM-port you use. You must be careful to use the correct values here.

Here are the valid interfaces for DosFBB and WinFBB:

For WinFBB:

The mostly-used interface for WinFBB is 6. This interface replaces ESS, MBBIOS etc and FBBIOS. The driver is FBBCOMM.DRV, which is installed during the installation-process.

In WinFBB ONLY THESE interfaces are available:

```
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#
#           4 = DRSI
#
#           5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#           as stated in INIT.SRV, if any..
#
#           6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#
#           7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#
#           TNC-emulation is T (see below)
#
#           8 = TFWin.dll (only WinFBB32)
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
```

```
#          replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
#          used with WinFBB !
```

For LinFBB (Linux):

```
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#              socket (X) or via Telnet port (T).
```

For DosFBB:

If you use drivers like COMBIOS, BPQ or DRSI-card, you must be careful to give the correct values here. If you use COMBIOS, FBBIOS, BPQ etc, they must always be loaded before you start the BBS-program.

In DosFBB ONLY THESE interfaces are available:

```
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
```

Example for WinFBB:

I use 2 TNCs, one on COM1 and one on COM3, and I use the standard FBBCOMM.DRV loaded by Windows:

```
# Com       : COM-number (1,2,...8)
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#             7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll (only WinFBB32)
# Address   : Address of port in hexadecimal (Needed for multiplexer).
#             In LinFBB:
#             Address is the device name (/dev/cua0).
#             Be sure you have the rights to access to the device (rw-rw-rw-).
#             When using kernel AF_AX25 socket, address is not used.
#             When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1      6          3F8           9600
3      6          338           9600
#
```

Same example for DosFBB:

I use 2 TNCs, one on COM1 and one on COM3, and I use the ESS-driver (ESS must be properly loaded before FBB is started):

```
# Com       : COM-number (1,2,...8)
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
# Address   : Address of port in hexadecimal (Needed for multiplexer).
#             In LinFBB:
```

```

#           Address is the device name (/dev/cua0).
#           Be sure you have the rights to access to the device (rw-rw-rw-).
#           When using kernel AF_AX25 socket, address is not used.
#           When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1    1          3F8             9600
3    1          338             9600
#

```

Same example for LinFBB (Linux):

```

# Com      : COM-number (1,2,...8)
# Interface 9 = Linux. Can work via serial port (D), via AX25 domain
#           socket (X) or via Telnet port (T).
# Address   : Address of port in hexadecimal (Needed for multiplexer).
#           In LinFBB:
#           Address is the device name (/dev/cua0).
#           Be sure you have the rights to access to the device (rw-rw-rw-).
#           When using kernel AF_AX25 socket, address is not used.
#           When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (Device)  Baud
1    9          /dev/cua0       9600
3    9          /dev/cua2       9600
#

```

Once more, the number of lines below must be the same as number of TNCs in use. One line for each TNC. I think most of it is self-explanatory. MultCh will normally be 1. If you use DRSI-card, you can use a value from 0 to 7. If you use BPQ, the first TNC must have MultCh 0, the next must have MultCh 1 and so on. This is very important. If you have a KAM in host-mode, you must use 1 for VHF and 2 for HF. When using kernel AF_AX25 socket in Linux, MultCh is the interface name (eg: ax0). MxBloc decides how many kb will be forwarded one way to another BBS before the forward is reversed. Leave this value at 10 for VHF/UHF and a little smaller for HF. Type host-mode depends on your TNC, and if you use BPQ :

```

# TNC      : Number on TNC in use. Use 0 for file-forward !
# NbCh     : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com      : Number of the COM-port. Com1, Com2 etc.
# MultCh   : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#           With BPQ first TNC must have MultCh 0, the next 1, etc.
#           When using kernel AF_AX25 socket in Linux, MultCh is the
#           interface name (eg: ax0)
# Paclen   : PACLEN on this TNC.
# Maxframe : The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.

```

```

#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode. Must use ESSKAM driver.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1     7   1   1     230   4     1    10    30/60  UDYW 433.650
2     1   3   1     80    2     1     5    17/30  GDW  15/20m
#

```

At the end of the file, you can specify one (or more) callsigns and SSIDs for some channels. You might use this for special callsigns on forward-channels. If you don't need this (you probably don't, at least not the first times..), place an # in front of the line. This only works with WA8DED interface.

```

# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1     2   LA1B-1      B
#
# End of file.
#

```

2. Advanced setup, with more details.

The BBS may use a special port-multiplexer so that you can have up to 4 TNCs per COM-port.

For DosFBB you may also use other drivers like COMBIOS, MBBIOS, FBBIOS etc. These drivers must be loaded BEFORE the BBS-program. You can include the drivers in APPEL.BAT for convenience. It is possible to use telephone-modem with the BBS. In that case the number of TNCs include the modem. The FBBIOS-driver must be loaded before the BBS starts.

For WinFBB you do not need COMBIOS, MBBIOS, ESS or FBBIOS. They are all included in FBBCOMM.DRV (interface 6).

Example for WinFBB:

Standard PORT.SYS with 1 TNC (COM1) with WA8DED-hostmode, FBBCOMM.DRV-driver, 4 channels, 1 channel for outgoing forward, 1 channel reserved for BBSs with callsign LA1B-1 :

```

# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer

```

```

# there can be up to 4 TNCs per port.
#
#Ports TNCs
1      1
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#             7 = TCP/IP.  Needs WINSOCK.DLL.  Put port-address as 17.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll (only WinFBB32)
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB.  Interface 6
#         replaces both.  (FBBCOMM.DRV).  Neither ESS nor FBBIOS can be
#         used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux.  Can work via serial port (D), via AX25 domain
#               socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#
# Address      : Address of port in hexadecimal (Needed for multiplexer).
#               In LinFBB:
#               Address is the device name (/dev/cua0).
#               Be sure you have the rights to access to the device (rw-rw-rw-).
#               When using kernel AF_AX25 socket, address is not used.
#               When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud         : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)   Baud
1      1          3F8             9600
#
# TNC          : Number on TNC in use.  Use 0 for file-forward !
# NbCh         : Number of channels I want to use in the TNC.
#               Maximum available channels depend on firmware.
# Com          : Number of the COM-port. Com1, Com2 etc.
# MultCh       : Number of channel if port-multiplexer is used, otherwise 1.
#               In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#               With BPQ first TNC must have MultCh 0, the next 1, etc.
#               When using kernel AF_AX25 socket in Linux, MultCh is the
#               interface name (eg: ax0)
# Paclen       : PACLEN on this TNC.
# Maxframe     : The maximum nb of frames the TNC will send at a time.
# NbFwd        : Number of channels for OUTGOING forward at same time.
# MxBloc       : Size of forward-block in kb.
# M/P-Fwd      : Minute of the hour for start of forward, and period
#               (how many minutes between each forward-start).
# Port mode, one of these:
#               B : BBS-mode.
#               G : "Guest"-mode.
#               U : Normal-mode.

```

```

# Type host-mode, one of these:
#       D : WA8DED
#       K : KAM hostmode.  Must use ESSKAM driver.
#       P : PK-232
#       Q : BPQ v 4.x
#       T : Ethernet/TCP-IP
#       X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#       L : Send unproto beacon after each arriving mail.
#       M : Telephone-modem.
#       Y : Yapp allowed on this QRG.
#       W : Gateway allowed TO this QRG.
#       R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1    4    1    1      230    4      1     10    30/60  UDYL  433.650
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1    1    LA1B-1      B
#
# End of file.
#

```

Same example for LinFBB (Linux):
Standard PORT.SYS with 1 TNC (COM1) with WA8DED-hostmode,
4 channels, 1 channel for outgoing forward, 1 channel
reserved for BBSs with callsign LA1B-1 :

```

# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
1      1
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#             as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#             7 = TCP/IP.  Needs WINSOCK.DLL.  Put port-address as 17.
#             TNC-emulation is T (see below)
#             8 = TFWin.dll (only WinFBB32)
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB.  Interface 6
#         replaces both.  (FBBCOMM.DRV).  Neither ESS nor FBBIOS can be
#         used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux.  Can work via serial port (D), via AX25 domain
#                 socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#

```



```

#           3 = Telephone-modem with FBBIOS
#           4 = DRSI card with driver
#           5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#           as stated in INIT.SRV, if any..
#
# Address    : Address of port in hexadecimal (Needed for multiplexer).
#              In LinFBB:
#              Address is the device name (/dev/cua0).
#              Be sure you have the rights to access to the device (rw-rw-rw-).
#              When using kernel AF_AX25 socket, address is not used.
#              When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud       : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1      9          /dev/cua0      9600
#
# TNC        : Number on TNC in use.  Use 0 for file-forward !
# NbCh       : Number of channels I want to use in the TNC.
#              Maximum available channels depend on firmware.
# Com        : Number of the COM-port. Com1, Com2 etc.
# MultCh     : Number of channel if port-multiplexer is used, otherwise 1.
#              In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#              With BPQ first TNC must have MultCh 0, the next 1, etc.
#              When using kernel AF_AX25 socket in Linux, MultCh is the
#              interface name (eg: ax0)
# PacLen     : PACLEN on this TNC.
# Maxframe   : The maximum nb of frames the TNC will send at a time.
# NbFwd      : Number of channels for OUTGOING forward at same time.
# MxBloc     : Size of forward-block in kb.
# M/P-Fwd    : Minute of the hour for start of forward, and period
#              (how many minutes between each forward-start).
# Port mode, one of these:
#              B : BBS-mode.
#              G : "Guest"-mode.
#              U : Normal-mode.
# Type host-mode, one of these:
#              D : WA8DED
#              K : KAM hostmode.  Must use ESSKAM driver.
#              P : PK-232
#              Q : BPQ v 4.x
#              T : Ethernet/TCP-IP
#              X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#              L : Send unproto beacon after each arriving mail.
#              M : Telephone-modem.
#              Y : Yapp allowed on this QRG.
#              W : Gateway allowed TO this QRG.
#              R : Modem port allowed in Read-only mode.
# Freq.      : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacLn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      4      1      1      230      4      1      10      30/60  UDYL  433.650
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1      1      LA1B-1      B
#

```

```
# End of file.
#
```

Same example for DosFBB:

Standard PORT.SYS with 1 TNC (COM1) with WA8DED-hostmode,
ESS-driver, 4 channels, 1 channel for outgoing forward, 1 channel
reserved for BBSs with callsign LA1B-1 :

```
# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
1      1
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#             7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll (only WinFBB32)
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
# replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
# used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#               socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
# Address      : Address of port in hexadecimal (Needed for multiplexer).
#               In LinFBB:
#               Address is the device name (/dev/cua0).
#               Be sure you have the rights to access to the device (rw-rw-rw-).
#               When using kernel AF_AX25 socket, address is not used.
#               When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud         : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1      1      3F8      9600
#
# TNC          : Number on TNC in use. Use 0 for file-forward !
# NbCh         : Number of channels I want to use in the TNC.
#               Maximum available channels depend on firmware.
# Com          : Number of the COM-port. Com1, Com2 etc.
# MultCh       : Number of channel if port-multiplexer is used, otherwise 1.
#               In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
```

```

#           With BPQ first TNC must have MultCh 0, the next 1, etc.
#           When using kernel AF_AX25 socket in Linux, MultCh is the
#           interface name (eg: ax0)
# Paclen   : PACLEN on this TNC.
# Maxframe: The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode. Must use ESSKAM driver.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1    4    1    1      230    4      1     10    30/60  UDYL  433.650
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1    1    LA1B-1      B
#
# End of file.
#

```

Note that the differences in the three PORT.SYS files above (for WinFBB, LinFBB and DosFBB) is ONLY the Interface and (for LinFBB:) the address !

In the next examples, all comment-lines are removed, for clarity.

One example for WinFBB:

Now an example of a file with 4 ports, 13 channels, FBBCOMM.DRV, with a port-multiplexer on COM 1. Port 1 has limited access (guest-user), port 2 and 3 allows YAPP-transfer and port 4 uses a PK-232. Port 2 and 3 also allows unproto-beacon-lists.

```

# FBB7.00
#
#Ports TNCs
1      4
#
#Com Interface Address (device)  Baud
1      6          3F8          9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
1    4    1    1      230    4      0      1    00/60  GD   433.650

```

```

2      4      1      2      230      4      1      5      10/30      UDYL      144.675
3      4      1      3      230      3      4      10     20/60      UDYWL      433.650
4      1      1      4      80       1      1      5      15/30      BP       15/20m
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#
# End of file.
#

```

Same example for LinFBB (Linux):

Now an example of a file with 4 ports, 13 channels, with a port-multiplexer on COM 1. Port 1 has limited access (guest-user), port 2 and 3 allows YAPP-transfer. Port 2 and 3 also allows unproto-beacon-lists.

This example is really no good for LinFBB, as port 4 uses a PK-232 which is not supported by LinFBB (yet ?). But if it ever will be, it will probably look something like this:

```

# FBB7.00
#
#Ports TNCs
1      4
#
#Com Interface Address (device)      Baud
1      9      /dev/cua0      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode      Freq
1      4      1      1      230      4      0      1      00/60      GD      433.650
2      4      1      2      230      4      1      5      10/30      UDYL      144.675
3      4      1      3      230      3      4      10     20/60      UDYWL      433.650
4      1      1      4      80       1      1      5      15/30      BP       15/20m
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#
# End of file.
#

```

Same example for DosFBB:

Now an example of a file with 4 ports, 13 channels, ESS-driver, with a port-multiplexer on COM 1. Port 1 has limited access (guest-user), port 2 and 3 allows YAPP-transfer and port 4 uses a PK-232. Port 2 and 3 also allows unproto-beacon-lists.

```

# FBB7.00
#
#Ports TNCs
1      4
#
#Com Interface Address (device)      Baud
1      1      3F8      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode      Freq
1      4      1      1      230      4      0      1      00/60      GD      433.650
2      4      1      2      230      4      1      5      10/30      UDYL      144.675
3      4      1      3      230      3      4      10     20/60      UDYWL      433.650
4      1      1      4      80       1      1      5      15/30      BP       15/20m
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#

```

```
# End of file.  
#
```

Note that the difference in the 3 PORT.SYS files above (for WinFBB, CFBB and DosFBB) is ONLY the Interface and /for LinFBB:) the address !

Next an example of a file with 4 ports, 13 channels.
This one is for DosFBB, but it is identical for WinFBB, with the exception of interface, which must be changed to 6 for WinFBB.

```
# FBB7.00  
#  
#Ports TNCs  
4      4  
#  
#Com Interface Address (device)  Baud  
1      1      3f8      9600  
2      1      2f8      9600  
3      1      3e8      9600  
4      1      2e8      9600  
#  
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq  
1      4      1      1      230  4      0      1      00/60  GD   433.650  
2      4      2      1      230  4      1      5      10/30  UDY  144.675  
3      4      3      1      230  3      4      10     20/60  UDYW 433.650  
4      1      4      1      80   1      1      5      15/30  BP   15/20m  
#  
#TNC Nbs Callsign-SSID Mode  
#1      1      LA1B-1      B  
#  
# End of file.  
#
```

Same example for LinFBB (Linux):

```
# FBB7.00  
#  
#Ports TNCs  
4      4  
#  
#Com Interface Address (device)  Baud  
1      9      /dev/cua0      9600  
2      9      /dev/cua1      9600  
3      9      /dev/cua2      9600  
4      9      /dev/cua3      9600  
#  
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq  
1      4      1      1      230  4      0      1      00/60  GD   433.650  
2      4      2      1      230  4      1      5      10/30  UDY  144.675  
3      4      3      1      230  3      4      10     20/60  UDYW 433.650  
4      1      4      1      80   1      1      5      15/30  BP   15/20m  
#  
#TNC Nbs Callsign-SSID Mode  
#1      1      LA1B-1      B  
#  
# End of file.  
#
```

2.1 For BPQ:

Now an example of a file with 2 ports with BPQ-node and 4 channels on each port. Note that MultCh starts with 0 on first port !

This one is 100% identical for DosFBB and WinFBB !
This one is not possible for LinFBB as BPQ cannot be run in Linux.
BPQ must be correctly installed for DOS or WINDOWS (respectively) before
it can be used with FBB.

```
# FBB7.00
#
#Ports TNCs
1      2
#
#Com Interface Address (device)  Baud
8      2      0      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
1      4      8      0      230   4      1      1      00/60  UQY  433.650
2      4      8      1      230   4      1      5      10/60  UQY  144.675
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#
# End of file.
#
```

Note that I use COM8. This is because this is a "virtual" port, and if I use
for example COM2, I will not be able to use COM2 for other purposes.

2.2 For DRSI:

Next an example of a file with 1 DRSI-card and 8 channels, max 1 channel on
HF. As for BPQ, this one is 100% identical for DosFBB and WinFBB !

```
# FBB7.00
#
#Ports TNCs
1      2
#
#Com Interface Address (device)  Baud
8      4      0      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
1      8      8      0      230   4      1      10      00/60  UDY  433.650
2      1      8      1      80     2      1      10      30/60  GDW  HF
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#
# End of file.
#
```

Same example for LinFBB (Linux):

A file with 1 DRSI-card and 8 channels, max 1 channel on HF.

This example needs z8530drv by DL1BKE properly installed!

```
# FBB7.00
#
#Ports TNCs
1      2
#
#Com Interface Address (device)  Baud
8      9      ****      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
```

```

1      8      8      scc0  230   4      1      10      00/60   XUY  433.650
2      1      8      scc1   80   2      1      10      30/60   XGW   HF
#
#TNC Nbs Callsign-SSID Mode
#1      1      LA1B-1      B
#
# End of file.
#

```

Last, SM6TKY's own (big) setup for Linux :

Standard kernel AF_AX25 socket, 7 interfaces, 42 channels, 1 interface for file forward, 1 interface to a KISS TNC (ax0), 2 interfaces to a PA0HZP compatible SCC card (scc0 & scc1), 1 interface to Pseudo-TTY loopback for LinuxNode (ax2), 1 interface for EtherBPQ (bpq0), and 1 interface to NET/ROM in kernel (nr1).

```

# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
1      6
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#              4 = DRSI
#              5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#                  as stated in INIT.SRV, if any..
#              6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#              7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#                  TNC-emulation is T (see below)
#              8 = TFWin.dll (only WinFBB32)
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
#          replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
#          used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#                  socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#              2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#              3 = Telephone-modem with FBBIOS
#              4 = DRSI card with driver
#              5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#                  as stated in INIT.SRV, if any..
# Address    : Address of port in hexadecimal (Needed for multiplexer).
#              In LinFBB:
#              Address is the device name (/dev/cua0).
#              Be sure you have the rights to access to the device (rw-rw-rw-).
#              When using kernel AF_AX25 socket, address is not used.
#              When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#

```

```

#Com Interface Address (device)  Baud
1    9          ****          9600
#
# TNC      : Number on TNC in use.  Use 0 for file-forward !
# NbCh     : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com      : Number of the COM-port. Com1, Com2 etc.
# MultCh   : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#           With BPQ first TNC must have MultCh 0, the next 1, etc.
#           When using kernel AF_AX25 socket in Linux, MultCh is the
#           interface name (eg: ax0)
# Paclen   : PACLEN on this TNC.
# Maxframe: The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode.  Must use ESSKAM driver.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
0    0    0    0      0      0      0      0      00/01 ---- File-fwd.
1    8    1    ax0    250    7      2     10     30/01 XUWYL ax0
2    8    1    scc0    250    7      2     10     30/01 XUWYL scc0
3    8    1    scc1    250    7      2     10     30/01 XUWYL scc1
4    8    1    ax2    250    7      2     10     30/01 XUY   Loopback
5    8    1    bpq0    250    7      2     10     30/01 XUWYL bpqether
6    2    1    nr1     250    7      2     10     30/01 XUY   netrom
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1    2    XXXXX-1      B
#
# End of file.
#

```

4.5 INITTNCx.SYS:

```

-----
INITTNCx.SYS (\FBB\SYSTEM).

```

This file is also in the SYSTEM-directory. When the BBS starts, it sends some standard parametres to the TNC, like PACLEN, MAXFRAME etc. These

parametres are in PORT.SYS. But in addition to this we may send some more parametres to each TNC. We send parametres to TNC 1 with the INITTNC1.SYS, to TNC 2 with INITTNC2.SYS etc. Here is just one example from LA1B BBS : (you do not need to use the same parametres)

```
C MAIL v LA7QR
U 0
N 10
M IU
P 64
T 40
```

C MAIL V LA7QR means that unproto mail-beacon will be sent to MAIL via the digipeater with callsign LA7QR. You may use just C MAIL if you do not want to send via digipeaters. U 0 means that the TNC will not send any TNC-message to user when he connects to the BBS. You should use this parametre.

N 10 means that RETRY will be set to 10.

M IU means that that all I and UI frames will be shown in monitor-windows. P 64 means a p-persistence value of 64. This is normal.

T 40 sets a TXDELAY of 40. This value depends on Transceiver in use. So, you can use any suitable parametres in this file. You need one file for each TNC.

The commands sent to the TNC, are the commands described in the documentation for the firmware you use, or in the documentation for PK*-232, DRSI, BPQ etc.

If you are using BPQ, DRSI, PK-232, KAM or TFPCX/TFPCR, look in the appendix for special commands for them.

4.6 MAINTn.SYS:

MAINTx.SYS (\FBB\SYSTEM).

This file is in the SYSTEM directory. This works exactly like INITTNCx.SYS, but the parametres are sent to the TNC when the BBS is shut down. The file can look like this:

```
Y 1
U 1 BBS ($c) was shut down for service $d $T.
```

Y 1 means that only 1 station can connect my TNC now.

U 1 means that the TNC will send this message to any user that connects after the BBS has shut down.

If you use a PK-232, the same file would look like this:

```
UR1
CTBBS ($c) was shut down for service $d $T.
```

The commands sent to the TNC, are the commands described in the documentation for the firmware you use, or in the documentation for PK-232, DRSI, BPQ , KAM etc.

4.7 BBS.SYS:

BBS.SYS (\FBB\SYSTEM).

This file is in SYSTEM directory. In this file you MUST put the callsign of every BBS you will forward directly to. There must be 80 (from 1 to 80) lines, even if you do not use them all. The format is very critical, so do NOT change line-arrangement !

All the BBSs you forward to (the callsigns after A in FORWARD.SYS) must also be in BBS.SYS. Before you start your BBS for the first time, you should remove all not-used callsigns in BBS.SYS, and put in the file only the BBSs that you forward to in your FORWARD.SYS. Later it is recommended that you do NOT replace one callsign with another, as messages then will go to the wrong BBS.

This file should speak for itself.

```
#####
# BBS.SYS for FBB 7.00  #
# for LA6CU BBS by LA6CU #
#####
#
# File containing all BBSs we forward to.
#
# Make sure that NO numbers are removed.
#
01 LA1B
02 F6FBB
03 AMSAT
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80

4.8 FORWARD.SYS:

FORWARD.SYS (\FBB\SYSTEM).

I have divided this section in two parts:

- 1 : Simple instruction (for first-time setup).
- 2 : Advanced setup, with more details.
 - 2.1 : Line types.
 - 2.2 : Definition of a forward-file.
 - 2.3 : Include-files.
 - 2.4 : Automatic forward to a file.
 - 2.5 : End-of-file DOS-command specific to a port.

Simple instruction (for first-time setup).

This file is in SYSTEM directory. In this file all necessary data for forwarding of messages are held. First I will give an example of a file that contains the minimum of data that MUST be in FORWARD.SYS if you have forward

to only 1 other BBS, called LA2D in this example.

The file is organised in blocks, one block for each BBS we forward directly to. A block starts with A (callsign) and ends with ----- We can have as many blocks as we want in FORWARD.SYS.

```
A LA2D
#
P C
#
C C LA2D
#
B LA2D
F LA2D
#
-----
```

All lines starting with a '#' are comments-lines, and are ignored by the BBS. The first line must always start with the letter A, and next the callsign of the BBS we want to forward to. Here: A LA2D

Then there must be a line telling on which port to start the forward. This line must be there even if we only have 1 port.

In this example I use port C, which is the same as TNC 3 in my PORT.SYS.
Command: P C

So I will send the connect-command to the TNC. That line must start with a C before the actual connect-command. So if I call LA2D directly, I need to write C C LA2D. (Only one C will not work). And lastly I must write which messages I shall forward to that BBS. In this case I type only B LA2D, which means all messages with @ LA2D, and F LA2D, which means all messages TO LA2D. The forward-file must end with a string of ----- . If you have more than one BBS to forward to, you write all data for that BBS in a similar block in this same FORWARD-SYS-file (starting with A, and ending with ----) right after this block for LA2D.

You can also use one include-file for every BBS, but we will look at that later in this sysop-documentation.

Well. I expect you do not want to forward only those messages that are TO LA2D or @ LA2D (in this example), I expect you want to forward bulletins also. Just include some lines starting with G, like G EU, G ALL, G WW etc in the file. G EU means all bulletins addressed to @ EU. So may want to add this in each file, like this:

```
A LA2D
#
P C
#
C C LA2D
#
B LA2D
F LA2D
#
G EU
G ALL
G WW
-----
```

OK. Next you may want to forward messages through LA2D, that is not really for LA2D, but that LA2D can forward on to this other BBS. These other BBSs can be included in lines starting with a B, like B LA1G, B LA40, B LA5G etc.

like this:

```
A LA2D
#
P C
#
C C LA2D
#
B LA2D
F LA2D
#
B LA1G
B LA4O
B LA5G
#
G EU
G ALL
G WW
-----
```

Also, if you want to forward ALL messages for other SM-BBSs this way, you can use wildcards (as in DOS) like B SM*, B SK*, B OH* etc. like this:

```
A LA2D
#
P C
#
C C LA2D
#
B LA2D
F LA2D
#
B LA1G
B LA4O
B LA5G
#
B SM*
B SK*
B OH*
#
G EU
G ALL
G WW
-----
```

Well. Now the file is almost complete for most needs. We may want to add the possibility to forward (on VHF or UHF) via several nodes. This is very easy, as we just add another C C (callsign) line for each node we have to call. If I must call first my local node LA7QR, next the local node of LA2D (LA1EAX-7), and lastly LA2D, the complete file for LA2D will now look like this:

```
A LA2D
#
P C
#
C C LA7QR
C C LA1EAX-7
C C LA2D
#
B LA2D
F LA2D
```

```

#
B LA1G
B LA4O
B LA5G
#
B SM*
B SK*
B OH*
#
G EU
G ALL
G WW
#
R
#

```

I put in an R at the end. This forces my BBS to call LA2D BBS and ask him for forward from him to me (reverse forward), even if I have nothing to forward to him.

Well, this was a simple forward-file. There is a lot more we can do with this file, so you should read Advanced setup very carefully, I think you will find all you need there. But you now have learnt the basics..

Advanced setup (with more details).

The FORWARD.SYS file defines the setup of the various adjacent BBS's to which we will route the mail. The file is organized in blocks, each block being specific to a BBS. Only the adjacent BBS's should be defined.

This file and its includes are compiled at the BBS start-up and after editing with the built-in editor, or return from the DOS Shell (AltF9 in DosFBB) if one of the files has been modified.

If the FORWARD.SYS file or one of its includes has been modified all the active messages are scanned, and new routings will be set according to the updated file(s). The scanning can be stopped with AltF3. (Can be convenient if the modification does not impact the routing.)

Every line of the file begins with a letter which defines the function of the line. Any line beginning with a # is regarded as a comment and is ignored.

Line types :

A (callsign)

All blocks must start with a line like this. The callsign is the callsign of the BBS we will forward to.

B (callsign)

All blocks must include at least one line like this. Personal messages addressed to somebody with address '@ (callsign)' will be forwarded.

C C (callsign1) [V DIGI1 DIGI2 ETC...]

The complete line of connection (C callsign V digi1 digi2) must follow the letter C. In most cases, there will be two letters C successively, with a space between them. The letter V, specifying a string of repeaters is optional if PROMs TF4 (or equivalent) are used - see WA8DED.DOC. There may be several lines C, in the case of cascaded connections (use of THENET, as per an example). A line of conditional connection (enclosed between an IF and an ENDIF) allows calls on preset times. Up to 8 lines are allowed, each line can

hold up to 80 characters.

In case of KA-node and X-connect, we can send C X (callsign). When connected to a node, all command after the first C is actually sent to that node. Like here, the X is sent too.

Only for LinFBB and WinFBB:

To forward with Telnet the syntax is:

C C <callsign> <IP address> [port]

IP-address is required, port is optional (default 23).

D (command)

Command to be performed by the DOS (transceiver frequency set, for example). For WinFBB:

A separate DOS-window is opened, and the command performed there. The window is closed when the command has been executed. In WinFBB the D (command) and X (command) is identical.

For DosFBB and LinFBB:

The screen is saved before the command, and restored when done, thus enabling screen access for this command. When running BAT files, a COMMAND.COM is automatically first sent. That needs some more memory.

DC (command)

Same as D-command above, but executed when the connection is done. This allows to change parameters or program a function when the other BBS is connected. When the connection is made from another BBS, the LC, DC and XC are executed if the BBS exists in forward file.

ELSE

Complementary conditional lines to the previous IF.

ENDIF

End of conditional lines defined by an IF.

F (callsign)

Callsign which will be automatically routed (except if a routing has been specified) to the BBS which has been previously stated by a B line (must follow a B line). There may be several F lines following a B line. If this line is not preceded with a B line, then the bulletins will be routed on their destination.

G (route)

Group routing indicator for bulletins (EU for example). This indicator applies only to bulletins. The private messages are sent according to the BBS indicator (see line B). There may be several routing indication lines per adjacent BBS.

H (route)

Route definition using the hierarchical addressing mode. The wildcards (* or ?) must be used to replace the missing part of the address as a function of the destination. H-routing should be used as much as possible.

IF C1 ... IF C9

Conditional test of the route selection. The value for "n" is a figure, from 1 to 9, depending upon the selection. The first selection is 1, and the maximum is 9. The lines located between the IF Cn and the ENDIF will be taken into account only if the current selection is n. If the connection has not been successful (remote BBS prompt not received), the selection number will be incremented, and the block will be tested again if the port is still the same. If the port is different, the block will be tested during the tests related to the new port. If the connection is successful, the selection number is reset at 1 for the next call, otherwise the selection number is

incremented. If all the selections failed, the selection number is reset at 1, the call is given up, and the system manages the next block.

IF D xx

Conditional test of the days of the week. This line defines a set of lines which will be taken into account if the expression which defines the day(s) following the IF is true. 0 corresponds to Sunday and six to Saturday. The expression 'IF D 0,2,4-6' is true for Sunday, Tuesday, and Thursday to Saturday. There can be several embedded IF's (indentation recommended), they shall never expand outside a block. The ENDIF defines the end of the conditional block.

IF H xx

Conditional test of the hour (IF). This line defines a set of lines which will be taken into account only if the hour period following the IF is TRUE. The hour period written in the format 0,4,12-23 is true at 0h, 4h, and for the hours from 12h to 23h. There may be several IF conditions embedded (indentation should be used), but never outside of the blocks.

IF FREE

Conditional test for the availability of a port. The test will be true if none of the channels for the port is in use. This test can be a necessity before the autoqsy of a transceiver as for an example. There can be several embedded IF's (indentation is recommended), they shall never expand outside a block. The ENDIF, defines the end of the conditional block.

IF MAN

Conditional test only true if forward has been asked manually by F4, FR or FW commands.

IF N (number)

Tests on number-of-day in the month (1-31)

J (data means 7+ here)

- J 0 : no data sent
- J 1 : data is sent (default)
- J 2 : personal data is sent
- J 3 : only data is sent (exclusive)

K Channel

Forces forwarding to start on the specified channel. If the channel is busy, forwarding will not start.

L (command) [value]

Parametres programming for the TNC. The parameters will be assigned to the related channel. The syntax is the one of the TNC hostmode. Only the command B (Paclen = packet size) has been supplied in all cases. B parameter must be used with all TNCs (including KAM) to specify a paclen. The paclen is always managed by the BBS software, as the different host modes are transparent to the paclen. The PACLEN of the KAM should be 0/0 (256) to make the BBS paclen work in all circumstances.

LC (command) [value]

Same as L-command above, but are executed when the connection is done. This allows to change parameters or program a function when the other BBS is connected. When the connection is made from another BBS, the LC, DC and XC are executed if the BBS exists in forward file.

N (number)

Assigns the standard forward protocol (MBL/RLI) for the specified BBS.

- 1 = FBB forwarding allowed
- 2 = FBB+BIN forwarding
- 4 = FBB+BIN+Resume

8 = XFWD
Ex: N 15 = Sum of all protocols (default)

O (minutes)
Defines a timeout for the specified BBS. Timeout is expressed in minutes.

P (port)
Selects the port on which the connection is to take place. The port is defined by a letter from A to H for the ports 1 to 8.

Q
Prevents the reverse forwarding from my BBS to take place.

R
Specifies that a call should be made to the adjacent BBS even if there is no mail in queue, so as to trigger the reverse forwarding.

S Connect Failure Busy
It is a set of supplementary test in the case of sequential connections (TheNet network for example). The 3 tests should be defined. The tests "Connect" "Failure" and "Busy" are already defined in the software. Ex.: S Conne Echec Occup. The software will interpret "Conne" like Connect, "Echec" like Failure, and "Occup" like Busy. This allows to comply with nodes messages which are not those commonly used by TheNet.
Up to 4 S lines (set of tests) can be defined in a forwarding block. S line (choices of forward) now checks default as a last choice. Up to four previous choices are checked as declared before. Warning, these tests are case independent.

T options
Definition of the maximum size for a file authorized to be forwarded, type of messages and strategy. This line can be made conditional (between an IF and an ENDIF).

T P : Send only personal messages
T S : Send smaller first (instead of older first)
T 2 : Send only messages less than, or equal to, 2 KB.
T D : Disable the feature of sending only 1 message from each
callsign in each forward-block

It is possible to combine one or more of these option like :

T P S 2 : means only private less or equal to 2 KB, smallest first.

In all cases, personal messages are forwarded first, bulletins next.

U ROUTE1 ROUTE2 ... (up to eight routes)
Give the priority when forwarding bulletins. Bulletins @ROUTE1 will be forwarded first, then bulletins @ROUTE2, then other bulletins. It may be up to eight specified routings. Private mail (including P, T and A messages) are always forwarded first.

V (text)
Text sent on connection of the called BBS. (Mandatory on some TCPIP BBS to declare the protocol in use). Allows sending callsign and password when connecting to a phone BBS.

W
Command W allows to skip one or more prompts (default one) before starting a forward session (mainly used when "Linked to ..." gives more than one prompt)

X (command)

Command to be executed by DOS (Selection of a frequency, as for an example).

For WinFBB:

A separate DOS-window is opened, and the command performed there. The window is closed when the command has been executed. In WinFBB the D (command) and X (command) is identical.

For DosFBB and LinFBB:

The screen is not saved before the command, thus, the latter should not access the screen. This command is identical to the D command, but it is transparent and faster. When running BAT files, a COMMAND.COM is automatically first sent. That needs some more memory.

XC (command)

Same as X-command above, but executed when the connection is done. This allows to change parameters or program a function when the other BBS is connected. When the connection is made from another BBS, the LC, DC and XC are executed if the BBS exists in forward file.

Y

Automatic time update of PMS having the clock ability. (if SID looks like [PMS-3.0-C\$]). PMS is set with the local time.

Z route

NTS routing indicator for NTS messages. This routing indicator applies only to messages of "T" type. Wildcards can be used to define masks. There may be several routing indication lines per adjacent BBS.

----- (string of dashes)

Defines the end of the block for the adjacent BBS. The block begins with an A line.

! Line

Defines an exception. Will be used with lines of types B, F, G, H. The exception should always be set before the true condition statement. An exception is always from the same type statement (you cannot have a !F exception to a B line).

Ex:

! B F6FBB Except for F6FBB BBS

B F* Forward to all the BBS which begin with an F

The number of the exceptions is not limited.

< File

Include the file specified. The name of the file should be in the directory SYSTEM or in one of its subdirectories.

#

Comment line.

Definition of the forward file :

A forward file is made of a main file including a set of blocks, each block corresponds to an adjacent BBS, that is to say that can be connected.

Definition of the minimum block :

A block will be an included file in the FORWARD.SYS file. It is always made with a first line beginning with an A, defining the name of the adjacent BBS, and of a last line made of a string of dashes. Inside this block, there will be at least one P line (port used) a C line (connection request), and a B line (destination BBS). An F line with the SYSOP callsign may follow the B, enabling the automatic routing for his mail.

```

A FC1CDC
P A          Port A
C C FC1CDC-1 F6FBB-5 connection line
B FC1CDC BBS  desserved by FC1CDC
F FC1CDC      callsign automatically forwarded to FC1CDC
-----

```

This type of block is well suited for PMS of BBS which do not support automatic mail forwarding.

Example of a more complex block :

```

A FC1HAQ
#
G EU          FC1HAQ will receive
G FRA         the bulletins for those
G WW          various routings.
G AMSAT
F AMSAT       bulletins to AMSAT are also marked.
#
H *.NA        All the H. routes which end
H *.SA        with .NA, .SA
H *.EU        and .EU
#
O 5           5 minutes timeout.
#
IF C1         1st choice parameters
P C
L B 250       Paclen is 250 bytes
L N 4         Retry 4
C C FC1HAQ-1 F6FBB-7 019301
ENDIF
IF C2         2nd choice parameters
P C
IF 8-17       from 8h to 17h
T 5           files of 5 kb Max.
ENDIF
L B 250       Paclen is 250 bytes
L N 4         Maxframe is 4
C C FC1HAQ-1 F6FBB-7 019002
ENDIF
IF C3         3rd choice parameters
P A           port A is used
T P           private mail only
C C F6FBB-2   Connection of the 1st Thenet.
C C FF6KUB-2  Connection of the 2nd Thenet.
C C FC1HAQ-1  Final connection.
ENDIF
#
B FC1HAQ      Definition of the various
F FC1HAQ      BBS routed by FC1HAQ and
F F6CQU       of its SYSOPs and /or direct
#             specified users.
B F6ABJ
F F6ABJ       Other BBS and miscellaneous
#             callsigns forwarded by FC1HAQ.
B F5LO
F F5LO
#
B G*          other BBSs.
B ON*

```

```

B D*
B HB*
#
-----
                                End of the FC1HAQ block

```

Include-files.

It is possible to use include-files with FBB. In this way we can have a very small FORWARD.SYS, with only the callsign of the BBSs to forward to, and then we have another directory with all the forward-blocks. One file for each BBS, for example. Just transfer one whole block (starting with A (callsign) and ending with -----) to file, and put the filename in FORWARD.SYS with a '<' in front of it, like this:

Example of FORWARD.SYS file with include-files.

```

#
# FORWARD FILE F6FBB
# -----
#
# UHF
#
< FWD\FE6BIG
< FWD\FC1GJC
< FWD\FF6KNI
< FWD\FC1HAQ
< FWD\F1EBV
< FWD\FD1JGK
< FWD\EA3BKZ
#
# PMS
#
< FWD\FD1CDC
< FWD\FD1MEU
< FWD\FD1OAC
< FWD\F8PD
< FWD\F6HED
< FWD\FE1JYH
< FWD\F1LMI
#
# HF
#
< FWD\LA1B
< FWD\LA6CU
< FWD\SV1IW
< FWD\4X1RU
< FWD\RS3A
< FWD\I7XGL
< FWD\EA8AML
< FWD\EA8RT
#
# VHF
#
< FWD\F6CDD
#
# FILES
#
< FWD\AMSAT
< FWD\MAIL
#
# Back to the park frequency.
#

```

```

P B
IF 0-3,21-23
    X ICOM A2E8 M14
ELSE
    X ICOM A2E8 M4
ENDIF

```

Example of an include file FWD\EA8RT

```

-----
A EA8RT
#
G EU
G ALL
G AMSAT
G WW
G EA
G EANET
G LATNET
#
P B
#
O 2
#
L T 20
L B 80
L O 2
L P 100
L F 2
L @T2 20
#
#
IF 7-22
    X ICOM A2E8 M6
    C C EA8RT-2
ENDIF
#
B EA8RT
F EA8RT
F E?1*
F E?4*
F E?7*
F E?8*
#
B E?1*
#
B E?4*
#
B E?7*
#
B E?8*
#
-----

```

Automatic forward to a file.

It is interesting to allow automatic routing of messages or bulletins to a file, for further processing, archive, or communication between BBSs which co-exist in the same computer. The communication can be made through files. The incoming mail file (see INIT.SRV) is tested once every minute; in case of

existence, the mail it contains is imported into the BBS.

On the other hand, a file can be defined in a forward block and updated once every 15 minutes. This file is defined in a block of forward identical to the forward between BBSs, the port being @ and the connection line given the name of the file created or updated. The name of the BBS (possibly dummy) used will be declared in BBS.SYS, of course.

Example of a forward to a file (forward to the file B:MAIL.FBB of my own private mail) :

```
A MAIL
P @
C B:\MAIL.FBB
B F6FBB
F F6FBB
-----
```

End-of-file DOS command specific to a port.

Each DOS command (D or X line) defined out of a forward block will be run only if the forwarding port matches with the last port selection.

Example :

You use the autoqsy capabilities of your HF transceiver with a special program named "ICOM". To be sure to return to your standby frequency, you must set your frequency at the end of the forward file, only if the forward applies to the HF port. You only have to select the port before running the DOS command.

```
      B xxxxxx
      #
-----
#                                     End of last forward block
P D                                     Select HF port (port D)
X ICOM A3E8 M12                       Send the DOS command ICOM with its
#                                     arguments if the file is scanned for
#                                     D port. IF and ENDIF are allowed.
```

4.9 BEACONx.SYS:

BEACONx.SYS (\FBB\SYSTEM).

This file is in the SYSTEM directory. This file makes the text that will be broadcast as mail-beacon from the BBS. The first line is a parametre that says how often to send the beacon. %15 means "send one beacon every 15 minutes". The next line says to what unproto-address the beacon will be sent. ! MAIL v LA7QR means send beacon to MAIL via LA7QR digipeater. There may be several !-lines for different paths for each port.

The next lines are the actual beacon-text. Here we can use variables like \$d \$T etc. We cannot use variables that are specific to one channel alone. You do not need to change anything else then the !-line in this file. But you will need one file named BEACON1.SYS for TNC1, one called BEACON2.SYS for TNC2 etc.

```
%15
! MAIL v LA7QR
```

```
$d $T <<< Mailbox $O $c >>> $N active messages.  
Messages for$W$Q
```

4.10 EPURMESS.INI:

This file is also in the main FBB-directory. This is the file that controls the lifetime of messages. Each night EPURMESS will be activated at the time specified in INIT.SRV. Normally you will not need to change this, but if you did any changes during installation, you must also change the same pathnames here. That goes for all lines that include a pathname.

Next, you decide if you want to save "killed" messages in the OLDMAIL-directory. If you do, write a 1 first, if you want to save old personal messages, and another 1 if you want to save old bulletins.

```
# Personal Bulletins (0=kill 1=archive in oldmail)  
0 0  
#
```

Next we have the actual parametres (in days) for the lifetimes of bulletins and personal messages. Let me just explain the letters in use here:

```
P=Personal message  
B=Bulletin  
N=Not read  
Y=Had been read  
F=Forwarded (forwarded to all BBSs that should receive this msg)  
X=Expired (still readable, but will not be forwarded to other BBSs)  
K=Killed (readable only by sysop)  
A=Archived (same as killed, but will be deleted at once, or moved to  
  OLDMAIL)
```

I think this is self-explanatory:

```
# Parametres in days:  
#  
# PARAMETRES FOR PERSONAL MESSAGES:  
#  
# PN -> PX (days after message is written)  
30  
#  
# PY -> PX (days after changed to Y)  
7  
#  
# PF -> PK (days after changed to F)  
7  
#  
# PX -> PK (days after changed to X)  
0  
#  
# PK -> PA (days after changed to K)  
0  
#  
# PARAMETRES FOR BULLETINS  
#  
# BN -> BX (days after the message is written)  
7  
#  
# B$ -> BX (days after the message is written)  
7  
#
```

```

# BY -> BX (days after changed to Y)
7
#
# BX -> BK (days after changed to X)
14
#
# BF -> BX (days after message is written)
14
#
# BK -> BA (days after changed to K)
0
#
#
# Automatic generated return-messages for non-forwarded messages.
# Each line may be max 255 characters long. Disable with only an 0.
#
# Timeout for valid forward-path (number of days + text, end with $W)
21 This message could not be forwarded, due to bad link.$W
#
# Unknown routing or wrong H-address (number of days + text, end with $W)
2 This message could not be forwarded, due to bad callsign or bad route.$W
#
# Max number of lines in a returned message:
10
#
-----
#
# At the end of the file, you can have special lifetimes for special messages.
# In the example below, messages that are TO LA6CU will be 365 days old,
# messages FROM system will be 0 days, messages @LA will be 365 days. I have an
# # in front of them, this means they are disabled right now.
# List of lifetime-tests
#
# To/from/via Callsign Days
@ VEST 365
@ LA 365
> LA6CU 365
< system 0
#
# End of this file.
#

```

4.11 SWAPP.SYS:

```
-----
```

SWAPP.SYS (\FBB\SYSTEM).

This file will reside in SYSTEM-directory. This file is used for automatically re-route messages. This file normally changes the @BBS-address, but can also change the TO-field. You might not need so many callsigns etc. here, but you should make one line with only your callsign after the @. In that case the @BBS will be removed from messages to users in your own BBS. For the first tries, it is enough to have one line starting with @, and then your BBS-callsign. You can fill in the rest later.

```

#####
# SWAPP.SYS for FBB 7.00 #
# for LA6CU by LA6CU #
#####
#
# File SWAPP.SYS is used to change @BBS-adresses, to-adresses etc

```



```

# > changes to-callsign
# < changes from-callsign
# @ changes @BBS
# Combinations also allowed, like      > LA8AK @ OZ2PAC.
#
# Other possibilities (recommended) is to delete own @BBS. That is done
# by putting own BBS-call in first column.
#
# Swapping may use hierarchy. For instance :
# @ F6FBB.FRA.EU @ F6FBB.FMLR.FRA.EU
#     Only the first full address will be swapped to the other
# @ F6FBB @ F6FBB.FMLR.FRA.EU
#     All F6FBB addresses will be swapped to the other
#
@ LA8AK LA9K
@ LA5IV LA9K
@ LA6XXX
@ HF EU
@ WWW WW
@ HFNET WW
@ LA9WO LA9H
@ TEST

```

4.12 INIT.SRV:

This file is the next file that FBB looks into during startup. This file will be found in the main FBB-directory. All basic data is collected from this file. Use your standard ASCII-editor to edit the following lines: (all lines starting with # are comments)

In WinFBB, INIT.SRV is created or updated with a small program called INSTWFBB.COM. This program may also be run inside WinFBB, when WinFBB is running, by clicking on Config at the top of the screen.

In LinFBB, INIT.SRV is created or updated with a small program called INSTALL.SH.

In DosFBB, INIT.SRV is created or updated with a small program called FBBSETUP.EXE.

This program is organized as pages of menus, and you can edit or change any topic in your configuration. It is no longer necessary to edit the INIT.SRV file manually.

The version of the software is now indicated in the first comment line. This will allow further automatic updates.

```

# FBB7.00
#
# Callsign and H-address (Do not include SSID !)
LA1B.#BRG.NOR.EU
#
# SSID for the BBS
0
#
# Qra locator
JP20RH
#
# Qth (variable $c)

```

Bergen
#

In the above lines you must replace the callsign and H-address with your own BBS-call and H-address. You also must replace the SSID, that is the number after the callsign. In LA1B-6, the 6 is the SSID. If you do not want any SSID, type 0 here. You also must replace the QRA-locator with your own locator (ask some other ham for help on this, if you don't know the BBSs QRA-locator. Do NOT use the value from this example-file ! You also must replace the QTH (don't make it too long...). If the QTH-name is short, you might add the QRA-locator here too, like I have done.

Next we have some filenames and path-names. These filenames and paths are the same as used in the installation-procedure. If you did no changes in this during installation, you will not need to change any of these now. Note the syntax, and note that directory-names MUST end with a \

```
# Directory for files used by the BBS
C:\fbb\system\
#
# Directory for messages
C:\fbb\MAIL\
#
# Directory for binary-messages
C:\FBB\BINMAIL\
#
# Directory for users DOS
# Up to 8 drives may be specified, starting with drive A:
# In this example drive A and B are not used, and therefore replaced
# by a * each.
*,*,C:\fbb\users\,*,*,*,*,*
#
# Directory for Yapp-files
C:\fbb\users\yapp\
#
# Directory for DOCS
C:\FBB\DOCS\
#
```

That was not so difficult ?

Next, check the first-name of sysop, and the callsign of sysop. Here are MINE :

```
# First-name of sysop
Per
#
# Sysops callsign
LA6CU
#
```

In the following line you CAN (not necessary) replace the ***** with a callsign. If you do, that callsign will receive a copy of all SYSOP-messages that arrives. You might want to put your own callsign there. You may also include more than one callsign, separated by space.

```
# Callsign (and route if needed) that shall have copy of messages
# addressed to SYSOP
# (** for no copy, FD1CDC@FD1CDC.FMLR.FRA.EU if route)
*****
#
```

No need to change this:

```
# Path and filename for import-file
\FBB\MAIL.in
#
```

Here you must first write the code for the kind of monitor you use, and then you must have a number for how many minutes you want before the screen should go blank, after no activity on the keyboard. In this example I have a CGA-monitor, and my screen will go blank after 5 minutes since I last touched the keyboard.

```
# Type video:0=CGA, 1=EGA, 2=VGA Time before screen-blank.
#
0 5
#
```

Complete log gives you a log-file for every week. If you write NO here, no log-file will be made. The log-file will appear in the directory named LOG. Direct video uses fast BIOS-calls if you answer OK. If you plan to use Desqview or Windows (with DosFBB, that is..), you must change it to NO. Indication of channels will normally be OK (that means that monitored information on screen will be more complete, with callsign of user before every frame from each user, etc).

```
# Complete log (OK or NO)
OK
#
# Direct video (OK or NO)
OK
#
# Indication of channels
OK
#
```

Next we have Test Mode (NO - OK). If you type OK, the BBS will start, but will not access the TNC(s). Use this the first times, to make sure that everything works fine. When you are sure that all is well, connect the TNC(s) and change this to NO. Restart the FBB, and now the TNCs will be used, as normal.

```
# Test-Mode (NO - OK)
OK
#
```

Forward type FBB is faster and more efficient than standard MBL-type forward. Use this whenever possible (OK). If you type NO, FBB-type forward will not be allowed. You may also use a parametre after OK/YES from this list (add the ones you want):

- 1 : A space is mandatory before the @ in a send message command
- 2 : The length of the fields of a hierarchical address is not tested to be 6 characters
- 4 : The header line of a message is not truncated to the space before the 79th character
- 8 : Header MBL/RLI
- 16 : If there is no BBS field, the callsign of the BBS is sent to the PMS
- 32 : Deletes the DATA messages sent to SYSOP
- 64 : Don't use the BID recovered from headers and use a new one
- 128: Accepts forwarding only from pre-declared BBS
- 256: WP Messages are not held.
- 512: XForwarding protocol has priority on FBB protocol.

1024: Generation of an alternate BID like F6FBB-12345 (for dual BBS site)
2048: Checksum unvalidated on XFwd.
4096: Test of callsigns is less strict. Allows all "callsign" as long as they have one figure (0-9) anywhere in the callsign.

Use (when possible) forward type FBB
OK
#

The same goes for compressed forward. Compressed forward is very much faster, and should always be used.
Parametre after "use compressed forwarding" give the options between compressed protocols :
1 = FBB compressed forwarding
2 = XFWD compressed forwarding
"OK 3" allow both FBB and XFWD.

Use (when possible) compressed forward
OK 3
#

Hroute update validation. If you type OK, the HROUTE look-up will be validated. This takes some tens KB of memory. A route without hierarchical extension will be changed to hierarchical route if found in the HROUTE.SYS database. This database is automatically updated while receiving headers.

HROUTE look-up and database update validation
OK
#

Information query. This ensures that the user will give his information. This will be helpful for the update of the white pages database. If "OK", the user will be prompted until he answers to the queries.

User MUST give Name, HomeBBS, Qth and ZIPcode
OK
#

Next we have 'Masque'. This is a default value that is given to all new users when they log on the BBS for the FIRST TIME. Select the things you want for all new users, and the values for them. If you f.ex. want all new users to get Paging, and be in Guest mode, add 32 and 64, and put the result 96 in the file. If you want none of the things listed here, put in a 0. The "see-all_mail" mask will be active only if the user has the "read-all-messages" security code (see behind).

First connection mask.
0 : disable mask.
1 : Excluded.
2 : Local.
4 : Expert.
8 : Sysop.
16 : BBS.
32 : Pagination on
64 : Guest.
128 : Modem.
256 : List/read all personal messages
512 : Request for unproto beacon-lists allowed
1024: Get list of new messages at each connect
Add up the values for your choices.
1568
#

Next we have 'Security-codes'. Here we give 3 values. The first says what ALL users are allowed to do. The next says what sysops are allowed to do when they connect the BBS, and the third says what a sysop is allowed to do after he has successfully performed the SYS-command. The same applies here as above; choose the values you want and add them up.

```
# Security-codes.
# All users can:
# 1  : Read all messages, including all personal messages.
# 2  : Kill all messages.
# 4  : Send SYS-command.
# 8  : Use remote-sysop commands (edit, forward etc).
# 16 : Edit labels in YAPP, FBBDOS, DOC.
# 32 : Can delete files in YAPP and FBBDOS.
# 64 : Have access to all gateways.
# 128: Run dos-programs (command DOS in FBBDOS).
# 256: Have access to the entire hard-disk.
# 512: Have access to command /A (stop BBS) and /R (reboot PC).
#
# All: Sysop: Sysop after successful SYS-command:
#      0      127      1023
#
```

Next we have warning-messages to sysop (mask):

```
# 1  : Less than 1MB in disk
# 2  : Error in system file (FORWARD, BBS, REJECT...)
# 4  : Server error/warning
# 8  : Ping-Pong warning
# 16 : No-route warning
# 32 : No NTS warning
# 64 : Message file not found
# 128 : Error in proposal
# 256 : Message rejected in remote BBS
# 512 : Message held in remote BBS
1023
```

Next we have the time for cleanup of messages. This should be done in a low-traffic hour, as the BBS is completely shut down during cleanup. Type also the timeout-values you want, the default values here should be OK. Put also in the number of hours (+ or -) between the PCs local-time and UTC.

```
# Time (hour) for cleanup of messages
# (Forced disconnection of all links at housekeeping time+15 min.)
02
#
#
# Time-out for normal use (minutes) and during forward (minutes)
15 5
#
```

Max download-size (pr period) for Yapp and via modem, in kbytes. 0 0 will disable it

```
#
100 500
#
```

```
# Hours +/- in relation to UTC
0
#
```

Type how many callsigns you want (maximum) in each mail-beacon (variable \$Q). If you precede the number by a B, also bulletins to local users (type BN or BY) will be broadcast in the mail-beacon. Like this:

```
# Number of callsigns in mail-beacon
B20
#
```

Number of lines in scroll-buffer may be changed. Just remember that they use a lot of memory (each line takes 160 bytes of memory, but can be in high memory), so I advice you to use these values:

```
# Number of lines in scroll-buffer
# User Console Monitoring
1000    1000    1000
#
```

Leave this one alone:

```
# Text for forward-header (variables OK), appears LAST in the R:line.
# Please do not change the contents and the order of the fields except
# the FBB$E field which is only an information and can be removed.
# $c is the QTH as declared before in this file.
[$c] FBB$E $$:$R
#
```

Next you must decide how many BIDs to keep.
For WinFBB and LinFBB: Normally you will have enough memory. So you may use the max. However, I found 10000 to be a good value.
For DosFBB: Depending on how much memory you have. 3000 is a good value, but 10000 is better (hi).

```
# How many BIDs are saved (32000 max)
3000
#
```

Leave the next one alone. This one means that bulletins that are older than (in this case) 30 days, will not be forwarded again from my BBS, and will go directly to X-status. Be aware: a too low value here will cause problems for other BBSs that you forward to..

```
# Lifetime for bulletins. Number of days from message is written
30
#
```

You should answer 1 for memory-cache, followed by the list of topics you want to load into memory. The BBS will do these operations very much faster.

```
#
# Use memory-cache ? 0=No, 1=Yes (preferred).
# and optional list of topics to put in memory-cache
#
# Topics : BID = Bulletins or messages identifiers
#          MSG = Message lists
#          HIE = Hierarchical information
#          SCR = Screen buffers
#          WPG = White Pages database
#          REJ = Reject/hold information
#          FWD = Forwarding information
#
2 BID MSG WPG HIE SCR REJ FWD
#
```

List of routes to send WP update messages. First, keep this line blank until you have a WP network available, then give the route(s) to send your updates.

```
# List of routes to send WP messages
#
LA1B LA6CU
#
```

Zip code of the BBS. Give the post code area where is the BBS. Mandatory.

```
# ZipCode of the BBS
#
31120
#
```

Here you will give the parameters allowing the list of messages in unproto mode. This list can be interpreted by some terminal softwares (as TPK) and greatly reduce the traffic on the local frequency.

The first parameter is the maximum backward number. If a unproto user asks a list which is lower than this number backward, he will be limited to this parameter (IE: if the current message number is 20000 and a user asks the list of the message 15000, he will be then limited to the 19500 if the backward number is 500).

The second parameter is the speed of the unproto updates given in seconds. The frames will be sent with this period.

The third parametre(s) have this meaning:

```
A = ACK messages
V = transit private messages
P = end-user private mail
M = subject is replaced by stars in private mail
    (Bulletins are always displayed)
Any unauthorized message will be unproto like
12345 #
```

```
#
500 5 VPA
#
```

Leave this lines alone, unless you are absolutely sure you want to change it:

```
# DRSI and TFPC interrupt (Hexadecimal, default FF and FE)
# FF FE
```

Then you have two optional programs of batches which are called at the end of the initialization, and just before shutdown. If you don't need them, keep these lines empty.

```
#
# BBS-UP program (default empty)

#
# BBS-DW program (default empty)

#
```

You can change these colours if you like:

```
# Colour on texts :
#
```

```

# Black      : 0      Dark grey   : 8
# Blue       : 1      Light blue  : 9
# Green      : 2      Light green  : 10
# Cyan       : 3      Light cyan   : 11
# Red        : 4      Light red    : 12
# Magenta    : 5      Light magenta : 13
# Brown      : 6      Yellow       : 14
# Light grey : 7      White        : 15
#
#
# COULEUR FONDS
#
# Status (top)
1
# Bandeau (middle)
7
# Text (bottom)
0
# Menus
4
#
# Colour of characters
#
# Status (top)
7
# Bandeau (middle)
1
# Menus
14
# Text sent
10
# Text received
12
# UI (Headers)
15
# UI (traffic)
7
# Console
14
# Beacon
6
# Channel-marking
11
#

```

You may change these lines if you like. These are the servers that are normally available in most FBB-BBSs, and they are automatically installed for you from the diskettes. So you can leave them alone. Or you can disable them with a # in front of each of these lines.

Two built-in servers already exist, but you MUST declare them in this file to make them available. As they are built-in, the program-name MUST be replaced with stars.

```

-----
# List of "servers" :
#
# To Program-name Description
#
REQCFG ***** Request configuration
WP          ***** WP server
REQDIR REQDIR Request directory-listings from other BBSs.
REQFIL REQFIL Request files from other BBSs.

```


NEWDOC NEWDOC Upgrade files under DOCS.

```
#
# End of file
#
```

Ok. That was the INIT.SRV file. This file is VERY important, so check and double-check to make sure that you have it all OK. The best way to check it, is by using INSTWFBB for WinFBB, INSTALL.SH for LinFBB or FBBSETUP for DosFBB.

4.13 CRON.SYS:

CRON.SYS (\FBB\SYSTEM).

This file is in the SYSTEM directory. The BBS looks in this file once every hour. If the hour and date is right, the BBS will execute the commands in this file.

DOS-programs are run with X or D command, like X DOTHIS.EXE or D DOTTHAT.EXE. When running BAT files, a COMMAND.COM is automatically first sent. That needs some more memory.

In DosFBB the the screen is saved with D-cpmmand, and reappears after the program has terminated. With X the program will not write to the screen.

```
#
# File CRON.SYS. Here DOS-commands can be executed to fixed
# hours, also Gateway, Yapp, Bip and Talk can be turned on or
# off.
# Format: TALK YES (NO), BIP YES (NO), GATE 1 YES (NO), YAPP 2
# YES (NO).
# DOS-program is run like in FORWARD.SYS with D or X command.
#
# As in FORWARD.SYS, you can use H for hour and D for day.
# 0=sunday, 1=monday,.....6=saturday.
# Also IF-ENDIF and IF-ELSE-ENDIF is allowed
#
# Possible formats for time:
# IF H 0-9 All hours between 00 and 09.
# IF H 0,9 Only at 00 and 09.
# IF H 0,9-11 At 00, and all other hours between 09 and 11.
# IF D 0-2 Sunday, monday tuesday.
#
TALK NO
BIP NO
GATE 1 YES
GATE 2 YES
YAPP 1 YES
YAPP 2 YES
IF D 1-5
    IF H 15-23
        GATE 3 YES
        YAPP 3 YES
    ELSE
        GATE 3 NO
        YAPP 3 NO
    ENDIF
ELSE
    IF H 8-23
        GATE 3 YES
        YAPP 3 YES
```

```

ELSE
    GATE 3 NO
    YAPP 3 NO
ENDIF
ENDIF
X DOTHIS ARG1 ARG2 ARG3
D DOTTHAT ARG1 ARG2

```

4.14 REJECT.SYS:

REJECT.SYS (\FBB\SYSTEM).

This file is in SYSTEM-directory. With this file it is possible to reject or hold certain types or sizes of messages.

The first letter of each valid line specifies the action :

R = Reject : The message will not be received.
H = Hold : The message will be received but held until the sysop reviews.
L = Local Hold : Only messages created on this BBS will be held.

```

# File for rejecting messages. They are rejected with N-BID:
#
# Type, from, @BBS, to, BID, maximum size:
#
# * and ? can be used as wildcards (as in MS-DOS)
#
R B TOTO ALL TATA * 0
R B * * VENTE * 0
R B * VENTE * * 0
H * P1RAT * * * 0
L B * * * * 0

```

4.15 PROTECT.SYS:

PROTECT.SYS (\FBB\SYSTEM).

This file is in the SYSTEM directory.

In this file you can specify some directories in which you do NOT want any sysop to be able to delete any files, without a special syntax. To delete files in protected directories, the command must look like this:
DEL +FORWARD.SYS In an un-protected directory the command looks like this:
DEL FORWARD.SYS. So you can see that you need an extra '+' sign in protected directories.

Example of PROTECT.SYS:

```

\
\FBB
\TOOLS

```

4.16 LANGUE.SYS:

LANGUE.SYS (\FBB\SYSTEM).

This file is in the SYSTEM-directory. You do not need to change this file for the first startups.

First there are 3 numbers. The text in the file explains how to use them.

Next there is a list of all available languages. These names must be the same names as used in the .TXT and .HLP files in the directory called LANG. So, there must be an ENGLISH.TXT and ENGLISH.HLP there, if ENGLISH is in LANGUAGE.SYS file.

After some comment-lines, there is a long list of prefixes, and the corresponding number of language that each user will be presented with for his first connect. The user himself, can change language later with OL-command in the BBS.

Remember, only the "Number in PC at the time" decides how much memory the languages will use. All languages will be available anyway, but if more then (in this case) 4 are in use at the same time, the BBS will swap the less used language to disk, and load a new one from disk.

```
#
# File that decides what language each user will use in the first
# connect. He can change language himself later...
# All prefixes not listed here, will receive language no 1 (english).
# 3 parametres :
# How many languages - How many in PC at the time - Console-language.
#
17 4 1
#
ENGLISH
NORSK
NORSK8
FRANCAIS
SVENSKA
SVENSKA8
DANSK
DANSK8
SUOMI
PORTUGUE
ITALIANO
DEUTSCH
CATALA
ESPAGNOL
NEDERLAN
RUSSE
HRVATSKI
#
# 1 : English
# 2 : Norwegian (7-bits)
# 3 : Norwegian 8-bit
# 4 : French
# 5 : Swedish (7-bits)
# 6 : Swedish 8-bit
# 7 : Danish
# 8 : Danish 8-bit
# 9 : Finnish
# 10: Portugues
# 11: Italian
# 12: German
# 13: Catalan
# 14: Spanish
# 15: Dutch
# 16: Russian
```

```
# 17: Croatian
#
CT* 10
CU* 10
C3* 13
DU* 1
D* 12
EI* 1
E* 14
F* 4
9A* 17
```

etc. etc.

4.17 THEMES.SYS:

THEMES.SYS (\FBB\SYSTEM).

The file THEMES.SYS is configures the theme-zone. The first word in each line is the "theme". The rest of the words on the same line, are the to-fields of messages that will fit in the theme. Empty line (like the TRASH), covers all the remaining to-fields.

TRASH								
COMPUTER	IBM	APPLE	MAC	AMIGA	C64	OS2	MSDOS	TRUCS
SOFTWA UNIX	LINUX							
COMMERCE	ACHAT*	VENTE*	ECH*	RECHER				
SATELLITE	AMSAT	SAT*	UO22	KO23	UOSAT	PACSAT	DOVE	KEP*
SPACE								
RESEAU	FPAC*	FLEX*	RMNC	THENET				
DX	*DX*	VHF	UHF	HF	SHF	QSL*	EME	CONTES
PROPAG DIPLOM	SUN							
IMAGES	JPG	JPEG	GIF	IMAG*	*FAX*	SSTV		
TECHNIQUE	9600	MODEM	TECHNI					
TRX	YAESU	ICOM	KENWO	ALINCO	PYE	KENW		
ATV	ATV	SATTV						
GAMES	JEUX	GAMES						
SCOUTS	SCOUT*							
DIGEST	*DIG							
BBS	FBB	RLI	REBBS	MSYS	TCPIP	CLUS		
TPK	TPK							
ASSOCIATION	REF	URC	ARRL	DARC	RASEC			
EDUCATION	EDUC	ECOLE	SCHOOL					

4.18 REDIST.SYS:

REDIST.SYS (\FBB\SYSTEM).

REDIST is a built-in server for resending personal messages as bulletins. May be used for sending a bulletin in another area than the users own area.

```
#
# Configuration of the REDIST server
#
# Route for the local BBS (SP LOCBBS)
LOCBBS=LA1B
# Name of the BBS coverage
LOCBBS_DESC=LA1B BBS
#
```

```

#Route(s) for local bulletins (SP LOCAL)
LOCAL=LA6CU,LA1B
# Coverage of the local BBS
LOCAL_DESC=Bergen, LA1B and LA6CU BBSs
#
# Route(s) for regional bulletins (SP REGION)
REGION=VEST
# Name of the region
REGION_DESC=Hordaland, Rogaland (Western Norway)
#
# Route(s) for national bulletins (SP NATION)
NATION=NOR
# Name of the country
NATION_DESC=Norway
#
# Default destination
DEFAULT_TO=ALLE
#
# Send a mail to the sysop
MAIL_SYSOP=YES
#

```

4.19 MEMO.SYS:

MEMO.SYS (\FBB\SYSTEM).

This file was designed for DosFBB and works fine there. See comments on use with WinFBB at the end of this file.

This ascii file will substitute a text when pressing Ctrl-F1 to Ctrl-F10. This text can hold variables and is limited to 255 characters. It can include \$W variable (CR) and then will be formatted as more than one line.

This file will have 10 lines, the first line corresponding to Ctrl-F1, and the last one to Ctrl-F10. If a line is not used, it must have a dot in first column.

Example of MEMO.SYS file :

```

$W$O BBS (FBB$E) in $c ($?).$W
.
.
.
.
.
.
.
$W Vy 73. Jean-Paul, F6FBB @ F6FBB.FMLR.FRA.EU$W$W/EX$W
$W 73 Qro. Jean-Paul, F6FBB@F6FBB.FMLR.FRA.EU$W$W/EX$W

```

In WinFBB this does not work the same. It works only in the editor, and only Ctrl-F2, Ctrl-F3, Ctrl-F5 to Ctrl-F9 works. The rest are captured by Windows itself.

4.20 PK-232 HOST MODE:

If you plan to use a PK-232 with this software, you must make some changes. First, check again PORT.SYS. You must have a P in "Type host mode". For example:

```
#
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1 7 1 1 230 4 1 10 30/60 UPYW 433.650
2 1 3 1 80 2 1 5 17/30 GPW 15/20m
#
```

Example of INITTNCx.SYS for PK-232:

```
UNMAIL v LA7QR means Unproto MAIL v LA7QR
RY10 " Retry 10
MN6 " Monitor 6
MC6 " Mcon 6
```

Example of MAINTx.SYS for PK-232:

```
UR1
CTBBS ($c) was shut down for service $d $T.
```

PK-232 host-mode commands (from F6AIW) :

```
8B 8BITCONV AU AAB AB ABAUD AG ACHG AA ACRDISP
AK ACRPACK AT ACRRTTY AE ADDRESS AD ADELAY AI ALFDISP
AP ALFPACK AR ALFRTTY AL ALIST AM AMTOR AC ARQ
AO ARQTMO AS ASCII AY ASPECT AW AWLEN AV AX25L2V2
AX AXDELAY AH AXHANG BA BAUDOT BE BEACON BI BITINV
BK BKONDEL BT BTEXT CL CANLINE CP CANPAC CX CASEDISP
CU CBELL CC CCITT CF CFROM CB CHCALL CD CHDOUBLE
CH CHSWITCH CK CHECK CQ CMDTIME CM CMSG CI CODE
CN COMMAND CE CONMODE CO CONNECT CY CONPERM CG CONSTAMP
CI CPACTIME CR CRADD CT CTEXT CW CWID DS DAYSTAMP
DA DAYTIME DC DCDCONN DL DELETE DF DFROM DI DISCONNE
DW DWAIT EA EAS EC ECHO ES ESCAPE FA FAX
FN FAXNEG FE FEC FL FLOW FR FRACK FS FSPEED
FU FULLDUP GR GRAPHICS HB HBAUD HD HEADERLN HI HID
HO HOST HP HPOLL ID ID IL ILFPACK IO IO
JU JUSTIFY KI KISS LR LEFTRITE LO LOCK MX MAXFRAME
MB MBX MC MCON MD MDIGI MM MEMORY MI MFILTER
MF MFROM MH MHEARD MN MONITOR MO MORSE MP MSPEED
MR MRPT MS MSTAMP MT MTO MA MYALIAS ML MYCALL
MG MYSELCAL MK MYALTCAL NE NEWMODE NO NOMODE NR NUCR
NF NULF NU NULLS OK OK OP OPMODE PA PACKET
PL PACLEN PT PACTIME PR PARITY PS PASS PX PASSALL
PE PERSIST PP PERSIST PC PRCON PF PRFAX PO PROUT
PY PRTYPE RW RAWHDLB RB RBAUD RC RCVE RE RECEIVE
RX RXREV RD REDISPLA RL RELINK RS RESET RP RESPTIME
RT RESTART RY RETRY RF RFEC SE SELFEC SP SENDPAC
SI SIGNAL SL SLOTTIME SQ SQUELCH SR SRXALL ST START
SO STOP TB TBAUD TC TCLEAR TM TIME TR TRACE
TW TRFLOW TI TRIES TD TXDELAY TF TXFLOW TX TXREV
UN UNPROTO UR USERS US USOS VH VHF WI WIDESHFT
WO WORDOUT WR WRU XW XFLOW XM XMIT XO XMITOK
XF XOFF XN XON
```

4.21 G8BPQ driver:

G8BPQ-node (uses TNCs in KISS-mode, DRSI cards, etc...).

If you plan to use BPQ-node with this software, you must make some changes.
Before you start FBB, you must load the BPQ-program.

From version 4.05 of BPQcode, separate ports can be declared. Level 2 connections issue to the MultCh in the same order as PORTs declared in the BPQCFG.TXT (G8BPQ distribution) file. First port corresponds to MultCh 0. Level 4 connections (from the network) always issue to MultCh 0.

In PORT.SYS you must use 2 in INTERFACE, and Q in type host mode. Version 4.05 or up is recommended. COM can be from 1 to 8, but will exclude existing COMs. Choosing 8 is a good solution to keep existing COMs. We recommend that you use COM 8.

Only the INITTNCx.SYS for the 1st port must exist. You must check that there is only one INITTNCx.SYS file for all BPQ-ports, otherwise you will have big problems! The parameters for other ports will be taken by default, and then no conflict should happen between ports. If you want to define the streams differently on each port, you can do it with the INITTNCx.SYS file, but be sure to give the right configuration. INITTNCx.SYS will look like this:

```
A1
R0
UMAIL
M1
N1
```

If paclen is equal to 0, then the default BPQ paclen of the port will be taken, otherwise the specified paclen will overwrite the default BPQ paclen.

If you have validated the gateway, you must configure ENABLE_LINKED=A in BPQCFG.TXT to give to the stream the callsign of the user.

In BPQCFG.TXT you will require to set the TNCPORT section up for COM 8.

As follows:-

```
TNCPORT
    COM=8
ENDPORT
```

Here is an example of PORT.SYS with BPQ-node and 2 TNCs:

```
#
#Ports TNCs
1      2
#
#
#Com Interface Address (Hex) Baud
8      2          0          4800
#
#
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      8    8    0     230   4     1     10    30/60  UQYW VHF
2      1    8    1     80    1     1      5    30/60  UQYW HF
#
```

Example of INITTNCx.SYS for BPQ 4.05 (Only ONE INITTNCx.SYS) :

```
A1      Number of application
R0      Returns to node (0=No,1=Yes)
UMAIL   Unproto MAIL
M1      Monitor (0=No,1=YES)
N1      Number of first stream
```

The parameters for other ports will be taken by default, For instance, as the first stream of the first port is 1, the first stream of the second port will be 9. (8 channels defined in PORT.SYS for port 1). This will be done automatically, and the second INITTNCx.SYS should not exist.

In FORWARD.SYS there are a few changes, here is a short example:

```
A LA2D
#
P B
#
C C SWITCH      Connect first BPQ switch.
C C 2 LA2D      Connect LA2D on port 2 of BPQ.
#
B LA2D
F LA2D
#
-----
```

For WinFBB:

If you are using WinFBB together with BPQ-code you have to copy the files

```
BPQCODE.386
BPQDLL.DLL
```

to the \WINDOWS\SYSTEM directory. These files are normally included in the BPQ programpack.

After this you starts BPQ in the same way as with DOSFBB, but before starting WINDOWS. The best is to start BPQCODE from AUTOEXEC.BAT.

For LinFBB:

BPQ cannot be used with LinFBB....

4.22 Kantronics KAM in host-mode:

If you plan to use a KAM with this software, you must make some changes.

For WinFBB:

Use interface 6 in PORT.SYS.

For DosFBB:

Before you start the BBS, you must load the RS232 driver. You must use ESSKAM or ESS (ESS must be version 1.10 or higher). Do not use another driver such as COMBIOS or MBBIOS.

Run it like this:

```
ESSKAM 1          (Com1, Address and IRQ = defaults)
ESSKAM 1 03F8 4    (Com1, Address=03F8Hex, IRQ4)
```

The KAM must be in the right baudrate, and in hostmode. You put it into hostmode with these commands:

```
INTF HOST
PERM
```

There is a special format for INITTNCx.SYS. For HF, use this:

```
UNPROTO MAIL/
MONITOR ON/
MCOM ON/
MCON ON/
```



```

MRESP ON/
PID ON/
RETRY 15/
RESPTIM 0/
FRACK 2/
CHECK 30/
HID OFF/

```

For VHF use this:

```

UNPROTO /MAIL
MONITOR /ON
MCOM /OFF
MCON /ON
MRESP /ON
PID /ON
RETRY /8
CHECK /30
HID /OFF

```

As you can see, the position of the "/" is very important for HF/VHF !

In PORT.SYS you must use 1 for VHF and 2 for HF in MultCh. You must also use K in "Type host-mode".

```

#
# Same number of lines as number of TNCs.
#
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1    4    1    1    230    4    1    10    30/60  UKYW 144.675
2    1    1    2    80    1    1    6    15/60  UKYW HF
#

```

4.23 DRSI-card:

If you plan to use DRSI-card with this software, you must make some changes. Before you start FBB, you must load the DRSI-driver that comes with the DRSI-card. This must be TNCTSR-R or TNCTSR-L with a version-number higher or equal to 2.1.

The first DRSI-card (MultCh 0 and 1) must be at address 300.
The second DRSI-card (MultCh 2 and 3) must be at address 310.
The third DRSI-card (MultCh 4 and 5) must be at address 308.
The fourth DRSI-card (MultCh 6 and 7) must be at address 318.

Before installing cards in the PC, use the program CHKADDR to verify that the addresses corresponding to the cards have a value of FF. If there is a problem, check your configuration to find what peripheral is already using this address. Install just one card at the time, and verify each cards presence with CHKADDR.

When all cards are ok, configure the driver by means of the TAILORnn program (depending of the version). TNCTSR-S will be ok if you use only a few channels, while TNCTSR-L should be used for big configurations, up to 32 channels.

One problem: What DRSI calls DRSI-PORT, is what we call MultCh in FBB ! What DRSI calls PORT, is what FBB calls port or TNC ! We always use FBB-names here...

Each DRSI-card works like 2 radio-ports. The first DRSI-card is always

MultCh 0 and 1, and the forth card is MultCh 6 and 7. If one radio-port is used on HF, that port must be the second port on the card.

All the DRSI-cards together, work like only one COM. And this COM-port does not need to exist in the PC. You may call that port COM 7 or COM 8 and keep free the already existing COMS.

Baud-rate does not matter, but should be set to a standard value, to avoid error-messages.

Here is an example of PORT.SYS with 2 DRSI-cards (4 radios) :

```
#
#Ports TNCs
1      4
#

#
#Com Interface Address (Hex) Baud
7      4      0      4800
#

#
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      8      7      0      230  4      1      10      30/60  UDYW 433.650
2      1      7      1      80   2      1      5      12/30  GDW  15/20m
3      8      7      2      230  4      1      10      36/60  UDYW 433.650
4      8      7      3      230  4      1      5      10/30  GDW  145.300
#
```

If you are using DRSI-card, you also must change the INITTNC1.SYS. You will need only ONE file for all the ports.

Example of INITTNC1.SYS that initializes 4 ports:

```
MUIS
U0
p0 1 64 10 4 4 10 100 18000 30 2 0
p1 1 64 10 4 1 16 100 18000 30 1 0
p2 1 64 10 4 4 10 100 18000 30 2 0
p3 1 64 10 4 4 10 100 18000 30 2 0
```

The same goes for MAINT1.SYS, only 1 file is needed for all 4 ports, like this:

```
Y 1
U 1 BBS ($c) was shut down for service $d $T.
```

In FORWARD.SYS there is no changes. Use standard syntax.

Well, I think that should be all (for DRSI). Oh, by the way: DRSI cannot limit the number of connections per radio-port. But in FBB we still can have NbCh set to for example 1. When a station then connects to the BBS on that radio-port, and there already is one user connected there, the BBS will send a message to him: "Sorry, no more channels available" and disconnect him.

4.24 TFPCX / TFPCR / TFKISS (Hostmode KISS-driver):

TFPCX / TFPCR / TFKISS (Hostmode KISS-driver).

It is possible to link FBB to a TheNetNode or similar systems running on a separat PC with a simple 0-modem cable.

For this TF-KISS, TFPCR or TFPCX can be used as com-driver, and the link must be defined as a KISSLINK in the nodesystem.

Another possibility is to use the driver with tnc's in kiss-mode etc..
The only restriction is that baycom modem do not work, but SCC cards do work.

It must be called with the same interrupt as stated in INIT.SRV as TFPC-interrupt, or 0xFD by default (if the INIT.SRV does not specify it).
It can be used with DosFBB or WinFBB running under WIN3.xx or WIN95.

IMPORTANT : TFPCX, TFPCR or TFKISS always must be called before starting WINDOWS and/or FBB.

Better add the call into your AUTOEXEC.BAT with a line like this (example using TFKISS) :

```
TFKISS -I:FD -B:9600
```

Comport 1 is default, but other ports as well as other parameters can be specified (look in your manual).

Place the driver itself in your \FBB\BIN directory.

In PORT.SYS you must use 5 in INTERFACE, and D in type host mode.

The interface behaves in many ways as WA8DED hostmode.

If you intend to use higher speed than 9600 baud you must use UART 16550 in your serial ports and PCs not slower than 386 40 MHz.
Otherwise you probably will get errors during binary transfer or forward.

Example of PORT.SYS:

```
# FBB7.00
#
#Ports TNCs
1      1
#
#Com Interface Address (device)  Baud
1      5      3F8      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
0      0      0      0      0      0      0      0      00/01  ----- File-fwd.
1     10      1      1     236      7      4     30     00/15      UDLY VHF/UHF/SHF
#
#TNC Nbs Callsign-SSID Mode
#1      1     LA1B-1      B
#
# End of file.
#
```

Example of INITTNC1.SYS:

```
I OZ7BOX      # Set callsign
M IUS         # Monitor I, U and S-frames - only necessary with UNPROTO
function active
@T2 150       # T2 timer settings
W 10          # Slottime
```

```
P 255          # P-persistence
O 7            # Maxframe
Y 10          # Number of channels
```

Other parameters can be set - look in your manual.
Parameters can also be set with F7 - TNC commands.

If you have outgoing on more than one channel at the time, you have to use SSID's in your forward connects for each channel.
With hostmode SSID's can be set for a channel using the L-command (send tnc-command) together with the I-command (hostmode tnc-command - SET CALLSIGN).

Examples (only part of the connect strings showed) :

Forward to the first call:

```
L I OZ7BOX-10   # Use SSID -10 as your outgoing call
C C ...        # The connect strings
```

Forward to the second call:

```
L I OZ7BOX-9    # Use SSID -9 as your outgoing call
C C ...        # The connect strings
```

etc....

SSID's from -0 to -15 can be used, but NOT SSID's already in use.

5 BBS COMMANDS:

Here is an overview of what the BBS-commands do:

? :

Reading the help file. Type ?[COMMAND] to obtain help on a basic command.

> :

"Break" between channels. Type > [CALLSIGN] [TEXT] to send a text to a connected callsign on another channel. The BBS will advice you that the text has been delivered. This text cannot be longer than one line.
This command is not allowed in read-only mode.

= :

Connection between two channels. Type = [CALLSIGN] to get connected to a callsign on another channel. The callsign can be connected only if it is not busy in a command; the BBS should be idle (waiting for a command). If the callsign is not available, you may either wait or you may cancel your request. The callsign currently busy in forwarding tasks cannot be connected. Use Ctrl-Z to leave this mode.
This command is not allowed in read-only mode.

! : Provide a short and minimal information on the BBS usage.

% : Use this command to see the status of all channels right now.

A : Abort the BBS output at any time.

When reading messages with paging, the commands N and C are available, too.
Use N to skip the reading of the rest of current message, and start on the next message.

Use C to read the rest of the message(s) without paging.

B or Bye :

Disconnects the BBS. Note that the "last listed" parametre is updated.
When the user does a "hard disconnect", the last-listed parametre is
NOT updated.

C : Enters the conference mode, multiconnection.

CW: Shows a list of the callsigns involved in the current conference.

D :

Invokes the FBBDOS, or receives a file from the BBS if the command is followed
by a file name.

F : Switches to Server mode, and access to special commands.

G : Accesses the Gateway mode, if gateway is allowed in the BBS.

H : Shows short help. For help with specific commands, see the ? command.

I :

Information about this BBS. This command displays the file FRANCAIS.INF (or
another language depending upon the configuration and the language attributed
to the involved channel).

I callsign :

Gives the information on callsign taken from the white pages database.
Callsign can include wildcards.

ID : Shows how many records there are in the WP database.

I@ bbs : Gives the users of BBS found in the WP database.

IH route :

Gives the users of the hierarchical route area from WP database
(wildcards allowed).

IZ Zip :

Gives the users of the ZIP code area from WP database (wildcards are
allowed).

J-commands list the callsigns heard by or connected on the BBS.

JK : Shows the last 20 connected callsigns.

JA : Connected callsigns on port A.

JB : Connected callsigns on port B, etc...

J1 : Heard callsigns on port A.

J2 : Heard callsigns on port B, etc...

K-commands delete messages sent by you or to you.

K msg# : Deletes a message identified by its number.

KM : Deletes all the messages addressed to you, that you don't yet
read. The messages not read will not be deleted.

L : Lists the new messages since the last usage of this command.

LA : List messages with status A.

LB : Lists bulletins.

LC [mask] :

List the messages which "TO" filed matches with mask. * (star) allows to
list all the messages and should be the default value. Type "LC *" to see
all messages. LC alone gives the current mask. Mask is recorded for each
user while disconnecting.

LD> YYMMDD : Lists the messages received after the specified date.

LD< YYMMDD : Lists the messages received before the specified date.

LF : List FORWARDED messages.

LK : List KILLED messages.

LM : Lists the messages TO YOU.

LN : Lists the NEW messages TO YOU.

LP : List PRIVATE messages.

LU : Lists all unread messages to/from the user.

LX : List messages with status X.

LL 10: Lists the last 10 messages.

LR : Equivalent to L but the messages are listed in reverse order.

LS text :
Searches for a text in the messages titles. Usage of wildcards is allowed.

L< Callsign :
Lists the message originated from [Callsign]. The wildcards are allowed.

L> Callsign :
Lists the messages destined to [Callsign]. Wildcards are allowed.

L@ [BBS] :
Lists the messages VIA BBS if specified. L@ alone lists the message without route. The wildcards are allowed.

L msg#- : Lists all messages, starting from msg#

L msg#-msg# : Lists messages between the given numbers.

M msg# File :
Transfers the text of the message msg# to the file File. This file will be placed in the directory USERS or from the root in case of a SYSOP user.

MH msg# File : Identical, but the format is the R command.

MV msg# File : Identical, but the format is the V command.

MA msg# File :
Transfers the text of the message msg# to the file File. This file will be placed in the directory USERS or from the root in case of a SYSOP user. The message is appended to the existing file.

MHA msg# File :
Identical, but the format is the R command. The message is appended to the file.

MVA msg# File :
Identical, but the format is the V command. The message is appended to the file.

N : Enters or modifies the name (max 12 characters allowed).

NH :

Enters or modifies the Home BBS. A point deletes the current Home BBS.

NL : Enters or modifies the QRA locator.

NP : Modifies the password for the access via telephone modem.

NQ : Enters or modifies the city (Qth).

NZ : Enters or modifies the Zip Code.

O : Displays the setting of various options.

OL : Lists the available languages.

OL number : Selects a language.

ON : Shows the number-base for messages. See also ON number below.

ON number :

Defines the number of the thousand base for the messages. Example: ON 37 defines number 37000 as being the base for numbering the messages. Then the command R 25 will cause the message 37025 to be read, although R 36025 will actually read the message nr 36025.

OP : Validates or disables the scrolling of pages.

OP nbr of lines : Initializes the page scrolling

OR :

Validates or disables the ability to list all messages, if the user has access to the whole list of messages by default (See security code 256 of INIT.SRV).

PG [program]:

PG alone gives the list of PG programs available in the BBS. If followed by a program name gives the control to this program. The PG program can also be called by its name. If the name is the same that an existing command, the existing command will be replaced with the PG program.

PS : Gives the list of servers available in the BBS.

R-commands are for reading messages. Note:

When reading messages with paging, the commands N and C are available, too. Use N to skip the reading of the rest of current message, and start on the next message.

Use C to read the rest of the message(s) without paging.

R msg# msg# : Reads the messages by their numbers.

RM : Reads all the messages to you.

RN : Reads all the NEW messages to you.

RU : Read all unread messages to/from you.

R> callsign : Read all messages to a callsign.

R< callsign : Read all messages from a callsign.

S[type] callsign :

Sends a message or a bulletin. If the type is not specified, it will be processed as a private, provided the destination callsign is a valid one, otherwise it will be considered as a bulletin.

SP callsign : Sends a private message.

S callsign @ BBSCALLSIGN :

Sends a message to a station AT another BBS through forward. BBSCALLSIGN can be an address of hierarchical type, for example F6FBB.FRA.EU. Type "SB ALL" to send a bulletin addressed to ALL. The message must be terminated by a Ctl Z or /EX followed by a return. Warning: the /EX must be in the first column.

SC Msg# Callsign @ BBSCALLSIGN :

Copies a message or bulletin to the specified callsign. Such a copy will always be of the "personal" type.

SR [Msg# [title]] :

Replies to a message already read or formerly suppressed, or to a message matching the specified number. If the title is not stated, it will be the title of the read message, with the mention "Re:" placed ahead.

T :

Pages the SysOp. Should the SysOp not be available within a minute, you will be advised accordingly and automatically returned to the normal BBS menu.

TH : Gives access to the themes-zone. Lists messages by themes, as setup in the file THEMES.SYS. Available commands in this zone, are

H (list themes)

L (list messages in current theme)

R msg# (read message in current theme)

number (change to theme#)

U :

Uploads a file from the user to the server of the BBS. The file will be written in the root directory of the FBBDOS. The U command is to be typed at the BBS prompt, it is not necessary to go to FBBDOS first.

V :

Version number of this BBS software. It gives also the number of active messages, and the next message number.

V msg# msg# :

Reads the messages by their numbers, including the headers.

VM : Reads all the messages TO you, including the headers.

VN : Reads all the NEW messages TO you, including the headers.

W :

Lists the files which are within the user directory. This command can be followed by a mask specifying search criteria or a directory name; in the latter case, the directory contents will be listed.

X :

Toggles "normal" or "expert" status. "Normal" grants the extended menus, "expert" provides only minimal information.

Y :

Binary transfer program using the YAPP protocol. As a user, you should also have the YAPP protocol available your end, in order to be able to transfer binary data.

YD File : Downloads a file FROM the BBS TO you.

YI [mask] : Lists the available binary files, and their description.

YN [mask] : Lists the NEW binary files since your last connection.

YU File :

Uploads a binary file from you TO the BBS. You cannot replace or modify an already existing file.

YW [mask] : Lists the available binary files.

YZ Filename : Deletes the specified filename in the YAPP directory.

Z Filename : Deletes the specified filename in the user directory.

6 SERVER -MODE:

The server commands are gathered in groups.

C : Connections - Statistics.

D : Documentation.

Q : QTH locator.

N : Ham Callsigns Directory ("Call-Book").

T : Satellite orbits computation.

F : Switches back to BBS mode.

B : Quit and disconnects

6.1 CONNECTIONS-statistics.

Statistical data about the operation of the Server and the Traffic Log.

O : Activity of the different groups. Percentage of usage for the different activities.

G : Global statistics.

- Number of active messages.

- Number of connections.

- Time of connections.

- Peak traffic periods.

L : Lists the connections (Traffic Log). For each and every connection, the channel - callsign - date and time - as well as the connection duration are indicated. These informations are listed in reverse chronological order.

H : Hourly usage.

J : Daily usage.

I : Users Callsign list.

List of the callsigns that have connected the server. This list is given in alphanumeric order.

F : Switches back to the Server main menu.

B : Quit and Disconnect.

6.2 DOCUMENTATION-server.

You will find under these comments the information or documentations given by various hams.

The documentation is organized like the image of a tree structure whose root is defined in INIT.SRV (C:\DOCS). The directories represent the topics and

the files represent the documentation.

The selection of the topics or the documentation is made through numeral command. The menu is presented with the form of one or several numbers followed with their description (topics or documentations).

In the case of topics, the description corresponds to the name of the MsDos directory. This name is eleven characters long, the dot between the name and the extension will not appear. It is possible also to simulate the space character with an underscore character.

In the case of documentation files, a label of 30 characters is associated to the name of the MsDos file. The label only will be presented to the user. To optimize the fill up of the screen, each line will contain two labels.

In the case of a user, given the right of SYSOP, the display will be different. One line will contain only the name of the MsDos file followed by its label.

L: L gives the list of the documentation on the current chapter.

D: Labelling of the files is made by the command D, reserved to SYSOP, or automated through the NEWDOC server (See paragraph SERVERS, NEWDOC).

Example : D BBS.DOC Documentation of the BBS.

F : Switches back to the Server main menu.

B : Quit and Disconnect.

6.3 QTH-locator server:

The QTH locator is made of 2 letters, 2 figures, and 2 letters yielding the geographical coordinates of a station. You can translate the QTH locator into Longitude/Latitude or vice-versa, compute the distance and azimuth between two QTH locator or compute your Log-Sheets for contests (adding together distances).

Q : Converts one QTH Locator into Longitude and Latitude. The West longitude and North latitude are given in degrees minutes, or grades.

D : Distance and Azimuth computation between two QTH Locators.
Computation of the distance and azimuth between two QTH Locators. The distance is given in Kilometers and the azimuth in degrees.

C : Computation of the distance and azimuth between two QTH Locators.
The distance is given in Kilometers and the azimuth in degrees.

L : Converts the Longitude and Latitude into QTH Locator. The west longitude and north latitude can be given in degrees, degrees minutes, or grades.

F : Switches back to the Server main menu.

B : Quit and Disconnect.

6.4 Callsigns-directory.

You will find in this module the information that the users will have accepted to leave there: name, first name, address, phone number, etc... You will be able, as well, to update your own various information.

N : Modify your name and address.
 R : Search for information about a user.
 I : List the connected callsigns.
 N : Modify your name and address. If the information in the file is erroneous, or should they have changed, you can modify them with this command. A space or a return as an answer to the question will not modify the content. If the content of the file is correct, you can skip modification by replying N to the modification request.
 R : Search for information about a user. If the user is known in the file, you will get information about him. You can concatenate the callsign of the user who is searched for behind the command.
 Ex.: R F6FBB will give information about F6FBB.
 I : List the callsigns of the users who are connected the server. This list is sorted in an alphanumeric order.

 F : Switches back to the Server main menu.
 B : Quit and Disconnect.

6.5 Satellite orbits server:

Satellite orbital parameters computation, and satellite characteristics.

P : Display of orbital parameters. Selected satellite orbital parameters display. These parameters will be used for the computation.
 C : Amateur Satellites characteristics.
 F : Switches back to Server main menu.
 T : Satellite orbital computation.
 M : Manual editing of orbital elements.

To select a satellite, type the number (a list of the satellites can be obtained with the command L instead of the number). Once the satellite has been selected, you will have to supply the date and the time to be taken into account for the computation. If you only answer with a return character, the current date (or time) will be taken. The geographical position is the QTH Locator of the connected user, if it is known in the file. Otherwise, the QTH Locator of the server will be used as a default. Depending on the satellites, the computation step can be variable, and one line of parameters will be displayed per step. This line corresponds to a minimal elevation of -5 degree s, that means the satellite 5 degrees below the horizon.

F : Switches back to the Server main menu.
 B : Quit and Disconnect.

7 FBBDOS COMMANDS:

The syntax of the various FBBDOS commands is very close to that of MsDos(R).

Several specific commands have been added: PUT and GET for ASCII file transfers. XPUT, YPUT and XGET, YGET for binary file transfers.

EDIT command for ASCII files editing.

When entering FBBDOS, the prompt shows the virtual directory in which the user is located. There is no physical drive concept (A, C, etc...), the user is situated in a directory defined in the INIT.SRV file; this directory becomes the root of the available directory tree. Should the user be a SYSOP,

and if he successfully validated his privileges using the SYS command, the root of FBBDOS will be the root of the current drive.

A file created by a user belongs to this user.
Only the owner of the file, or a SYSOP granted writing privileges (SYS command) are permitted to delete or modify this file.

BGET file :

A binary file is downloaded from the BBS. The Autobin protocol is used.

BPUT file :

A binary file is uploaded to the BBS. The Autobin protocol is used.

CD directory :

Changes the active directory. This command allows you to move within the directory tree by changing the active directory. "CD.." changes back by one directory "upstream" towards the root.

COPY Origin Destination :

Copies a file into another file. If the destination file already exists, it must belong to the user.

DEL file :

Deletes a file. A file cannot be deleted if it does not belong to the user.

DIR [mask] :

Displays all the files contained in the directory or these corresponding to the mask if it is specified. If the mask matches a directory name, the content of this directory will be listed.

DOS Program Arguments :

Allows a remote SYSOP (after SYS command) to run a DOS program. This program can not expect keyboard input. A list of arguments can be specified. The output to the screen will be send to the remote SYSOP at the completion of the program. If the program expects inputs, after a time-out of 2'45, it will be aborted (if possible) and the normal operation will resume.

DU [mask] : Gives the size of the tree, and local used disk.

EDIT file :

Edition of a text file, using the line editor. If the file already exists, it must belong to the user. Instructions on how to use the line editor is in a separat section of this documentation.

EXIT : Exit FBBDOS and return to the BBS or server.

GET file :

Displays the content of an ASCII file. Such a display is never scrolled by pages, and ends with a Ctrl-Z, thus allowing ASCII file transfers from the BBS to the user.

LIST mask :

Displays all the files contained in the directory or these matching the mask specified. if the mask matches a directory name, the content of this directory will be listed. Each line displays the filename and a descriptive comment (cf YI command).

MD directory :

Creates a new directory. The new directory is appended next to the current directory.

O : Display various options.

OL : Lists the available languages.

OL number : To select a language.

OP : Validates or disables the scrolling of pages.

OP nbr of lines : Set how many lines per page in page scrolling.

PRIV :

Access to a private directory (or return to Fbb-Dos directories). Each user can have access to a private directory provided it has been defined by the SYSOP (I.E. EU command). Several users may share a private directory.

PUT file :

Accepts an ASCII file from a user. The file must end with a Ctrl-Z to be validated. If the file already exists, it should belong to the user.

RD directory :

Deletes a directory. A directory cannot be deleted as long as it is not empty.

TYPE file :

Displays the content of an ASCII file. Eventually, this list can be scrolled by pages.

VIEW mask :

Displays the content of all the archive files contained in the directory and corresponding to the specified mask.

For DosFBB and WinFBB: This function uses the FV.COM shareware.

Thanks to the authors of this last one.

In LinFBB: VIEW command of LINUX version calls FBB_VIEW program. (or any program defined by the variable \$FBB_VIEW). fbb_view may be a shell script.

WHERE filename :

Command to find one or more files in FBBDos. Wildcards are allowed.

XGET file [offset]:

Only via telephone-modem:

A binary file is downloaded from the BBS. The Xmodem protocol is used. This command

works only on a PSTN modem port. If offset is specified, download will start from offset.

XlGET file : Same as above, with 1K-Xmodem protocol.

XPUT file :

Only via telephone-modem:

A binary file is uploaded to the BBS. The Xmodem protocol is used. This command works only on a PSTN modem port.

YGET file :

A binary file is downloaded from BBS.

Via radio:

Both YAPP and YAPC protocols may be used.

Via telephone-modem:

YGET starts the YMODEM, YMODEM-batch or YMODEM-G protocols.

YPUT file :

A binary file is uploaded to the BBS.

Via radio:

Both YAPP and YAPC protocols may be used.

Via telephone-modem:

YPUT starts the YMODEM protocole. This is NOT available if the BBS is running DosFBB !

ZGET file :

Only via telephone-modem:

ZGET starts download from the BBS using the YMODEM, YMODEM-batch or YMODEM-G protocols.

ZPUT file :

Only via telephone-modem:

YPUT starts upload to the BBS using the YMODEM protocole. This is NOT available if the BBS is running DosFBB !

8 SYSOP-COMMANDS:

Commands available under the BBS menu.

DU callsign : Displays the information related to the given callsign.

DB : Lists the BBS callsigns.

DE : Lists the EXCLUDED callsigns.

DL : Lists the callsigns with LOCAL status.

DM : Lists the callsigns with modem acces.

DP : Lists callsigns which are using the page scroll.

DS : Lists the SYSOPS callsigns.

DX : Lists the EXPERT callsigns.

E msg# :

Edits the message # header. The message header is first displayed, then a string of options will allow you to edit the various fields of this header. A "return" terminates the edit.

EM msg# :

Edits the body of a message. During edit the message is not being forwarded.

EU callsign :

Edits the main parameters of a user. The user's parameters are first displayed, then a string of options will allow you to edit the various parameters. A "return" terminates the edit. Most of fields can be cleared with a point. A user can be edited while connected and then action is immediate. Flags are :

R : All mail (included personal) can be read

P : Paging validated

B : BBS

S : Sysop

X : Expert

L : Local (Allowed to all commands on guest access).

E : Excluded (cannot connect to the BBS)

M : Modem access

U : Unproto (Allows unproto list requests).

I : Give list of new messages on logon.

F : PMS forwarding (allows forward but texts are like normal user).

Other commands must be followed with information :

- G number : selected language
- N name : name of the user
- W password : password of the user
- V directory: private directory of the user (12 characters max)
- H home-bbs : home-bbs of the user
- Z zip-code : zip-code of the user

FA msg# BBS :

Specifies that the message defined by its number will be forwarded to the specified adjacent BBS.

FB BBS : Gives the list of messages waiting for forward to this BBS.

FC Route :

Gives the list of BBSs that will receive the messages with the specified route. If the route is a possible callsign, the BBS will look for BBS routing, if composed with digits will look for NTS routing, otherwise will look for bulletins (group) routing.

FD msg# BBS :

Removes the message # from the forward to the specified adjacent BBS.

FG : Shows the partly received messages (may be resumed in next forward).

FH BBS : Gives the hierarchical extension for BBS.

FI BID command:

BID maintenance.

- Command = ? : Gives message number and the message type of the BID/MID.
- + : Adds the BID/MID to the list.
- : Deletes the BID/MID from the list.

FL [BBS] : Lists the messages in the forward-queue [for BBS if specified].

FN Msg# :

List of the BBSs which are in the waiting list for this message number, and the BBSs already forwarded or named in the preamble.

FP command callsign:

Swap test.

- Command = @ : Gives the swap (if existing) for the via callsign.
- > : Gives the swap (if existing) for the to callsign.
- < : Gives the swap (if existing) for the from callsign.

FR Argument :

Forces reverse forwarding to BBS (or port), even if there is no pending mail. If the argument is a callsign of known BBS, reverse forwarding to this BBS will start. The argument can be a portnumber, in this case the reverse forwarding will start on that port, if 9 is specified then the reverse forwarding will start for all the ports.

FS Argument :

Stops the forwarding. If the argument is a callsign of known BBS, the forwarding to this BBS will be stopped. The argument can be a portnumber, in this case the forwarding will be stopped on that port, if 9 is specified then the forwarding will be stopped for all the ports.

FT [BBS] :

Lists the volume of the messages which are waiting for forward to all BBSs, or to only one BBS if the callsign of that BBS is specified.

The command shows the volumes in real-time.

FU Channel :

Immediately disconnects the specified channel.

FV :

Recompiles FORWARD.SYS and its include files, SWAPP.SYS and BBS.SYS and then starts scanning the messages to update the forwarding list. This may be necessary after a FORWARD file change.

FW Argument :

Starts forwarding to BBS (or port). If the argument is a callsign of known BBS, forwarding to this BBS will start. The argument can be a portnumber, in this case the forwarding will start on that port, if 9 is specified then the forwarding will start for all the ports.

HO msg# : Puts a message in the HOLD-list.

IE Callsign :

Gives first the WP information using IL format (see below), and then prompts the deletion of the record. If the answer is Y, the record will be deleted, else you enter the edition mode. Type return to exit edition mode. The commands are :

U : Updates primary part with the temporary part (2 -> 1).

N name : Changes the name of the user.

1H Route : Changes the primary route.

1Q Qth : Changes the primary Qth.

1Z Zipcode : Changes the primary ZipCode.

2H Route : Changes the temporary route.

2Q Qth : Changes the temporary Qth.

2Z Zipcode : Changes the temporary ZipCode.

IL Callsign :

Gives all the information of a callsign from the white pages database.

Kx :

The fact that a leading K is entered before a kill command with the letter K (I.E. KK, KK<, ...) triggers immediate killing of the specified message. The archiving will be done at the next epurmess process according to epurmess parameters. The status of the message becomes "A". The syntax is identical to the command "K".

K> callsign : Deletion of any message to this callsign.

K< callsign : Deletion of any message sent from this callsign.

K@ route : Suppresses all the messages which have the specified route.

KF : Suppresses all the messages already forwarded.

\$ msg# :

Information about msg# forwarding: lists the BBS which have not yet received the message #.

LH : List HELD messages.

LU (RU) :

Lists (or displays) all the personal messages which have not yet been read by their addressee.

Mx :

Same as the command "L", with the same syntax, but the BID is inserted at the beginning of the "SUBJECT"-field. This command allows you to do some

search or to visualize BIDs in the message lists (commands MS, ML, etc...).

PR :

Enables or disables the printer from the console. The printer is automatically disabled on disconnect.

PR> filename : validates and redirects the printing to a filename.
(Works only in DosFBB)

PR> - : validates and redirects the printing to the standard printer.
(Works only in DosFBB)

RE : Releases HELD messages.

SR [#msg] [+filename] [subject]

Reply to a message already read or formerly suppressed, or to a message matching the specified number and appends the contents of filename. If the subject is not stated, it will be the title of the read message, with the "Re:" placed ahead.

SYS :

Enables remote Sysop mode. A callsign validated as SYSOP is permitted to access messages and files in the BBS according to the settings in PASSWD.SYS file. The SYS command will return a string of numbers corresponding to the position of a letter in a sentence located in the PASSWD.SYS file. If you send back the 5 letters (without any space) matching the 5 numbers, an OK will be sent out to confirm that the SYSOP mode is granted.

YL file : Modifies the description of an existing YAPP file.

Other commands:

M :

In the SATELLITE ORBITAL COMPUTATION menu, manual modification of the orbital parameters of the satellites.

N Callsign :

In the "Call-book" menu, this command allows to edit/modify call-book information of a user.

S file :

In the gateway and in command mode, send an ascii file.

W file :

In the gateway, opens a capture file. The file is closed with another W command.

YU Filename :

In gateway mode and only from the console send a binary file to a remote system under the YAPP protocol.

YD Filename :

In gateway mode and only from the console receive a binary file from a remote system under the YAPP protocol.

/K : Start housekeeping.

/L : Software reboot.

/M : Immediate software reboot.

You will also find a list of all function-keys in the Console Commands chapter.

9 CONSOLE COMMANDS:

(Function-keys and other).

There are a few differences in WinFBB from DosFBB.

For WinFBB:

Most of these commands will open a window. The server operation is NOT suspended while these windows stay open.

- F1 - Help.
- F2 - Connect to the BBS locally with the console-callsign.
- F3 - Disconnect a channel.
- F4 - Start or stop forwarding. Shows how many/big messages are waiting for forward right now (real-time). Click on the BBS(s) you want to start forward to, and click "Start Fwd". There are also buttons for "Start All" and "StopFwd", as well as import and export.
- F5 - Toggle monitoring on/off.
- F6 - Change console-callsign. That callsign is used when connecting to the BBS with F2 or via gateway F9. The callsign have sysop-status.
- F7 - Program TNC directly. This allows a dynamic modification of the parameters which will be affected as default to the channels of the related port. These parameters will not be taken into account for the channels already connected, and will be transferred to the channel upon disconnect. The commands are specific to the TNC in use: WA8DED with a TNC2, HostMode PK232 with a PK232, or the BPQ node.
- F8 - Not used
- F9 - Gateway, using console-callsign.
- F10 - Initiate a chat with a user (click on the user first).

- ALT-F1 - Not used
- ALT-F2 - Not used
- ALT-F3 - Start or stop forward-scan. Before starting scanning, FORWARD.SYS and its include-files, BBS.SYS and SWAPP.SYS are recompiled.
- ALT-F4 - Quit FBB.
- ALT-F5 - Start a text-editor.
- ALT-F6 - Not used
- ALT-F7 - Show last connections.
- ALT-F8 - Not used
- ALT-F9 - Not used
- ALT-F10 - Not used

- ALT-B - Not used
- ALT-D - Not used
- ALT-R - Not used
- ALT-N - Not used
- ALT-S - Not used
- ALT-X - Not used

CTRL-F1 to CTRL-F10 hold pre-defined texts, that are sent when pressing that key. The texts are in the file MEMO.SYS. Works only in the editor.

Exceptions:

CTRL-F1 and CTRL-10 are captured by Windows.

CTRL-F4 exits the editor.

In addition to the function-keys, there are quite a few push-buttons on the screen in WinFBB, in standard Windows-fashion. Click on them to get

the wanted result. This should be self-explanatory. Much information may be retrieved this way, and many things may be configured this way.

In addition to the main FBB-window, you may want to have open at the same time the DISPLAY ALL CHANNELS and MONITOR windows (I do, hi). They may be re-sized to fit beside and under the main FBB-window. The windows position and size and saved if you stop the BBS.

For LinFBB:

No function-keys are available in LinFBB.

For DosFBB:

Most of these commands will open a window. The server operation is suspended while these windows stay open, but the user requests are queued and will not be lost. If no reply is given, the window will close down after a few seconds, and the default reply will not be taken into account. Do not forget to validate your replies with "Enter".

- F1 - General help on all function-keys.
- F2 - Connect to the BBS locally with the console-callsign.
- F3 - Disconnect a channel.
- F4 - Start or stop forwarding.
Manual forwarding request, or cancellation of current forwarding. If a number up to 8 is specified, the corresponding port will be scanned for forwarding. If the number 9 is specified, all ports will be scanned. If a callsign is specified, it will be scanned for forwarding. Preceding the callsign with + will force reverse even if there is no waiting mail, preceding the callsign with - will stop the forwarding and disconnect.
- F5 - Toggle monitoring on/off. With monitor on, the lower part of the screen displays the monitored traffic, while the upper part of the screen shows traffic in the BBS. Number of lines in upper/lower part of the screen, can be adjusted with the arrow-up or arrow-down keys.
- F6 - Change console-callsign. That callsign is used when connecting to the BBS with F2 or via gateway F9. The callsign have sysop-status.
- F7 - Program TNC directly. This allows a dynamic modification of the parameters which will be affected as default to the channels of the related port. These parameters will not be taken into account for the channels already connected, and will be transferred to the channel upon disconnect. The commands are specific to the TNC in use: WA8DED with a TNC2, HostMode PK232 with a PK232, or the BPQ node.
- F8 - Reboot PC. Can be done at once, or when all channels are free.
- F9 - Gateway, using console-callsign.
- F10 - Talk to a user (that have sent a T) or initiate a chat with a user.
- ALT-F1 - Shows pending forward(s). Shows number of personal messages, bulletins, and the total amount of kB waiting to be forwarded.
- ALT-F2 - Shows all callsigns connected to the BBS.
- ALT-F3 - Start or stop forward-scan. Before starting scanning, FORWARD.SYS and its include-files, BBS.SYS and SWAPP.SYS are recompiled.
- ALT-F4 - Shows the last connected callsigns.
- ALT-F5 - Starts a very simple text-editor (use F1 for help there).
- ALT-F6 - Various "options" :
 - Sysop is in ? Enables T-command.
 - Beep at connect ?
 - Show callsigns etc. on every channel ?
 - Gateway allowed ?
 - Justification on text typed from console ?
 - Use screen editor ?

ALT-F7 - Import messages from file.
ALT-F8 - Export messages to file. Messages that are waiting to be forwarded to one BBS. These messages can be forwarded to a file.
ALT-F9 - Exit to DOS. WARNING: the BBS operation is suspended all this time long. The users will not be disconnected, and their commands will be queued. Return to BBS by typing EXIT.
ALT-F10- Stop the mailbox-program. Can be done at once, or when all channels are clear.

ALT-B - Blank screen.
ALT-D - Save scroll-buffer to file.
ALT-N - Allows reprogramming a telephone modem
ALT-R - Redisplay screen.
ALT-S - Show last DOS-screen. Press a key to return to BBS.
ALT-X - Stop the BBS (same as ALT-F10).

CTRL-F1 to CTRL-F10 holds pre-defined texts, that are sent when pressing that key. The texts are in the file MEMO.SYS.

Arrow -> - Display next channel.
Arrow <- - Display previous channel.
Ctrl -> or Ctrl <- Return to the display of all channels.
PG-UP Display the previous page if it exists
PG-DW Display next page.
HOME Display the first page.
END Display the last page.

10 FULL SCREEN EDITOR (DosFBB):

(Only for DosFBB, WinFBB has another editor)

The editor is validated for answering messages, with Alt-F6.

The Tiny-Edit is started with the Alt-F5 command. Normal operation of the BBS is maintained, the editing takes place in the console window, above the monitoring area of the display.

Most of the text files can be edited with this editor despite of its simplicity. The most concerning drawback is the fact that the line length can not be longer than 79 characters. The language files (xxxx.TXT) therefore cannot be edited.

Tabs are not generated and are presented on the display with a special character.

During the editing, the usual BBS command set are no longer available, they are replaced by the tiny-edit commands.

Upon activation of the editor, a dialog window ask for the filename to edit.

The default directory is the directory "SYSTEM" as defined in INIT.SRV.

Examples: FORWARD.SYS will edit SYSTEM\FORWARD.SYS
FWD\F6FBB will edit SYSTEM\FWD\F6FBB

It will be necessary to specify the full path for a file outside of the default directory root or located on another disk drive. If only the drive name is specified, the current directory on this drive is searched for the filename.

Example: C:\INIT.SRV will edit C:\FBB\INIT.SRV (while the current directory

is c:\FBB).

To edit a message, one should give the message number with a leading pound sign ('#').

Example : #42645 will edit the message number 42645.

In case the edited file is a compiled one (I.E. FORWARD.SYS) it will be compiled again after the termination of tiny-edit

Function-keys :

F1 : Help windows.
F2 : The edited file is appended to the incoming mail file of the BBS (MAIL.IN).
F3 : Insert file.
F8 : Save current file.
F9 : Save current file and exit.
F10 : Exit Tiny-Edit without saving ! Caution : No confirmation requested before closure of the current edit session.

Edit Commands :

Almost all commands are available from the numeric key-pad.

Up-arrow : The cursor moves up one line
Down-arrow : The cursor moves to next line
Right-arrow : The cursor moves forward one character. At the end of the line it moves to the beginning of the next line.
Left-arrow : The cursor moves backward one character position. At the beginning of the line it moves back to the end of the former line.
Page-down : The cursor moves toward the end of the current page. If it is already at the end of page position, then the next page will be displayed.
Page-up : The cursor goes back to the first line of the current page. If it is already at the first line, the former page will be displayed.
Home : The cursor moves to the first character of the current line.
End : The cursor moves to the character position next after the last character of the line.
Insert : Toggle the insert editing mode. The cursor is an underscore while in insert mode. It is a cursor block in overwrite mode.
Delete : Erase the character at cursor position, the cursor is not moved.
Back-space : The character preceding is deleted and the cursor moves backwards one position.
Ctl right-arrow : The cursor moves to the first character of the next word.
Ctl left-arrow : The cursor moves to the first character of the previous word.
Ctl home : The first page of the file is displayed and the cursor moves to the first line.
Ctl end : The last page is displayed and the cursor goes to the last line of the file.
Ctl Y : The current line is deleted.
Ctl Q : The current line is deleted from the cursor position up to the end.

11 VARIABLES:

Variables definition:

Most of the text files support usage of variables. These variables enable text or values to be placed, depending upon the context or the channels in use.

These variables are defined with the \$ sign, followed by a letter. The meaning will be different if the letter is in capital or in lower case. A second set of variables is defined with the % sign, followed by a letter.

They can also be followed by a figure and in this case, this is a parameter whose value is function of the context. These particular variables are used in texts, usually as computation results. Do not use them in other places than text files (*.TXT).

The \$ sign will have to be written \$\$, if it is not to be considered as a variable. The % sign will have to be written %%.

List of the variables:

- \$0 - \$9 : Temporary variables.
- \$A : @ BBS of the message header.
- \$a : Year of the current date.
- \$B : Inserts a Ctrl-G (Bell).
- \$b : Zip code of the user.
- \$C : Number of the next message.
- \$c : City where the BBS is located.
- \$D : Current date (Format : YYMMDD).
- \$d : Current date (Format : DD-Dec).
- \$E : Version number.
- \$e : City of the connected user.
- \$F : Minute of the hour for forwarding.
- \$f : File appended to the message header.
- \$G : Destination of the message header.
- \$g : Number of gateway ports available.
- \$H : Current time (Format : HH:MM).
- \$h : Home BBS of the user.
- \$I : First name of the connected user.
- \$i : Date and time of the message header (Format MMDD/HHMM).
- \$J : Date of the message header (Format YYMMDD).
- \$j : Date of the message header (Format 29-Dec).
- \$K : Time of the message header (Format : HH:MM).
- \$k : List of the users involved in the conference.
- \$L : Number of the last used message.
- \$l : Filter value of LC command.
- \$M : Number of the message header.
- \$m : Frequency of the port in use.
- \$N : Number of active messages.
- \$n : Size of the message header.
- \$O : BBS callsign.
- \$o : SSID of the BBS.
- \$P : Origin of the message header.
- \$p : Number of lines of scroll page option, for the connected user
- \$Q : List of the 8 callsigns which have new messages.
- \$q : Base messages number of the user.
- \$R : BID of the message header.
- \$r : L or space character. L means that the message was locally created.
- \$S : Subject of the message header.
- \$s : Status of the message header.
- \$T : Time of message header (Format : HH:MM).
- \$t : Type of message header.
- \$U : Callsign of the connected user.
- \$u : Current drive in FBBDOS.
- \$V : SysOp first name.

\$v : Complete route (with the hierarchical structure).
 \$W : Inserts a carriage return.
 \$w : Inserts an escape character.
 \$X : Date of the last connection (Format : 881229).
 \$x : Date of the last connection (Format : 29-Dec).
 \$Y : Time of the last connection (Format : 12:30).
 \$y : Year of message.
 \$Z : Last message read by the user (L command).
 \$z : Zip Code of the BBS.

\$* : Number of active channels.
 \$= : User's current channel.
 \$! : User's current port.
 \$^ : Number of allowed ports.
 \$? : BBS QTH-Locator.
 \$% : List of the connected users.
 \$: : Connection duration (Format : 12mn 05s).
 \$. : Machine operation time (Format : 12mn 05s).
 \$\$: Character \$

Second set of variables :
 (unformatted means that text is not padded with spaces) :

%A : @ BBS of the message header, unformatted.
 %C : Gives the number of times a message has been read.
 %d : Nb of KB downloaded to the user.
 %E : Give the version code (Linux, Dos, Windows).
 %G : Destination of the message header, unformatted.
 %i : Date and time of the message creation (Format MMDD/HHMM).
 %J : Date of the message creation (format YYMMDD).
 %j : Date of the message creation (format 29-Dec).
 %K : Time of the message creation (format HH:MM).
 %M : Number of the message, unformatted.
 %m : Maximum KB of download allowed on the port.
 %N : Gives the message number limited to the right 5 digits and modulo 65536. To be changed in the header definition of INIT.SRV.
 %n : Size of the message header, unformatted.
 %O : Callsign of sysop.
 %P : Origin of the message header, unformatted.
 %R : Callsign with full path (e.g. F6FBB.FMLR.FRA.EU).
 %r : = D if the message holds data (7+) else is space.
 %T : Name of the current theme.
 %t : Number of bulletins in the current theme.
 %X : Number of messages for the user.
 %x : Number of new messages for the user.

 %y : Year of the message creation.
 %% : Character %

12 MISCELLANEOUS TOOLS:

A set of tools is available to the SysOp. These tools are for manual or automatic maintenance. In most cases, these tools should be handled with caution, the process can be dangerous for the files.

Avoid using those tools while the server is in operation (the F9 command, as for an example, in DosFBB), because the modification of some files during operation can yield unpredictable corruptions.

Here is a list of the available tools:

FBBSETUP.EXE
INSTWFBB.COM
INSTALL.SH
EPURMESS.COM
EPURWP.COM
LOGSTAT.EXE
SATUPDAT.EXE
SLEEP.COM
MAKEPG.COM
SETUSER.COM
CLEANUP.COM
MAINTINF.COM
CUT.COM
FV.COM
ISDAY.COM
CLR_USER.COM

For LinFBB:

EPURMESS (works as EPURMESS.COM)
EPURWP (works as EPURWP.COM)
FBB_VIEW (works as FV.COM, but takes no parametres)
FBB_ZM
XFBBC
FBB_VIEW

12.1 INSTWFBB.COM (for WinFBB)
INSTALL.SH (For LinFBB)
FBBSETUP.EXE (for DosFBB)

INSTWFBB.COM: (Only for WinFBB)

This program is a full-screen configuration of INIT.SRV and PORT.SYS.
This program has its own program-icon, and can be started by double-clicking that icon. Most of the commands are self-documented.
You must only follow the instructions. The changes in INIT.SRV can be done without rebooting the BBS, but changes in PORT.SYS will not affect the BBS until after a reboot of the BBS.

INSTALL.SH: (Only for LinFBB)

This is a configuration-program for LinFBB. It will configure the most important parts of the file INIT.SRV. Mind you, you will need to check the rest of INIT.SRV manually, too !

FBBSETUP.EXE: (Only for DosFBB)

This program is a full-screen configuration of INIT.SRV and PORT.SYS.
Most commands are well documented. You must only follow the instructions.
FBBSETUP will support multiple language files in same format as ENGLISH.OLH and ENGLISH.MNU if the DEFAULT.OLH and DEFAULT.MNU files do not exists (after a language is selected, the files become the DEFAULT.XXX files).
Files normally belonging to FBBSETUP are:

FBBSETUP EXE
ENGLISH OLH
ENGLISH MNU
DEFAULT OLH
DEFAULT MNU

The DEFAULT files may first be created after the first run of FBBSETUP.
All files will normally reside in \FBB\BIN directory.

For all the above versions:

If the INIT.SRV file does not exist, you will be prompted to create a new

one. Be sure that you are in the FBB directory when running FBBSETUP.EXE.

If the INIT.SRV is not of the right version, you will also be prompted, but be careful, as the results could be hazardous.

The comments of the original file are not changed.

12.2 EPURMESS.COM:

EPURMESS is a tool which allows for modification of messages: status, removal, or archiving. This tool is called every night by the BBS in order to maintain the message list as a function of the time.

The EPURMESS configuration is made by the text file EPURMESS.INI. The parameters which are contained are given as a suggestion, and can be edited later if needed, to suit the targeted archiving and the mass of messages and bulletins received.

EPURMESS searches the current directory for its EPURMESS.INI initialization file. In that file, it will find out all the parameters for processing, as well as the file name for the report EPURMESS.RES (this file name can be changed in the EPURMESS.INI file).

At the beginning of the process, EPURMESS first copies DIRMES.SYS into DIRMES.OLD to keep an archive trace of the previous file, then creates DIRMES.NEW, in which the processing will take place. At the end of the process, DIRMES.NEW will be copied into DIRMES.SYS. A report of the processing will be written in the EPURMESS.RES file.

If the last update of the DIRMES.SYS file has been carried out more than 24 hours ago, EPURMESS will not make its process, in order to avoid any loss of data due to a possible clock error in the system. It will report the error in the EPURMESS.RES file. In such a case, after the reason of the error has been detected, it will be necessary to suppress the EPURMESS.RES report file, in order to re-enable the processing.

The ARCHIVE line has two digits, the first one applies to private messages and the second to bulletins. The '1' digit validate the archiving while the '0' will suppress the messages once and for all.

It is possible to specify origins, destinations or routing for which the obsolete time (X status) are different from the default values. These particular cases are specified at the end of the file with the form of supplementary lines. Each line specifies a particular case.

The line should begin with the character which specifies the field to be tested, then the content of the field and then the number of days before the status X is given.

Ex :
> ALL 1

The messages for ALL will be valid only one day.

Example of EPURMESS.INI file:

```
# File for deciding messages' lifetime..  
#  
# Directory for messages  
\FBB\MAIL\
```

```
#
# Directory for binary-messages
\FBB\BINMAIL\
#
# Directory for "killed" messages
\fbb\OLDMAIL\
#
# The file DIRMES.SYS (database-file)
\fbb\system\DIRMES.SYS
#
# Old database-file (backup)
\fbb\system\DIRMES.OLD
#
# New database-file (after EPURMESS (cleanup))
\fbb\system\DIRMES.NEW
#
# Result of cleanup is put into this file:
\FBB\EPURMESS.RES
#
# Personal Bulletins (0=kill 1=archive in oldmail)
0 0
#
# Parametres in days:
#
# PARAMETRES FOR PERSONAL MESSAGES:
#
# PN -> PX (days after message is written)
30
#
# PY -> PX (days after changed to Y)
7
#
# PF -> PK (days after changed to F)
7
#
# PX -> PK (days after changed to X)
0
#
# PK -> PA (days after changed to K)
0
#
# PARAMETRES FOR BULLETINS
#
# BN -> BX (days after the message is written)
7
#
# B$ -> BX (days after the message is written)
7
#
# BY -> BX (days after changed to Y)
7
#
# BX -> BK (days after changed to X)
14
#
# BF -> BX (days after message is written)
14
#
# BK -> BA (days after changed to K)
0
#
# Automatic generated return-messages for non-forwarded messages.
# Each line may be max 255 characters long. Disable with only a 0.
```

```

#
# Timeout for valid forward-path (number of days + text, end with $W)
21 This message could not be forwarded, due to bad link.$W
#
# Unknown routing or wrong H-address (number of days + text, end with $W)
2 This message could not be forwarded, due to bad callsign or bad route.$W
#
# Max number of lines in a returned message:
10
-----
#
# List of lifetime-tests
#
# Type To      Days
#@  VEST      365
#@  LA        365
#>  LA6CU     365
#<  system    0
#
# End of this file.
#

```

12.3 EPURWP.COM:

EPURWP is a tool which allows updates and modifications of the white pages database.

Two optional parameters may follow.

The first will specify how many days will go by before a data is valid. If the temporary part of the record has not changed during this time, it will be considered as stable and transferred to the primary (used) part.

The second parametre states how many days will go by before the record is deleted from the database. This is how many days go by with no updates or refreshes of this user's data.

12.4 LOGSTAT.EXE:

Statistical analysis software for the FBBLOG file written by FC1MVP.

The configuration file for this software is in the text file LOGFBB.CNF, which should be located in the same directory.

LOGFBB.CNF file layout :

This file is made of several lines indicating:

- BBS Callsign
- the name of the SysOp port, a comment.
- The abbreviated name of the 1st port, a comment.
- The abbreviated name of the 2nd port, a comment.
- and so on until the last port (8 ports maximum).
- Three stars (***) at the beginning of the line means the end of the ports.
- The BIOS screen access option.

The BBS callsign must have 6 characters maximum. It will be truncated should it be longer.

The abbreviated port name may have up to 10 characters, it is recommended to limit at 4 (ex VHF1 or HF2).

A comma separates the abbreviated name from the comment which follows it, without any space. The comment may have 10 characters too (frequency).

At the end of the comment, it is possible to add a comment which is internal to the CNF file, and which will not be used elsewhere, using the ":" as a separation.

If the BIOS line has a NON at the beginning, the screen access will be direct. The word OUI means that the screen access is made via the BIOS. The direct access improves the processing speed by 50 percent but is careless with DesqView.

Example of LOGFBB.CNF file:

```
F6FBB : BBS name.  
SYSOP,at keyboard : port name @ , comment.  
VHF, 144.675 : Port A , comment.  
VHF, 145.275 : Port B , comment.  
UHF, 430.675 : Port C , comment.  
HF, 21.107 : Port D , comment.  
SHF, 1299,675 : Port E , comment.  
MODEM, Telefone : Port F , comment.  
*** End of ports definition.  
NON : Screen access through BIOS (OUI) or  
Direct (NON).
```

Software limitations :

The software is able to handle a log file within the following limits:

- Maximum number of days : 35, or 5 weeks if you chain 5 log files.
- Maximum number of channels: 50
- Maximum number of ports : 8+1, corresponding to ports A to H handled by the BBS and the SysOp keyboard "port".
- Maximum nr of callsigns : 300 (This is not the number of the call signs known by the BBS, but the number of calls which were used the BBS during the log period.

In case an overflow occurs, the mention AUTRES for the callsign and 99 for the day will be displayed in the corresponding areas.

Regarding the forward efficiency, only the stations which have actually forwarded will be given a value, the others will be mentioned "nc".

Usage :

On the same diskette (or the same disk) there should be at least the program LOGSTAT.EXE, the config file LOGFBB.CNF and enough disk free space for the result files to be created. The log file may stay on another diskette.

For the startup you should type in LOGSTAT FBBLOG.nn (nn is the number of the week) or A:FBBLOG.nn (if the file is located on A:)

The results are recorded into the files LOG.®, LOG.A ... LOG.T being the total of the logs.

Archiving :

A suggestion: you should better archive your FBBLOG.nn file rather than the LOG.x files. You should ZIP them (or ARC or LZH) with the name LOG990nn.ZIP (.ARC or .LZH) with 990 standing for 1990 (000 will be for 2000...) and nn is the number of the week.

Log structure :

The number of resulting files created is equal to the number of ports of the BBS plus 2. The 2 supplementary files correspond to the console port and the sum of the various ports as being the total result.

The first log, named LOG.@ corresponds to the SysOp log (console), the last one, named LOG.T is the summation of the ports. The files LOG.A to LOG.H are the logs of the various ports.

Obviously, if a port has not been used, the corresponding file will not be created.

For each LOG.x file, you find in the following order:

- Diary of the connections per day (y-axis) and the time (x-axis) with the daily total at the end of the line and the hourly total at the bottom of each column.

- The total connection time: it is the sum of the connection times of each user. For the LOG.T file, the total connection time per channel will also be given.

- Mean time per connection : this is the total connection time divided by the number of connections.

- Mean time per user : total connection time divided by the number of users.

- Number of killed messages : self explanatory.

- Number of read messages : idem.

- Number of users.

- A matrix of forwarded messages per day (y-axis) and per hour (x-axis) is made like before. This matrix will not appear if there has not been at least one forward on this port.

- Number of forwarded messages: number of messages sent to another BBS or PMS.

- Number of messages received in forward: number of messages transmitted by other BBS in forward.

- Number of messages received in back-forward: number of messages received from another BBS when forwarding towards it.

- Number of messages refused as already received.

- An array giving the number and the size of the messages received from or sent to another BBS and the efficiency in Bits/s (Max 1200 !).

The BBS's are sorted by alphanumeric order, and the efficiency indication appears for callsigns having conducted only forwarding operations. This matrix will not appear unless there has been at least one BBS forwarded.

- A matrix of the main functions utilized, with on the y-axis the callsigns

sorted by alphabetical order, and on the x-axis the total time of connections in minutes, the number of connections, and the various functions used (see the comment following the array).

In the LOG.T file, the X.F column (forced disconnection entered by the SysOp) is replaced by the number of killing commands. At the end of each column is the compound sum.

Log of a specific station.

- The LOGCALL program yields generation of a log for a specific station

The principle of operation is the same as for LOGSTAT, except that a supplementary calling parameter is required to define the station.

```
LOGCALL FBBLOG.33 FE1ZZZ
```

The generated file will be of the same format as the FBBLOG, but it will enclose only the lines specific to the callsign. The name of the file will match the format FE1ZZZ.LOG, if the requested callsign is FE1ZZZ.

While executing LOGSTAT FE1ZZZ.LOG, you will have the statistics related to that station (in the LOG.A, LOG.B,...).

If you specify the SSID, you will create a log corresponding only to the callsign and the specified SSID. Ex:

```
LOGCALL FBBLOG.33 FE1ZZZ-2
```

Editing the file FE1ZZZ.LOG will allow you to display all the commands sent by that station.

12.5 SATUPDAT.EXE:

SATUPDAT.EXE will read messages from AMSAT copied via packet radio. This is the english version of AJOURSAT.EXE. This software was developed by F6BVP.

It extracts keplerian elements both in NASA and AMSAT formats and updates the results into the satellites data base of the F6FBB's BBS (SATEL.DAT)

Loading SATUPDAT without arguments will print the following message on your screen (between lines -----):

```
-----
Automatic update of satellites orbital parameters for F6FBB's bbs Version
1.77e - June 1991 - Bernard Pidoux, F6BVP
```

```
Usage: SATUPDAT [file name<.txt>] </option> </option>
```

Reading option:

```
/a AMSAT format (default)
/n NASA format
```

Update option:

```
/f merging new and old data (default)
/u update only satellites present in the
```

data base

```
/s keep only satellites present in the input file
```

```
<<767 satellites maximum>>
```

This version 1.77e verifies checksum both for AMSAT files and NASA files. For automatic update of keplerian elements into F6FBB's bbs just create a virtual BBS into BBS.SYS file. Its name could be AMSAT for example.

Then declare the following forward in the file FORWARD.SYS :

```
A AMSAT
*
G AMSAT
P @
C C:\FBB\SYSTEM\SAT\AMSAT.TXT
*
-----
```

CAUTION: The name of the disk unit and the path name should be the exact one of your configuration.

In order to run automatically SATUPDAT when your computer boots, you must modify the batch file like the following example:

In WinFBB (file RUNFBB.BAT):

```
maint ***** START of MAINT script *****
  cd %fbb%
    @echo off
    cd %fbb%\system\sat
    if not exist amsat.txt goto amsatend
    satupdat amsat.txt /n > satupdat.res
    satupdat amsat.txt >> satupdat.res
    del amsat.txt
:amsatend
  cls
  echo
cd %fbb%
```

In LinFBB :(There is no SATUPDAT for LinFBB yet..)

In DosFBB (file APPEL.BAT):

```
:direct
if not exist c:\fbb\system\sat\amsat.txt goto appel
cd \fbb\system\sat
satupdat amsat /n > satupdat.res
satupdat amsat >> satupdat.res
del amsat.txt
cd \fbb
:appel
echo Running the BBS ^C for stopping
sleep 3
serv -t
if errorlevel 2 goto direct
if errorlevel 1 goto suite
:suite
```

(Once again be careful with the disk name and directories !)

SATUPDAT OPTIONS:

The default extension for input file is .TXT, however you can specify another extension.

Both arguments after the filename in the command line are optional: the

first one indicates the reading format, either the AMSAT format or the NASA "2-line" format. The first one is selected by default.

The second option is for the choice of update mode. By default the program will merge the old list with the new one (same if you select option /f). Old data of satellites which are not present in the input file are kept as is and the other one are updated.

Option /u can be selected if you don't want to add new satellites to the one already present. Only keplerian elements from known satellites are updated. This prevents the list to grow with incoming satellites that you are not eventually interested in.

Option /s, on the contrary, gives you the possibility to keep only in the data base the data from the satellites whose name is in the input file.

This can help you to suppress old satellites from the data base.

In all cases the program takes care of the permanent data of satellites such as the frequency (for doppler) and the step for computing the tracking (you must provide both manually only the first time).

If the keplerian elements of the input file are older than the one already present in the BBS data base, the old one are kept.

The list is sorted by name in order to help the finding of the satellite name when someone is connected to the BBS.

If elements for one satellite are duplicated in the input file, the program will look again for the newest data.

Of course, if CRC computed by the program is not the same as the one in the input file, the program informs you that the CRC is BAD and it does not take the keplerian elements.

In the batch file example given above you have certainly noticed that I call twice the program SATUPDAT with two different format arguments. The first time I ask the program to read the NASA format and the second time to read the default format AMSAT. The reason is that the probability for an error to occur in both formats for the same satellite is very small. Thus if an error occurs in one format the program will complete the update when he will found unaffected keplerian elements in the other format. Thanks to the recently adopted checksum by AMSAT!

12.6 SLEEP.COM:

Utility which spends the time (in seconds specified as argument) while doing nothing. It can be interrupted by a Ctrl-C, and allows interruption of a Batch file running.

The syntax to call it is of the type :

SLEEP nn

nn is the number of seconds to wait.

12.7 MAKEPG.COM:

Utility used to generate .EXE (or .COM) program file out of the binary file (xxx.BIN) using English or French language.

This program may not be used in some versions of FBB software !!!
This program needs a .BIN-file (like WFBB.BIN) to convert to an .EXE or .COM file. If a BIN-file does not exist, the program will only be in english version (as a .COM or .EXE file).

The syntax is of the type:

MAKEPG program FR
To generate a program in French

MAKEPG program GB
To generate a program in English.

MAKEPG can transform the following programs :

SERV.BIN -> SERV.EXE
MAINTINF.BIN -> MAINTINF.COM
EPURMESS.BIN -> EPURMESS.COM
EPURWP.BIN -> EPURWP.COM

12.8 SETUSER.COM:

Utility giving the property of a file to a user for further use by FBBDOS.

The syntax is of the type:

SETUSER callsign filename.

The command "SETUSER F6FBB TOTO" will allow F6FBB, and him only, to modify or delete the file TOTO.

12.9 CLEANUP.COM:

Utility killing all messages files unused and not defined in DIRMES.SYS.
This program should be run in the SYSTEM directory.

The syntax is of the type :

CLEANUP mail_directory

Be careful when using this utility. If another directory than MAIL or BINMAIL is specified, it should delete all the files of the directory.

12.10 MAINTINF.COM:

Utility maintaining the INF.SYS file. The INF.SYS file holds the users' database. In case of problem in INF.SYS file, goto the SYSTEM directory and run MAINTINF followed by a number of months. The records of users not connected within this number of months will be deleted and these users will be asked as for a new connection. The number 0 will not delete any record and only check the INF.SYS file.

12.11 CUT.COM:

Utility allowing to peak a part of a file and copy or append it to another file. The format is :

CUT From-file To-file length offset [/A]

To-file will have added up to length bytes of From-file, starting at offset. If the switch /A is specified, then datas will be appended to To-file, otherwise To-file will be replaced.

This small utility is useful to take parts of file, or to do copy/paste with either binary nor ascii files.

Together with the specification of the offset in the XGET command, in case of crash, first note the amount of datas already received (ie : 50300), save the part of the file already received, and then ask "XGET filename 50000". The transfer will start at offset 50000 and then you will be able to receive the end of the file. The paste the two parts, either with COPY command or with the CUT command. It is more secure to cut the first received part before as the last bytes should be garbaged.

12.12 FV.COM:

FV.COM is a small program used by the BBS when a user sends the "View (filename)" command.

View contents of ARC, DWC, LBR, LZH, PAK, ZIP, ZOO files.

Usage: FV [d:][\path\]filespec[.ext] [specs] [/p] [/b] [/w] [/r] [/c] [/x]

Where: /P pauses after each screenful
/B suppresses display of d:\path
/W for wide display
/R update file date
/C displays comments
/X convert sfx to archive

12.13 ISDAY.COM:

ISDAY is a small program to use if you want some program to run only on specific days.

Format : ISDAY [/Nmday] [/Dday] [/Mmonth] [/H]

mday = day of the month, 1 to 31
day = day in the week, 0=sunday .. 6=saturday
month = month in the year, 1=january .. 12=december
All parameters are anded.

Each parameter can be specified as a value, or as an interval like "1,3,12-22", that is the first, third and 12th to 22nd

ISDAY exits with ERRORLEVEL 0 if parameters match else with ERRORLEVEL 1

Example :

```
ISDAY /D0-2
IF ERRORLEVEL 1 GOTO LABEL
PROGRAM TO RUN ON SUNDAYS, MONDAYS and TUESDAYS
:LABEL
```

...

12.14 CLR_USER.COM:

Small program to clear all users's amount of downloaded kb. This is used to "reset" all users, so they will again be able to download the full quota of kilobytes from the BBS.

12.15 FBB_ZM:

(Only for LinFBB)

Documentation for FBB_ZM

This utility is for sending and receiving files with ZMODEM , YMODEM or XMODEM protocol.

Receive file(s): fbb_zm [rz | rb | rx]

where:

rz = Receive file with ZMODEM
rb = Receive file with YMODEM
rx = Receive file with XMODEM

Sending file(s): fbb_zm [sz | sb | sx] file(s)...

where:

sz = Send file with ZMODEM
sb = Send file with YMODEM
sx = Send file with XMODEM

12.16 XFBBC:

XFBBC is a tool that allows to talk with LinFBB daemon.

Main features are:

- remote connection to the xfbbd console
- remote display of the traffic
- remote display of the monitoring

The command line is:

xfbbc [-c | -m channel] [-h hostname] [-p port] [-i mycall] [-w password]

without parameter, xfbbc defaults to "xfbbc -m 0"

Command line options:

-c : console connection. In this case, a callsign and a password must be provided. To provide them, you can use either command line options or environment variables.

-m channel : monitoring request.

channel may be

- * 0 : traffic activity of all channels is displayed.
- * n : traffic activity of the channel "n" is displayed

* -l : monitoring of all ports

-h : hostname of xfbdd (default localhost)

-p : socket port for the communication to xfbdd (default 1525)

-i : callsign for the console connection

-w : password for the console connection. The password must be the exact string corresponding to the callsign as defined in the passwd.sys file of xfbdd.

some environment variables may default the command lines options :

-h : XFBBC_HOST
-p : XFBBC_PORT
-i : XFBBC_CALL
-w : XFBBC_PASS

12.17 FBB_VIEW:

(Only LinFBB)

FBB_VIEW is a small program used by the BBS when a user sends the "View (filename)" command.

View contents of ARC, DWC, LBR, LZH, PAK, ZIP, ZOO files.

VIEW command of LINUX version calls fbb_view program (or any program defined by the variable \$FBB_VIEW). fbb_view may be a shell script.

13 ASYNC PORT DRIVERS AND TELEPHONE MODEM:

You may need several external drivers for your BBS.

For WinFBB:
FBBCOMM.DRV
BPQ

For LinFBB:
AF_AX25 domain socket
AF_NETROM domain socket
AF_ROSE domain socket
It is necessary to compile the Linux Kernel with SLIP (serial line).

For DosFBB:
ESS.COM
ESSKAM.COM
FBBIOS.COM
BPQ
TFPCX/TFPCR

Installation of a TELEPHONE MODEM.

13.1 FBBCOMM.DRV:

FBBCOMM.DRV: (Only for WinFBB)

The port-driver FBBCOMM.DRV is installed in Windows during the installation-process of WinFBB.

FBBCOMM.DRV is automatically installed in your Windows as you install WinFBB from the diskette. So there is nothing for you to do. The driver is automatically installed into your SYSTEM.INI in the [boot] section as comm.drv=fbbcomm.drv. If in doubt, you may want to check this.

13.2 ESS.COM:

ESS.COM: (Only for DosFBB)

ESS is an async port driver optimized for FBB software. It can drive either TNC2/PK232/clones or KAM.

This small device driver is loaded as resident at the beginning of the machine initialization (TSR).

One ESS handles one serial line and makes use of one IRQ.

The COM number, the address of the serial interface and the IRQ number are given as parameters appended to the command line. The address should be given as a FOUR digit hex number.

Example : Install ESS for COM1 at hex 03F8 and using IRQ 4 :
ESS 1 03F8 4

If only the COM number is specified, the address and the IRQ will default to the standard DOS value. Thus ESS 1 would have had the same effect than the example.

Any address or IRQ number can be specified. Carefully check that the address and the IRQ number match the board you intend to use, special attention should be given to the IRQ number which should not already be used by another peripheral device in the machine. On a PC the IRQ's can not usually be shared.

Should there be more than one port to handle, several ESS can be cascaded. The IRQ should be different for each ESS, except for the case in which the interrupt sharing is supported by the serial card used.

Example :

ESS 1 03F8 4
ESS 2 02F8 3

This sequence will load ESS's for COM1 and COM2.

ESS accommodates also the muxing arrangement of serial ports.

Version 1.08 of ESS-driver (included with DosFBB distribution) must be used with the MUX !

13.3 ESSKAM.COM:

ESSKAM.COM: (Only for DosFBB)

ESSKAM works exactly as ESS.COM, but only for Kantronics Kam TNCs.

13.4 FBBIOS.COM:

FBBIOS.COM: (Only for DosFBB)

FBBIOS is a serial line driver dedicated to the usage of a PSTN modem. Binary as well as ASCII file transfer are supported with the XMODEM protocol.

FBBIOS handles one serial line, one IRQ is required. The COM number, the address of the serial interface and the IRQ number are given as parameters appended to the command line, the address should be specified in FOUR hex digits.

Example : Install FBBIOS for COM1 at hex 03F8 and using IRQ 4 :
FBBIOS 1 03F8 4

If only the COM number is specified, the address and the IRQ will default to the standard DOS value. Thus FBBIOS 1 would have had the same effect than the example.

Any address or IRQ number can be specified. Carefully check that the address and the IRQ number match the board you intend to use, special attention should be given to the IRQ number which should not already be used by another peripheral device.

On a PC the IRQ's can not usually be shared.

If the modem in use is at the HAYES standard, FBBIOS can modify the line speed according to the one of the incoming call connection. In this case the parameter 'A' should be appended to the command line to activate this feature. This is not true with some buffered modems using a junction with a settled speed.

Example : FBBIOS 1 03F8 4 A

Upon receipt of the string CONNECT 1200 or CONNECT 2400, the driver will adjust its speed to the incoming call.

From version 1.30, FBBIOS.COM can run up to 56 KBauds. Be sure that your computer is fast enough !

14 SERVERS:

The BBS software offers various servers which are triggered when machinable messages are received. The number of offered servers is without limit.

The servers are activated when receiving a message whose destination had been given at the end of the INIT.SRV file.

Declaration of servers at the end of INIT.SRV :

```
#
# Indicateur de voies
11
#
-----
#
# Liste des services :
```

```
#
# Desti Nom du programme
#
REQDIR REQDIR.COM
REQFIL REQFIL.COM
NEWDOC NEWDOC.EXE
#
# Fin du fichier d'initialisation
#
```

Servers are exec programs whose behaviour is explained in appendix 9. They should be found in the directory containing the exec program SERV.EXE and the configuration file INIT.SRV.

When a private message is received, whose destination is defined in the INIT.SRV file, the corresponding program to the destination is called with the name of a temporary file on the command line. This temporary file contains a message function of the requested server (see description of servers below).

At the completion of the server program, the incoming mail file is checked to import an answer, should there be one. The message addressed to the server and the temporary file are erased after this processing.

Various services can be developped, time available has its limits, but the imagination has not !

```
REQDIR : Request directory-listing from a BBS's files-area.
REQFIL : Request an ASCII-file from a remote BBS.
NEWDOC : Update DOC-area of the BBS.
REQCFG : Request a list of the BBS's configuration.
MULTI  : Send multippel identical messages to different adresseses.
```

14.1 REQDIR server:

The REQDIR server sends back to origin of the special message the list of files (total or partial) of a directory.

The standard command line of the message is used. The message should be of the private type and addressed to the BBS that we want to ask :

```
SP REQDIR @ bbs-destination
```

The title of the message contains two information, the mask of the selection (*.EXE, *.* , etc...) and the @ sign followed the return bbs, that is to say yours. The return message will be addressed to the sender of the request via the return BBS. The mask can not be specified. In this case, all the files will be listed.

```
Mask @ bbs-return <- mask of the selection
```

The text of the message is not necessary, its content will be ignored by the server.

Example :

```
F6FBB BBS >
SP REQDIR @ F6ABJ.FRA.EU
Title of message :
YAPP\*.ZIP @ F6FBB.FMLR.FRA.EU
Text of message :
```

/EX

We assume with this example that YAPP is a sub-directory of the USERS directory (this a suggested configuration).

14.2 REQFIL server:

REQFIL server allows to send back the ascii file requested to the originator of the message.

The message sent to REQFIL has a well defined syntax. The standard command line of message is utilized. The message should be of the private type and addressed to the BBS which we want to ask.

SP REQFIL @ bbs-destination

The title of the message contains two information, the name of the requested file and the @ sign followed by the return BBS, that is to say yours. The return message will be addressed to the originator of the request to the return bbs.

filename @ bbs-return

The text of the message is not necessary, its content will not be taken into account by the service.

Example :

```
F6FBB BBS >
SP REQFIL @ F6ABJ.FRA.EU
Title of message :
DEMOS\ESSAI.TXT @ F6FBB.FMLR.FRA.EU
Text of message :
/EX
```

This example makes the assumption that DEMO is a sub-directory of the USERS directory.

The answer of the remote mailbox will be of the form :

```
SP F6FBB @ F6FBB.FMLR.FRA.EU < F6ABJ
Req File: DEMOS/ESSAI.TXT
Contents of ESSAI.TXT file
...
/EX
```

14.3 NEWDOC server:

The NEWDOC server allows a simple and fast update of the documentation database (topic DOCS of the server). This server will create the file and will give it a label. It will not create a directory if it doesn't exist. Therefore it is necessary to prepare the directory tree structure before.

A message will give a summary to the originator.

The standard command line for messages is used. The message should be of type private and addressed to the BBS which will be updated.

SP NEWDOC @ bbs-destination

The message title contains two information , the name of the file requested followed by the label to give to the file (see the description of the topics DOCS).

Filename Label eventually on several words.

The text of the message will be transferred in the documentation file.

Example :

```
F6FBB BBS >
SP NEWDOC @ F6ABJ.FRA.EU
Title of message :
BBS\DOC_1.DOC This is the label of the documentation.
Text of message :
This is the content of the documentation.
....
End of documentation.
/EX
```

This example assumes that the BBS is a subdirectory of DOCS, thus one of the topics of the documentation.

If the NEWDOC.SYS file is specified, the server will be limited to the callsigns declared in this file, one callsign in each line.

Example of NEWDOC.SYS file :

```
F6FBB
FD1CDC
```

14.4 REQCFG server:

REQCFG server allows to send back the current configuration of the software.

The message sent to REQCFG has a well defined syntax. The text of the message is not used. The message should be of the private type and addressed to the BBS which we want to ask.

```
SP REQCFG @ bbs-destination
```

The title of the message contains the @ sign followed by the return BBS, that is to say yours. The return message will be addressed to the originator of the request to the return bbs.

```
@ bbs-return
```

The text of the message is not necessary, its content will not be taken into account by the service.

Example :

```
F6FBB BBS >
SP REQCFG @ FD1CDC.FMLR.FRA.EU
Title of message :
@ F6FBB.FMLR.FRA.EU
Text of message :
/EX
```

The answer of the remote mailbox will be of the form :

SP F6FBB @ F6FBB.FMLR.FRA.EU < F6ABJ
ReqCfg V 1.2 (C) F6FBB 1992 - BBS F6FBB

Software F6FBB Version 5.15 compiled on Feb 07 1993

Mem Us:14620 Mem Ok:148288 Bid:30000 Ports:4 Ch:12 FBB Ok BIN Ok

Available volumes : C: D: E:

EMS V4.0

MSG : 29 pages (464 KB)
BID : 17 pages (272 KB)
HIE : No
FWD : 1 pages (16 KB)
REJ : 1 pages (16 KB)
OVR : 11 pages (176 KB)
SCR : 45 pages (720 KB)
WPG : 5 pages (80 KB)
Total : 109 pages (1744 KB)

Languages

1 : ENGLISH
2 : FRANCAIS
3 : ESPAGNOL
4 : ITALIANO
5 : NEDERLAN
6 : DEUTSCH
7 : CATALA
8 : NORSE
9 : PORTUGUE
10 : DANSK
11 : SVENSK
12 : HRVATSKI
13 : FRANC8

Servers

REQDIR : Demande de repertoire
REQFIL : Demande de fichier ascii
NEWDOC : Ajout de documentation
REQCFG : Demande de configuration
AUTO7P : Serveur 7plus
7PSERV : Serveur de fichiers 7PLUS
SWFMLR : Distribution Infos Reseau
WP : Serveur W.P.
MULTI : Multi-addresses

Port	Interface	Emulat.	Ch	Mode	Frequency
1	DRSI	WA8DED	9	UYWL	NODAL
2	ESS/COM.	KAM	1	GYW	15M/20M
3	FBBIOS	FBBIOS	1	UYM	MODEM
4	ESS/COM.	KAM	1	GYW	HF-PSK

14.5 MULTI server:

MULTI.EXE, server for FBB-BBS.
(By LA6CU, Per)

This server is used to send personal copies of special messages. The original is addressed to a special "call", and then there is automatically made personal copies to everyone that shall have one.

MULTI may work in 2 ways. You may use one of them, or both. I use both at the same time.

New in version 3.00:

The server may use different languages. There is a new file MULTI.TXT (reside in SYSTEM-directory) that holds all languages. There is one "module" for each language. See the text in MULTI.TXT for more info.

1: WITH DAT-FILES:

Installation:

MULTI.EXE is put in the same directory as SERV.EXE. MULTI will read from INIT.SRV...

In INIT.SRV, MULTI is included together with REQFIL, REQDIR etc. Example:

```
#
-----
#
REQFIL  REQFIL
REQDIR  REQDIR
CLUB     MULTI
TEST     MULTI
#
```

In this example all messages sent to
SP CLUB
will be copied to all callsigns in the file CLUB.DAT, while messages sent to
SP TEST
will be copied to all callsigns in the file TEST.DAT.

CLUB.DAT and TEST.DAT (in this example) must stay in the FBB-directory.
You must make one such .DAT-file for every to-field you define in INIT.SRV.

The .DAT-files must consist of callsigns (@BBS and H-addresses allowed) for all those that shall have a copy.

There are 4 different copy-types.

A (action)	:	This is the callsign that will receive the message, or perform the action described in the message.
C (copy)	:	This callsign receives a copy of the message.
I (information)	:	This callsign receives a copy for information.
H (hidden)	:	This callsign receives a copy, but his callsign will not appear in the footnotes of the message, telling who has received a copy.

There may be more than one callsign on each line (starting with A, I, C or H) separated by a comma. There may be maximum 100 callsigns altogether.
Comment-lines are extras, they MUST start with a #

From version 3.00 there will also be a separate line with a letter corresponding to the language the server will use. What letter corresponds to what language, is explained in MULTI.TXT.

Example of CLUB.DAT:

```

E               <----- Use ENGLISH language for CLUB-server
A LA6CU, LA1B, LA2D <----- Copies to the people doing the job
A LA6GDA        <----- " " " " " " " "
                  May be several lines if there is not
                  enough room in one line.
I LA5RBA        <----- LA5RBA is informed on the matter
C LA6IM @ LA6IM <----- LA6IM receives a copy for updating his
                  node-lists
H LA2GIA        <----- LA2GIA receives a copy, but will not appear
                  in the list of callsigns at the end of
                  each message.

```

Beware: The .DAT-files shall consist of callsigns and comments, nothing else ! No /EX, no * and no *** end of file.

MULTI will NOT send a copy back to the originator, even if he also is in the .DAT-file. This to avoid "looping" of messages.
MULTI will send an acknowledge-message back to the originator, telling him that the copying was successful, and show him a list of all callsigns that will receive such a copy.

R:lines on incoming messages are removed, until there has arrived 2 lines without R: After that. R:lines are accepted again. This is done to make it possible also to forward messages back the same route, and at the same time make it possible to include R:lines in the message itself.

R:lines are NOT checked to prevent forward to BBSs that are already in these R:lines, because a message to a MULTI-server only will be processed at the end-BBS anyway.

MULTI needs very little memory (I have not checked how much..), but I have been told that it needs less than 35k. MULTI does not write to screen.

MULTI.EXE must be in the same directory as SERV.EXE.
From version 3.00 the .DAT-files must be in the SYSTEM-directory.
MULTI.TXT must also be in SYSTEM-directory.

MULTI makes a temporary file MULTI.TMP on disk, and delete it afterwards. Everything else is done in memory, and the result is placed in MAIL.IN, as all servers do.

2: WITH ALL CALLSIGNS DIRECTLY IN THE ORIGINAL-MESSAGE.

MULTI still works exactly as before, with the same installation. But after a request from F6ABJ and F6FBB I have made a modification so that users themselves can decide what callsigns will receive a copy. This is done by including the callsigns in the message itself.

In INIT.SRV, MULTI is still included like before (see further up). But in addition, you now must have 1 extra line, like this:

```

#
-----
#
REQFIL  REQFIL
REQDIR  REQDIR
CLUB     MULTI
TEST    MULTI
MULTI    MULTI          <----- This line is new.
#

```

From version 3.00 SP MULTI uses the default language (see MULTI.TXT).
Other languages may be configured. For example you may have:

```
MULTI   MULTI   (for default language)
MULTIE  MULTI   (for english language)
MULTIF  MULTI   (for french language)
MULTIH  MULTI   (for netherlands language)
MULTIN  MULTI   (for norwegian language)
```

So, you may send the same message SP MULTI, SP MULTIE, SP MULTIF, SP MULTIH, SP MULTIN etc. The first 5 letters must always be MULTI, when used without DAT-file. Default language will typically be english (SP MULTI), but the users will have the possibility to use others, as configured.

Users will now be able to send messages to MULTI, like this:

```
SP MULTI      The program will now collect the info on who will receive a
copy, from the message itself, and not from a separat .DAT-file like
SP CLUB or SP TEST still will do.
```

There are 4 different copy-types.

```
A (action)      : This is the callsign that will receive the message, or
                  perform the action described in the message.
C (copy)        : This callsign receives a copy of the message.
I (information) : This callsign receives a copy for information.
H (hidden)      : This callsign receives a copy, but his callsign will not
                  appear in the footnotes of the message, telling who has
                  received a copy.
```

(In addition one may use CC: instead of only C for Copy. Special request..)

There may be more than one callsign on each line (starting with A, I, C or H) separated by a comma. There may be maximum 100 callsigns altogether.
Comment-lines are extras, they MUST start with a #

Example:

```
SP MULTI
Digipeater upgrade      <----- Title of message
A LA6CU, LA1B, LA2D     <----- Copies to the people doing the job
A LA6GDA                 <----- " " " " " " " "
I LA5RBA                 <----- LA5RBA is informed on the matter
C LA6IM @ LA6IM          <----- LA6IM receives a copy for updating his
                           node-lists
H LA2GIA                 <----- LA2GIA receives a copy, but will not appear
                           in the list of callsigns at the end of
                           each message.
-----                 <----- After the last callsign, there must be a
                           string of at least 5 '-' before the
                           actual message.
```

Then follows the contents of the message, and end with Ctrl-Z or /EX.

As usual, the sender will receive a message saying that the copies were made. The messages to all the other callsigns will be identical, and look like this:

Message via LA6CU MULTI-server (v3.00) :

Contents of message

```
=====
For action      : LA6GDA, LA6CU, LA1B, LA2D.
For copy       : LA6IM.
For information : LA5RBA.
=====
```

All copies are identical, but A, C and I determines in which line in the

footnotes the callsigns will appear. A callsign after an H will not appear in the footnotes.

73 de Per, LA6CU.

15 WHITE PAGES, Database and server.

DESCRIPTION.
UPDATE REQUESTS.
DATABASE DESCRIPTION.
DATABASE MANAGER.
EPURWP AND UPDATE MESSAGES.
WP SERVER REQUESTS.

Description.

The White Pages implementation in FBB software has been based upon the WORLI model (many thanks to Hank for his work). I've tried to maintain a high degree of compatibility whilst making further development to my own criteria.

I shall try to explain how FBB White Pages works.

I have probably mis-understood some features of WORLI's specifications but I hope that this will not greatly affect the compatibility.

First of all, why do we need White Pages?

White pages has some interesting features. Not least :

- A dynamic database containing users Name, zip code, HomeBBS and QTH (as well as other fields).
- Automatic addressing/routing of mail to the HomeBBS of the destination callsign.
- A White Pages server for remote interrogation of the database.

The database information is updated, firstly from the information given by users when they exercise the N, NH, NQ and NZ features at their home (or another WP equipped) BBS; and secondly, from information contained within the messages headers as they traverse the Network.

The database is dynamic, it is changing constantly, and it updates itself in real time. Either as soon as a line of a message header is received when in ASCII forwarding mode, or when a complete message is decoded in compressed forwarding mode; or else when a user disconnects from the BBS (this is to prevent multiple updates being generated during a session).

So, the database can hold many callsigns. In fact it maintains a list of all the callsigns seen from all individuals sending messages as well as all of the BBS's seen in the forwarding paths. More than 10,000 valid records is not impossible today, and this will surely increase as the number of packet radio users grows with each day. This will allow user to send messages to other users around the world without necessarily having to be concerned to find their full Hierarchical Address, the old principle of the user typing:

BBS PROMPT >
SP K6VAZ @ KM6WU.#CENCA.CA.USA.NOAM

should now be replaced by the user entering:

BBS PROMPT >

SP KM6VAZ

The BBS will add the HA and send the response:

```
BBS PROMPT >
SP K6VAZ
WP ROUTING @KM6WU.#CENCA.CA.USA.NOAM ADDED
TITLE ?
```

If the routing destination HA is not recorded in the database then the user will be advised and prompted to enter the address manually.

Another capability of FBB White Pages is the automatic sending of update messages to other BBS's. These messages are generated every night during House-Keeping and are a listing of the additions and modifications made to the database during that day. These messages are sent addressed both to and from WP.

When passing through or terminating at another White Pages equipped BBS, the message will automatically update the 'local' WP database at that BBS. This feature MUST BE USED WITH CARE, as updates can generate a lot of traffic and the Network must be able to support it.

*** It's not be a good idea to send these update messages on HF ! ***

A built-in White Pages server (WP) will provide information from the database in response to a remote request. This server is described in paragraph xx.

All files used by White Pages are in the FBB\SYSTEM\WP subdirectory.

Trace for WP updates (for debugging etc):
in the \windows\winfbb.ini file, add the following line in the main section :
TraceWp=1
You can replace 1 with 2 or 3. 3 gives the maximum information.
A file WP.DBG will be created in the WP directory.

UPDATE REQUESTS.

The database receives information from three sources. The s indicated on each line of the update message as a suffix to the callsign:-

- The /U suffix denotes that the information in this line of the update is User-Generated as is therefore assumed to be CORRECT. This information is collected by the BBS whenever the User responds to the N, NH, NQ or NZ commands. The date associated with the information is the date when the User disconnects that session.

- The /G suffix denotes that the information in this line has been gathered by examining the header of a message to GUESS at which BBS the sender is registered. The HomeBBS of the User is assumed to be the BBS shown in the first R: header line. The date associated with this information is the date shown on this R: header line.

- The /I suffix denotes information about forwarding BBS's taken from the R: header lines. This information can consist of the HA (the Hierarchical Address), the QTH (within brackets) and the zip code (following the Z:). The date of this information is again taken from the R: header line of the BBS in question.

When the BBS is idle the Database Manager is called and the update

information detailed above is processed.

DATABASE DESCRIPTION.

The database is composed of individual records. Each record following components :

- Callsign and Name.
- Active information.
- Temporary information.

The active and temporary information components are identical and each includes the following fields:

- Date of the information
- Hierarchical Address (one word)
- Zip code (one word)
- Qth (one or more words)

Only the Active information is used for addressing/routing and database requests.

DATABASE MANAGER.

This process freshens the database, following receipt of the new or changed information detailed above.

The update subroutine will first look for an entry in the database for the callsign which matches the received information. If it does not exist then a completely new record will be created in the database and the information be used to fill what fields it can, in both the active and the temporary components. The date will be then changed to the one associated with the update information.

If the record does already exist, then the unknown fields of both the temporary and active fields will be filled in, and those fields already known in the temporary part will be replaced by the new information if the date new information is younger than that already on file. The date will then be adjusted such that it is consistent with the updated information.

If the new information is of the /U category, then the current fields will be replaced by the new information in both the primary and secondary (Active and Temporary) parts of the record, as this information has been input directly from the user. If the information was of another category then only the secondary (Temporary) part of the record will be updated, so the Active or primary record will remain unchanged at this time.

If a field is changed, a flag giving the update request type is then validated. If the /U flag is already validated, it will not be replaced. This flag will be used in case the WP update messages are validated.

EPURWP AND UPDATE MESSAGES.

EPURWP is a maintenance program for the White Pages database which should be run during each House-Keeping cycle.

The program conducts a validity check on each of the entries, and discards any "unwanted" records (in the case of an invalid callsign for example).

The program also checks the date of the last update of the temporary part of

each record. If this date is older than a pre-defined number of days (given as a parameter, default 40 days) then the temporary part is considered as stable, and then the known fields will be transferred to the Primary or Active part, which is then used to answer all addressing/server requests.

This process ensures that the database is tolerant of users sending messages from mailboxes other than their normal HomeBBS. Once the Active or primary part of the record is set, then the temporary (or secondary) part can be updated/changed many times. Only once this temporary field has remained unchanged for 40 days, or the user exercises any of the "Nx" options at his new HomeBBS will the Active or Primary record be changed.

If the changes to the database are validated, then the record is marked with an update flag and a line will be appended to the file MESS.WP

Each line of the outgoing WP update messages looks like :

On 930123 FD1CDC/U @ F6FBB.FMLR.FRA.EU zip 31240 Claude Saint Jean

Any unknown fields are replaced by "?" like :

On 930123 FD1CDC/U @ F6FBB.FMLR.FRA.EU zip ? ? Saint Jean

The U character is the update type.

WP SERVER REQUESTS.

FBB software has an internal built-in WP server.

The format of the WP server requests are as shown below :

```
BBS PROMPT >
SP WP @ F6FBB
Title of message
WP Request (does not matter)
Text of message
```

F6FBB ?

EA3* ?

^Z (or /EX)

The server will answer to the request with a private message, addressed to the sender, and routed to the BBS according to the first R: header line of the incoming request.

The reply message is restricted to a maximum of 100 lines, as the use of wildcards in the request could generate a unacceptably long replies.

16 FORMAT OF SYSTEM FILES (see also chapter 4.1 - 4.24 for most-used files):

The miscellaneous files used for the configuration and the operation of this BBS software have a very accurate syntax. The good behaviour of your software depends upon how these files have been configured.

The first file to be configured is INIT.SRV. This file contains the information on the BBS and about your hardware.

The system holds more than 10 text files which can be configured. Do not forget to save the previous content of the file before modification. This

will ease the restore which might often be necessary. Never delete a file, you might loose major data. Never modify a binary file. Some files have a preset number of lines, and adding or deleting lines might generate an error message. This is the case of the configuration files and text files.

In most of the configuration files and within the text files, the character # in the FIRST COLUMN means that the line is a comment, and will not be taken into account for the number of lines.

Files may stay in different directories, FBB, BIN, SYSTEM, etc... The default directory is indicated after the file name inside parentheses.

Standard system-files (for sysop to change):

RUNFBB.BAT	Startup-file.	Only for WinFBB
XFBB.SH	Startup-file.	Only for LinFBB
APPEL.BAT	Startup-file.	Only for DosFBB
DSZ.BAT	Start binary transfer	
INIT.SRV	BBS initializing file	
EPURMESS.INI	Configure housekeeping	
PORT.SYS	Configure ports	
SWAPP.SYS	Swap to/from/etc in messages	
INITTNCx.SYS	Startup-parametres for TNC and/or modem	
MAINTx.SYS	Closedown-parametres for TNC and/or modem	
BEACONx.SYS	Configure beacon	
BBS.SYS	Define all BBSs that receive bulletins	
FORWARD.SYS	The main forward-file	
CRON.SYS	Run certain tasks/programs every hour	
REJECT.SYS	List of callsigns to be rejected/held	
PASSWD.SYS	Password-file for remote sysops	
PROTECT.SYS	Define "safe" directories	
LANGUAGE.SYS	Define languages	
THEMES.SYS	Define various theme-zones.	
ERROR.SYS	File for logging errors	
ETAT.SYS	Shows when the BBS was put in service	
HEARD.BIN	Stations heard on radio-ports	
REDIST.SYS	Define bulletin redistribution	
english.TXT	Texts used by the BBS. One for each language	
english.HLP	User-help. One for each language	
english.ENT	Logon-text. One for each language	
english.NEW	Logon-text. One for each language. Shows ONCE	
english.INF	Texts used by the BBS. One for each language	
FBBLOG.nn	Logfile for the BBS. One for each week	
MEMO.SYS	Define "hot keys" (DosFBB and WinFBB different)	
nnnnn.SAT	Characteristics of satellites	
NEWDOC.SYS	Authorize callsigns for NEWDOC-server	

Other file-formats (please do not change unless you are 100% sure that you know what you are doing):
(Incorrectly modifying these files will most likely cause your BBS to work incorrectly, or maybe even crash)

STATIS.DAT	Statistical data of BBS-usage
OPTIONS.SYS	Options for console-usage
DIRMES.SYS	Records for every single message
TPSTAT.SYS	
WFBID.SYS	Records of BIDs
INF.SYS	Records of users
YAPPLBL.DAT	Information on binary files
SATEL.DAT	Information on satellites
WP.SYS	White Pages database
MESS.WP	File from WP tp update BBS

16.1 DSZ.BAT:

DSZ.BAT (\FBB\BIN).

DSZ.BAT is a DOS batch file which prepares the parameters to be given to DSZ. DSZ mostly works using environment variables which are set in this batch file.

Do not change this file unless you really want to do something new !

@echo off

SET DSZLOG=%1
SET DSZPORT=%2,%3

shift
shift
shift
DSZ %1 %2 %3 %4 %5 %6 %7 %8 %9

16.2 PASSWD.SYS:

PASSWD.SYS (\FBB\SYSTEM).

This file is in SYSTEM directory.

The first line in this file is the generic password. That is, this password will be used for all sysops that do NOT have a special password of his own. Be advised to change this to your own, secret password !

The next line is a special password for one special sysop. In this case LA6CU. The two first numbers correspond to the same "security-numbers" in INIT.SRV (check this). So in this case of LA6CU, I will have the security-number 69 before a successful SYS-command, and 1023 after. In the next line the numbers for LA6IM is 123 and 1023.

```
jkshdfkjasdfuweiuryiusdfashdfabsmfzmxncvbiouweqyriuyweruyasjhf
LA6CU 69 1023 jkasdhfiuweyiruyasidkasfuiweriuoasdfkjlalsdfjlha
LA6IM 123 1023 uiweyjhaskjdfhasdfiuyaiuowyeruakjsdfhkjahfdlkja
```

Each line in PASSWD.SYS can have a maximum of 255 characters. For the first tests with the BBS-program, you will only need the first line. Write whatever you like in it, and have a copy of it on paper. And let nobody else see that paper (hi).

To get privileges as a remote sysop, you must first have been declared as a sysop with the EU-command. Set flag S for Sysop.

Next you must send the command SYS to the BBS. The BBS will answer like:
LA1B-5> 12 53 45 28 92 [0852415395]

You will have to respond by 5 characters, the 5 corresponding to the 5 numbers you have received from the BBS. The extra numbers in brackets, the [0852415395], is for an automatic MD2 password decoding system, which has been implemented in TPK and some other terminal-programs (I believe..).

You may ignore the numbers in the brackets.

16.3 ERROR.SYS:

ERROR.SYS file (\FBB)

This file gives a description of the error in the BBS, as it occurs.

This file has a free format, using text. The contents are error reports from the software, mostly with the TNC interface.

16.4 ETAT.SYS:

ETAT.SYS (\FBB\SYSTEM)

This binary file holds informations concerning the state of the BBS. The callsign is the last user of the /A or /R (Stop or Reset) command, and the date this command was issued.

LA6CU -2

Reset demande par LA6CU- le Tor 19/08/93 22:58

16.5 HEARD.BIN:

HEARD.BIN (\FBB\SYSTEM)

This file holds the callsigns of stations heard on each port.

This binary file holds the information on heard callsigns. This file must not be edited or changed. This file can only be opened for reading.

It is organized with records, each record containing information on a heard callsign. Never modify this file while the BBS is running.

Structure of one record (C language) :

```
typedef struct
{
    char  call[7]  ; /* 7  Callsign          */
    char  num      ; /* 1  SSID           */
} indicat ;

typedef struct {
    indicat  callsign ; /* 8  Callsign + SSID          */
    long     first    ; /* 4  Date of first heard      */
    long     last     ; /* 4  Date of last heard       */
    ushort   nb       ; /* 2  number of heards         */
} Heard      ; /* 18 bytes = length of one record */
```

Date is given as the number of seconds since january 1st, 1970 00:00.

In C language, all strings are ended with a NULL (00 hex) character.

16.6 TXT, HLP, ENT, NEW, INF:

(for each language)

Specific language files like
x.TXT, x.HLP, x.ENT, x.NEW, x.INF (SYSTEM\LANG).

These various files have the names of the languages followed by the suffixes TXT, HLP, ENT, or INF. They may contain variables and are searched for in the directory holding the files (SYSTEM\LANG).

TXT : file which contains text used during the operation of the software. Each line is associated to a text used by the software. The number of lines must not be altered, except for the comment lines which are not taken into account (lines beginning with a # in the first column).

HLP : Help file called by the ? or H command. The help is referred to a word following the command, (ex: ? HELP). Each help block should begin with a line of a special format.

ENT : Optional file displayed on the user connection, between the header and the list of commands. The FRANCAIS.ENT file will be displayed if it exists and if the caller uses french language.

NEW : Same as ENT above, but shown only ONCE for each user.

INF : File displayed when the command "I" is used. If the file does not exist, the display of the ! command will take place.

16.07 FBBLOG.nn:

FBBLOG.nn File (\FBB\SYSTEM\LOG).

Most of the executed commands are recorded in the file FBBLOG.nn - n is the number of the week. This yields a weekly archive which is not of a too big size.

This file will be updated if the line "Complete Log" of the file INIT.SRV is "OK".

It is an ASCII file which is automatically created, and does not request any special care. This single file keeps the activity record for all ports together.

All the lines begin with a string of figures, corresponding to the month, day, hour, minute and second of the trace, followed by the channel number, then the executed command.

A stats analysis software, LOGSTAT.EXE allows the processing of the log file (see 13.xxx).

FBBLOG file sample :

```
043004460500SI *** BBS Initialize
043004483200SA *** BBS Online
043004483305CB FC1NOU-0 VIA F6FNL-5
043004483905XD
043004484705CB FC1NOU-0 VIA F6FNL-5
```

```

043004500905MR 64072
043004511405XB
043005003201CA F1EBV-14
043005012816CC F1EBV-1 VIA FE6BEX-8 {FC1BPS-7,F1EBV-7,FF6KQK-2}
043005020701MW 64081P F:FC1GQL T:FE1JNY@F6DEG [281] S:F6CQP
043005022816MN B:19226_FF6KNL V:F1EBV
043005424401M> 62893 V:F1EBV [2881]
043005463616MN B:4759_EA6RCM V:F1EBV
043005463616MF 62896 V:F1EBV [393]
043005470813CC FC1HAQ-1 VIA F6FBB-7,019101
043005481701M> 62897 V:F1EBV [872]
043005485816MF 62897 V:F1EBV [872]
043005510701M> 62898 V:F1EBV [815]
043005514516MF 62898 V:F1EBV [815]
043005522901M> 62901 V:F1EBV [398]
043005530516MN B:26051_FC1HAQ V:F1EBV
043005530616MF 62901 V:F1EBV [398]
043005533313MW 64084B F:DF9WM T:ALL@EU [1161] S:Address Needed
043005572701M> 62931 V:F1EBV [2728]
043005580113XD
043005584516MF 62931 V:F1EBV [2728]
043006003012CB F6CDD-1
043006211401XT
043006303216CC FC1HAQ-1 VIA F6FBB-8,019101
043006323016MF 64081 V:FC1HAQ [281]
043006340716MF 64082 V:FC1HAQ [830]
043006354116M< 64086B F:DK3GI T:DXNEWS@EU [953] S:BY4AA in RTTY
043006443216M< 64089B F:OZ2FAR T:ALL@WW [1895] S:re qrt
043006452112CB F6CDD-1
043006465112MF 64086 V:F6CDD [953]
043006465212MF 64087 V:F6CDD [2632]
043006465312MF 64088 V:F6CDD [1626]
043006465312MF 64089 V:F6CDD [1895]
043006465412XB
043006474016M< 64090B F:VS6XQF T:ALL@WW [1748] S:CRY FOR TR-9500
043006490316M< 64091B F:DJ0AW T:ALL@WW [972] S:mfj-1278
043006534816M< 64092B F:DB2OS T:KEPLER@WW [3684] S:Elements 118.
043007002612CB F6CDD-1
043007005516M< 64095B F:VK5CGB T:ALL@WW [1166] S:RNARS PACKET
043007022116M< 64096B F:DF7MX T:COMMODO@EU [913] S:SUPPORT
043007034316M< 64097B F:I1ZEU T:ALL@EU [1096] S:ID1V
043007065416SQ *** BBS Quit

```

All Xx lines in log are disconnections (eXit)

XI Invalid callsign

XE Excluded callsign

XB is normal disconnection by the BBS (after a B command for instance)

XT Time-out disconnection

XF Forced disconnection (after a XI or XE)

XM Too many errors

XP Password error (modem)

16.8 nnnnn.SAT:

nnnnn.SAT (\FBB\SYSTEM\SAT)
(Satellite characteristics).

The nnnnn.SAT files (where nnnnn is the NASA catalog number for the satellite) will be displayed when the command C will be typed from the satellite menu. They are text files and are located in the SAT subdirectory.

16.9 NEWDOC.SYS:

NEWDOC.SYS (\FBB\SYSTEM).

The NEWDOC server allows a simple and fast update of the documentation database (topic DOCS of the server). This server will create the file and will give it a label. It will not create a directory if it doesn't exist. Therefore it is necessary to prepare the directory tree structure before.

A message will give a summary to the originator.

The standard command line for messages is used. The message should be of type private and addressed to the BBS which will be updated.

SP NEWDOC @ bbs-destination

The message title contains two information , the name of the file requested followed by the label to give to the file (see the description of the topics DOCS).

Filename Label eventually on several words.

The text of the message will be transferred in the documentation file.

Example :

```
F6FBB BBS >
SP NEWDOC @ F6ABJ.FRA.EU
Title of message :
BBS\DOC_1.DOC This is the label of the documentation.
Text of message :
This is the content of the documentation.
....
End of documentation.
/EX
```

This example assumes that the BBS is a subdirectory of DOCS, thus one of the topics of the documentation.

If the NEWDOC.SYS file is specified, the server will be limited to the callsigns declared in this file, one callsign in each line.

Example of NEWDOC.SYS file :

```
F6FBB
FD1CDC
```

16.10 STATIS.DAT:

STATIS.DAT (\FBB\SYSTEM).

This binary file holds the information on all connections. This file is very important and must not be edited or changed otherwise the list of the connections could be lost. This file can only be opened for reading.

It is organized with records, each record containing information on a connection. Never modify this file while the BBS is running.

Structure of one record (C language) :

```
typedef struct {  
    char  indcnx[7]      ; /* 7   Callsign                */  
    char  port           ; /* 1   (Port x 32) + channel        */  
    long  datcnx         ; /* 4   Date of the connection      */  
    int   tpscnx         ; /* 2   duration of the connection  */  
  
} statis                ; /* 14 bytes = length of one record */
```

Date is given as the number of seconds since january 1st, 1970 00:00.

In C language, all strings are ended with a NULL (00 hex) character.

16.11 OPTIONS.SYS:

OPTIONS.SYS (\FBB\SYSTEM)

This file holds the different options of the software. Each option is defined as an integer (2 bytes).

```
struct {  
  
    int beep           ; /* 2   Connection beep validated    */  
    int ok_chat        ; /* 2   T command allowed            */  
    int ok_disp        ; /* 2   Indication of channels       */  
    int separe         ; /* 2   Position of the split line   */  
    int split          ; /* 2   DOS screen is splitted       */  
    int gate           ; /* 2   Gateway validated            */  
    int justif         ; /* 2   Justification validated      */  
    int upd_fwd        ; /* 2   Message list is being rescanned */  
    int sed            ; /* 2   Internal screen editor validated */  
  
};
```

16.12 DIRMES.SYS:

DIRMES.SYS (\FBB\SYSTEM).

This binary file holds the information on all active messages. This file is very important and must not be edited or changed otherwise the list of the messages could be lost. This file can only be opened for reading.

It is organized with records, each record containing information on a message. The first record of the file only contains the number of the last message. Never modify this file while the BBS is running.

Structure of one record (C language) :

```
#define NBBBS 80
```

```
#define NBMASK NBBBS/8
```

```
typedef struct {  
  
    char type          ; /* 1   Type of message (A,B,P,T)    */  
    char status        ; /* 1   Status of message ($,A,F,K,N,Y) */  
    long numero        ; /* 4   Number of the message        */  
    long taille        ; /* 4   Size of message in characters */  
  
}
```



```

long date          ; /* 4   Date of message          */
char bbsf[7]       ; /* 7   Adjacent BBS giving the message */
char bbsv[41]      ; /* 41  Route                      */
char exped[7]      ; /* 7   Destination of the message */
char desti[7]      ; /* 7   To field                    */
char bid[13]       ; /* 13  BID or MID                  */
char titre[61]     ; /* 61  Title of message           */
char free[16]      ; /* 16  Reserved bytes             */
long datesd       ; /* 4   Date of the message creation */
long datech       ; /* 4   Date of last status change  */
char fbbs[NBMASK] ; /* 10  Mask of BBSes to forward to */
char forw[NBMASK] ; /* 10  Mask of BBSes already forwarded */

} bullist          ; /* 194 bytes = length of one record */

```

A null type of message (00 hex) invalidates the record.

Dates are given as the number of seconds since january 1st, 1970 00:00.

Masks are bit fields, number of bit corresponds to the number of the BBS in the BBS.SYS file.

In C language, all strings are ended with a NULL (00 hex) character.

16.13 TPSTAT.SYS:

TPSTAT.SYS (\FBB\SYSTEM)

This file holds the statistics of the rubrics. Deleting it will restart these statistics from scratch. Do not delete the file while the BBS is running. You will be prompted to create it when booting the software.

The file is organized as 21 records.

Each record is a long integer holding the number of seconds used for the current record. Each record correspond to the level (first number of the three numbers of the status bar).

```

record 0 : Server menu
record 1 : Message (sending or receiving)
record 2 : Qra locator
record 3 : Statistics
record 4 : Documentations
record 5 : Nomenclature
record 6 : Trajectorygraphy
record 7 : Unused
record 8 : Sending bin message
record 9 : FbbDOS menu
record 10 : Gateway
record 11 : Modem
record 12 : Binary transfer
record 13 : X forwarding
record 14 : Mailbox usage
record 15 : Forwarding
record 16 : Console chat
record 17 : YAPP
record 18 : Conference
record 19 : Editor
record 20 : Themas area

```

16.14 WFBID.SYS:

WFBID.SYS (\FBB\SYSTEM).

This binary file holds the last received BIDs. The number of records of this file is defined in the INIT.SRV file. This file is very important and must not be edited or changed otherwise the list of the messages could be lost. This file can only be opened for reading.

It is organized with records, each record containing information on a BID. This is a circular file and the first record of the file contains the pointer of the last overwritten record in the msg_number field. Never modify this file while the BBS is running.

Structure of one record (C language) :

```
typedef struct {  
    char mode          ; /* 1   Type of message (A,B,P,T)      */  
    char fbid[13]      ; /* 13  BID of the message      */  
    long msg_number    ; /* 4   Number of the message   */  
} bidfwd              ; /* 18  bytes = length of one record */
```

In C language, all strings are ended with a NULL (00 hex) character.

16.15 INF.SYS:

INF.SYS file (\FBB\SYSTEM).

This binary file holds the information on all users of the BBS. This file is very important and must not be edited or changed otherwise the list of the users could be lost. This file can only be opened for reading.

It is organized with records, each record containing information on a user. Never modify this file while the BBS is running.

Structure of one record (C language) :

#define uchar unsigned char

```
typedef struct {          /* Callsign structure used in info */  
    char callsign[7]      ;  
    char ssid            ;  
} indicat                ;  
  
typedef struct {  
    indicat indic          ; /* 8   Callsign          */  
    indicat relai[8]       ; /* 64  Digis path        */  
    long lastmes           ; /* 4   Last L number     */  
    long nbcon             ; /* 4   Number of connexions */  
    long hcon              ; /* 4   Last connexion date */  
    long lastyap           ; /* 4   Last YN date      */  
    unsigned flags         ; /* 2   Flags             */
```

```

unsigned on_base ; /* 2   ON Base number          */
uchar   nbl      ; /* 1   Lines paging              */
uchar   lang     ; /* 1   Language                */
long    newbanner ; /* 4   Date of new banner (lang.NEW) */
ushort  download ; /* 2   Download size in KB      */
char    free[20] ; /* 20  Reserved                */
char    thema    ; /* 1   Current thema selection  */
char    nom[18]   ; /* 18  1st Name                 */
char    prenom[13]; /* 13  Christian name           */
char    adres[61] ; /* 61  Address                  */
char    ville[31] ; /* 31  City                     */
char    teld[13]  ; /* 13  home phone               */
char    telp[13]  ; /* 13  job phone                */
char    home[41]  ; /* 41  home BBS                 */
char    gra[7]    ; /* 7   Qth Locator              */
char    priv[13]  ; /* 13  PRIV directory           */
char    filtre[7] ; /* 7   LC choice filter         */
char    pass[13]  ; /* 13  Password                 */
char    zip[9]    ; /* 9   Zipcode                  */

} info ; /* 360 bytes = lenght of one record */

```

Dates are given as the number of seconds since january 1st, 1970 00:00.

In C language, all strings are ended with a NULL (00 hex) character.

16.16 YAPPLBL.DAT:

YAPPLBL.DAT (\FBB\SYSTEM).

This is a binary file which holds the informations of the FbbDOS files.
Do not edit or change this file, otherwise files of FbbDOS could be hidden or lost.

```

#define LABEL_FIC 80
#define LABEL_NOM 40
#define LABEL_OWEN 8

typedef struct {
    char  nomfic[LABEL_FIC] ; /* 80 Filename (virtual path of FBB) */
    char  label[LABEL_NOM]  ; /* 40 Label of the file              */
    char  owner[LABEL_OWEN] ; /* 8  Callsign of the owner (no ssid) */
    long  index             ; /* 4  Number of uploaded file (from 1) */
    long  date_creation     ; /* 4  date of the file creation        */
    char  free[24]          ; /* 24 unused (padded with NULL)      */
} Rlabel ; /* 160 bytes */

```

16.17 SATEL.DAT:

SATEL.DAT file (\FBB\SYSTEM\SAT).

This binary file holds the information on the satellite data base of the BBS. This file is very important and must not be edited or changed otherwise the list of the satellites could be lost. This file can only be opened for reading.

It is organized with records, each record containing information on a satellite. Never modify this file while the BBS is running.

Structure of one record (C language) :

```
typedef struct {  
    char    sat_name[18]; /* 18 Name of the satellite */  
    int     year          ; /* 2 Year of reference */  
    double  day           ; /* 8 Day of reference */  
    int     month         ; /* 2 Month of reference */  
    int     hour          ; /* 2 Hour of reference */  
    int     minute        ; /* 2 Minute of reference */  
    int     second        ; /* 2 Second of reference */  
    double  inclination   ; /* 8 Inclination */  
    double  raan          ; /* 8 R.A.A.N. */  
    double  excentricity; /* 8 Excentricity */  
    double  a_perigee     ; /* 8 Argument of perigee */  
    double  mean_anomaly; /* 8 Mean anomaly */  
    double  a             ; /* 8 must be 0.0 */  
    double  mean_motion   ; /* 8 mean motion */  
    double  dec_rate      ; /* 8 Decay rate */  
    long    revolution    ; /* 4 Epoch revolution */  
    double  frequency     ; /* 8 Frequency for doppler computing */  
    double  v             ; /* 8 must be 0.0 */  
    int     step          ; /* 2 Step of processing (minutes) */  
    long    last_upd      ; /* 4 Date of last update */  
    long    catalog       ; /* 4 NASA Catalog Number */  
    int     free[4]       ; /* 8 Unused */  
}  
} satel ; /* 138 bytes : lenght of one record */
```

In C language, all strings are ended with a NULL (00 hex) character.

16.18 WP.SYS:

WP.SYS file (\FBB\SYSTEM\WP).

The WP.SYS file is composed of records. Each record is 194 bytes long.

```
typedef struct {  
    char    callsign[7]    ; // Callsign of the record  
    char    name[13]      ; // Name of the user  
    char    free          ; // Unused  
    char    changed       ; // Flag (U, G, I) of modification  
    ushort  seen          ; // Number of updates  
    long    activ_date     ; // Date of active part  
    long    temp_date     ; // Date of temp part  
    char    activ_homebbs[41]; // Home BBS in active part  
    char    temp_homebbs[41]; // Home BBS in temp part  
    char    activ_zip[9]   ; // Zip code in active part  
    char    temp_zip[9]    ; // Zip code in temp part  
    char    activ_qth[31]  ; // Qth in active part  
    char    temp_qth[31]   ; // Qth in temp part  
}  
} Wps;
```

In C language, all strings are ended with a NULL (00 hex) character.

This file should NOT be modified while the BBS software is running !

16.19 MESS.WP:

MESS.WP file (\FBB\SYSTEM\WP).

The MESS.WP file is an ascii file composed of standard WP update lines. It has no size limit and will be sent as messages to the different routes defined for WP updates in INIT.SRV. The outgoing messages will be truncated to 5Kb sections if required.

This file is read at the end of the BBS start-up.

17 GLOSSARY:

This is an attempt to explain difficult/strange words from this documentation. Send all your suggestions for this page to LA6CU.

Autobin : A protocole for transfer of binary files via radio-ports.
Supported by several terminal-programs.

DLL : DLL is Dynamic Link Library. This is used in the windows environment and allows a program to attach a library which holds some code during the session or only during a phase of a program (like filters in FBB). This allows a very fast and efficient mechanism. Also DLL may be compact.

GPF : Windows error-message. Means that the BBS has performed an action that has caused a General Protection Failure.

VXD : VXD is more complex. I don't know exactly but they are something like drivers for Windows using 32 bit code.

YAPP : "Yet Another Packet Program". Originally a program by WA7MBL for simple transfer of binary files on packet radio. The protocol from his program is widely known as the YAPP-protocol.

Xforward : Another kind of compressed forward, implemented in FBB.

18 APPENDIX:

APPENDIX 1 : EDIT Commands
APPENDIX 2 : PK232 Host-Mode
APPENDIX 3 : G8BPQ interface
APPENDIX 4 : Kantronics KAM
APPENDIX 5 : DRSI-card
APPENDIX 6 : PROCESS NUMBERS AND HELP
APPENDIX 7 : RECORDING A MESSAGE
APPENDIX 8 : TRICKS AND TIPS
APPENDIX 9 : FBB FORWARD PROTOCOLE
APPENDIX 10: COMPRESSED FORWARDING
APPENDIX 11: FORMAT OF ACK MESSAGES
APPENDIX 12: REPLACEMENT CHARACTERS OR WILDCARDS
APPENDIX 13: PROGRAMMING TECHNIQS FOR SERVERS
APPENDIX 14: CONTENTS OF THE DIRECTORIES
APPENDIX 15: CONNECTION FILTERING
APPENDIX 16: MESSAGE FILTERING
APPENDIX 17: PG PROGRAMS DEVELOPMENT
APPENDIX 18: UNPROTO MESSAGE LISTS
APPENDIX 19: EXTENSIONS TO THE YAPP PROTOCOLE
APPENDIX 20: Telephone modem
APPENDIX 21: File filtering

APPENDIX 22: MAIL.IN (import messages directly)
APPENDIX 23: Installation-help for LinFBB
APPENDIX 24: TELNET installation for LinFBB and WinFBB
APPENDIX 25: TFPCX / TFPCR / TFKISS (Hostmode KISS-driver)
APPENDIX 26: EMS/XMS memory.
APPENDIX 27: Using FBB with PE1CHL NET and the BPQ Hostmode Emulator
APPENDIX 28: Conference-mode
APPENDIX 29: Gateway-mode
APPENDIX 30: TFWIN.DLL Kiss-interface for WinFBB32

18.1 APPENDIX 1: How to use the line-editor:

The FBBDOS offers a small line editor allowing the process of ASCII files. It can help the remote SysOp to maintain the system files of the software.

The commands are straightforward, and for better efficiency, they can be chained on a single line.

Here are the available commands:

? : Shows all commands that can be used in the editor.

A : Adds a line after the current line. The text for the new line shall follow the letter A. If more commands are to follow after the text, the text must be ended with a "/" before next command. If the sign "/" itself is to be used in the text, a "\" must be written just before the "/". If again the "\" is to be used inside the text as a valid character, it must be written twice ("\\"). The pointer will be at the start of the new line.

B : Go to start of file.

E : Go to end of file.

F : Search for the first occurrence of a character (or string of characters) in the file. The search starts at the current line. The string of characters must be ended with a "/" if more commands are chained (see the paragraph on the A-command).

I : Insert a line before the current line. Read about ending with a "/" in the paragraph on the A-command.

K : Delete the number of lines that are stated just before the letter K.
Example: 3K
If no number is used, then 1 line is deleted.

L : Move a number of lines from the current line. The number can be negative to move towards the start of the file.
Examples: 5L -10L

N : Toggle line-numbering on/off.

P : Show a number of lines from the current line. Example 8P. This will show 8 lines, starting with the current line.

R : Search and replace. Search for a string of characters and replace it with a new one. Example: RALL/ALLE will replace all ALL with ALLE from the current line to the end of the file.

S : Save file.

Q : Exit the editor without saving the file.

Examples::

"EDIT>B5L10P" sets pointer at start of file. Move 5 lines down, and show 10 lines from there.

"EDIT>B4L6K-2L10P" sets the pointer to the start of file. Move down 4 lines. Delete 6 lines. Move back 2 lines, and show 10 lines.

"EDIT>BFBonjour/K-2L5P" sets the pointer to the start of file, searches for "Bonjour", deletes the line containing "Bonjour", moves back 2 lines and shows 5 lines.

*

"EDIT>BFBonjour/IC'est une nouvelle ligne avec un \ / dedans/-1L3P" sets the pointer to start of file. Searches for "Bonjour", inserts the new line. The new line also contains the character "/". Goes back 1 line and shows 3 lines.

"EDIT>SQ" save file and quit the editor.

18.2 APPENDIX 2: PK-232 HOST MODE:

If you plan to use a PK-232 with this software, you must make some changes. First, check again PORT.SYS. You must have a P in "Type host mode". For example:

```
#
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1 7 1 1 230 4 1 10 30/60 UPYW 433.650
2 1 3 1 80 2 1 5 17/30 GPW 15/20m
#
```

Example of INITTNCx.SYS for PK-232:

```
UNMAIL v LA7QR means Unproto MAIL v LA7QR
RY10 " Retry 10
MN6 " Monitor 6
MC6 " Mcon 6
```

Example of MAINTx.SYS for PK-232:

```
UR1
CTBBS ($c) was shut down for service $d $T.
```

PK-232 host-mode commands (from F6AIW) :

8B 8BITCONV	AU AAB	AB ABAUD	AG ACHG	AA ACRDISP
AK ACRPACK	AT ACRRTTY	AE ADDRESS	AD ADELAY	AI ALFDISP
AP ALFPACK	AR ALFRTTY	AL ALIST	AM AMTOR	AC ARQ
AO ARQTM0	AS ASCII	AY ASPECT	AW AWLEN	AV AX25L2V2
AX AXDELAY	AH AXHANG	BA BAUDOT	BE BEACON	BI BITINV
BK BKONDEL	BT BTEXT	CL CANLINE	CP CANPAC	CX CASEDISP
CU CBELL	CC CCITT	CF CFROM	CB CHCALL	CD CHDOUBLE
CH CHSWITCH	CK CHECK	CQ CMDTIME	CM CMSG	CI CODE
CN COMMAND	CE CONMODE	CO CONNECT	CY CONPERM	CG CONSTAMP

CI	CPACTIME	CR	CRADD	CT	CTEXT	CW	CWID	DS	DAYSTAMP
DA	DAYTIME	DC	DCDCONN	DL	DELETE	DF	DFROM	DI	DISCONNE
DW	DWAIT	EA	EAS	EC	ECHO	ES	ESCAPE	FA	FAX
FN	FAXNEG	FE	FEC	FL	FLOW	FR	FRACK	FS	FSPEED
FU	FULLDUP	GR	GRAPHICS	HB	HBAUD	HD	HEADERLN	HI	HID
HO	HOST	HP	HPOLL	ID	ID	IL	ILFPACK	IO	IO
JU	JUSTIFY	KI	KISS	LR	LEFTRITE	LO	LOCK	MX	MAXFRAME
MB	MBX	MC	MCON	MD	MDIGI	MM	MEMORY	MI	MFILTER
MF	MFROM	MH	MHEARD	MN	MONITOR	MO	MORSE	MP	MSPEED
MR	MRPT	MS	MSTAMP	MT	MTO	MA	MYALIAS	ML	MYCALL
MG	MYSELCAL	MK	MYALTCAL	NE	NEWMODE	NO	NOMODE	NR	NUCR
NF	NULF	NU	NULLS	OK	OK	OP	OPMODE	PA	PACKET
PL	PACLEN	PT	PACTIME	PR	PARITY	PS	PASS	PX	PASSALL
PE	PERSIST	PP	PPERSIST	PC	PRCON	PF	PRFAX	PO	PROUT
PY	PRTYPE	RW	RAWHDL	RB	RBAUD	RC	RCVE	RE	RECEIVE
RX	RXREV	RD	REDISPLA	RL	RELINK	RS	RESET	RP	RESPTIME
RT	RESTART	RY	RETRY	RF	RFEC	SE	SELFEC	SP	SENDPAC
SI	SIGNAL	SL	SLOTTIME	SQ	SQUELCH	SR	SRXALL	ST	START
SO	STOP	TB	TBAUD	TC	TCLEAR	TM	TIME	TR	TRACE
TW	TRFLOW	TI	TRIES	TD	TXDELAY	TF	TXFLOW	TX	TXREV
UN	UNPROTO	UR	USERS	US	USOS	VH	VHF	WI	WIDESHFT
WO	WORDOUT	WR	WRU	XW	XFLOW	XM	XMIT	XO	XMITOK
XF	XOFF	XN	XON						

18.3 APPENDIX 3: G8BPQ driver:

G8BPQ-node (uses TNCs in KISS-mode, DRSI cards, etc...).

If you plan to use BPQ-node with this software, you must make some changes. Before you start FBB, you must load the BPQ-program.

From version 4.05 of BPQcode, separate ports can be declared. Level 2 connections issue to the MultCh in the same order as PORTs declared in the BPQCFG.TXT (G8BPQ distribution) file. First port corresponds to MultCh 0. Level 4 connections (from the network) always issue to MultCh 0.

In PORT.SYS you must use 2 in INTERFACE, and Q in type host mode. Version 4.05 or up is recommended. COM can be from 1 to 8, but will exclude existing COMs. Choosing 8 is a good solution to keep existing COMs. We recommend that you use COM 8.

Only the INITTNCx.SYS for the 1st port must exist. You must check that there is only one INITTNCx.SYS file for all BPQ-ports, otherwise you will have big problems! The parameters for other ports will be taken by default, and then no conflict should happen between ports. If you want to define the streams differently on each port, you can do it with the INITTNCx.SYS file, but be sure to give the right configuration. INITTNCx.SYS will look like this:

```
A1
R0
UMAIL
M1
N1
```

If paclen is equal to 0, then the default BPQ paclen of the port will be taken, otherwise the specified paclen will overwrite the default BPQ paclen.

If you have validated the gateway, you must configure ENABLE_LINKED=A in BPQCFG.TXT to give to the stream the callsign of the user.

In BPQCFG.TXT you will require to set the TNCPORT section up for COM 8.

As follows:-

```
TNCPORT
  COM=8
ENDPORT
```

Here is an example of PORT.SYS with BPQ-node and 2 TNCs:

```
#
#Ports TNCs
1      2
#
#
#Com Interface Address (Hex) Baud
8      2          0          4800
#
#
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      8      8      0      230  4      1      10      30/60  UQYW VHF
2      1      8      1      80   1      1      5      30/60  UQYW HF
#
```

Example of INITTNCx.SYS for BPQ 4.05 (Only ONE INITTNCx.SYS) :

```
A1      Number of application
R0      Returns to node (0=No,1=Yes)
UMAIL   Unproto MAIL
M1      Monitor (0=No,1=YES)
N1      Number of first stream
```

The parameters for other ports will be taken by default, For instance, as the first stream of the first port is 1, the first stream of the second port will be 9. (8 channels defined in PORT.SYS for port 1). This will be done automatically, and the second INITTNCx.SYS should not exist.

In FORWARD.SYS there are a few changes, here is a short example:

```
A LA2D
#
P B
#
C C SWITCH      Connect first BPQ switch.
C C 2 LA2D      Connect LA2D on port 2 of BPQ.
#
B LA2D
F LA2D
#
-----
```

For WinFBB:

If you are using WinFBB together with BPQ-code you have to copy the files

```
BPQCODE.386
BPQDLL.DLL
```

to the \WINDOWS\SYSTEM directory. These files are normally included in the BPQ programpack.

After this you starts BPQ in the same way as with DOSFBB, but before starting

WINDOWS. The best is to start BPQCODE from AUTOEXEC.BAT.

For LinFBB:

BPQ cannot be used with LinFBB....

18.4 APPENDIX 4: Kantronics KAM in host-mode:

If you plan to use a KAM with this software, you must make some changes.

For WinFBB:

Use interface 6 in PORT.SYS.

For DosFBB:

Before you start the BBS, you must load the RS232 driver. You must use ESSKAM or ESS (ESS must be version 1.10 or higher). Do not use another driver such as COMBIOS or MBBIOS.

Run it like this:

ESSKAM 1 (Com1, Address and IRQ = defaults)

ESSKAM 1 03F8 4 (Com1, Address=03F8Hex, IRQ4)

The KAM must be in the right baudrate, and in hostmode. You put it into hostmode with these commands:

INTF HOST

PERM

There is a special format for INITTNCx.SYS. For HF, use this:

```
UNPROTO MAIL/  
MONITOR ON/  
MCOM ON/  
MCON ON/  
MRESP ON/  
PID ON/  
RETRY 15/  
RESPTIM 0/  
FRACK 2/  
CHECK 30/  
HID OFF/
```

For VHF use this:

```
UNPROTO /MAIL  
MONITOR /ON  
MCOM /OFF  
MCON /ON  
MRESP /ON  
PID /ON  
RETRY /8  
CHECK /30  
HID /OFF
```

As you can see, the position of the "/" is very important for HF/VHF !

In PORT.SYS you must use 1 for VHF and 2 for HF in MultCh. You must also use K in "Type host-mode".

```
#  
# Same number of lines as number of TNCs.  
#  
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq  
1 4 1 1 230 4 1 10 30/60 UKYW 144.675
```

```

2      1      1      2      80      1      1      6      15/60      UKYW HF
#

```

18.5 APPENDIX 5: DRSI-card:

If you plan to use DRSI-card with this software, you must make some changes. Before you start FBB, you must load the DRSI-driver that comes with the DRSI-card. This must be TNCTSR-R or TNCTSR-L with a version-number higher or equal to 2.1.

The first DRSI-card (MultCh 0 and 1) must be at address 300.
The second DRSI-card (MultCh 2 and 3) must be at address 310.
The third DRSI-card (MultCh 4 and 5) must be at address 308.
The fourth DRSI-card (MultCh 6 and 7) must be at address 318.

Before installing cards in the PC, use the program CHKADDR to verify that the addresses corresponding to the cards have a value of FF. If there is a problem, check your configuration to find what peripheral is already using this address. Install just one card at the time, and verify each cards presence with CHKADDR.

When all cards are ok, configure the driver by means of the TAILORnn program (depending of the version). TNCTSR-S will be ok if you use only a few channels, while TNCTSR-L should be used for big configurations, up to 32 channels.

One problem: What DRSI calls DRSI-PORT, is what we call MultCh in FBB ! What DRSI calls PORT, is what FBB calls port or TNC ! We always use FBB-names here...

Each DRSI-card works like 2 radio-ports. The first DRSI-card is always MultCh 0 and 1, and the forth card is MultCh 6 and 7. If one radio-port is used on HF, that port must be the second port on the card.

All the DRSI-cards together, work like only one COM. And this COM-port does not need to exist in the PC. You may call that port COM 7 or COM 8 and keep free the already existing COMS.

Baud-rate does not matter, but should be set to a standard value, to avoid error-messages.

Here is an example of PORT.SYS with 2 DRSI-cards (4 radios) :

```

#
#Ports TNCs
1      4
#

#
#Com Interface Address (Hex) Baud
7      4      0      4800
#

#
#TNC NbCh Com MultCh Pacl Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      8      7      0      230 4      1      10      30/60      UDYW 433.650
2      1      7      1      80 2      1      5      12/30      GDW 15/20m
3      8      7      2      230 4      1      10      36/60      UDYW 433.650
4      8      7      3      230 4      1      5      10/30      GDW 145.300
#

```

If you are using DRSI-card, you also must change the INITTNC1.SYS. You will need only ONE file for all the ports.

Example of INITTNC1.SYS that initializes 4 ports:

```
MUIS
U0
p0 1 64 10 4 4 10 100 18000 30 2 0
p1 1 64 10 4 1 16 100 18000 30 1 0
p2 1 64 10 4 4 10 100 18000 30 2 0
p3 1 64 10 4 4 10 100 18000 30 2 0
```

The same goes for MAINT1.SYS, only 1 file is needed for all 4 ports, like this:

```
Y 1
U 1 BBS ($c) was shut down for service $d $T.
```

In FORWARD.SYS there is no changes. Use standard syntax.

Well, I think that should be all (for DRSI). Oh, by the way: DRSI cannot limit the number of connections per radio-port. But in FBB we still can have NbCh set to for example 1. When a station then connects to the BBS on that radio-port, and there already is one user connected there, the BBS will send a message to him: "Sorry, no more channels available" and disconnect him.

18.6 APPENDIX 6: Process numbers and help:

Process number and on-line help identification.

The various processing functions contained in the software are identified by three numbers displayed in the status banner. The first of those three numbers is the main process system (BBS, FBBDOS, Satellite Computation, etc...), the second number is the process function (in the BBS, list, message sending, etc..) and the third number is a sub-function (record of the message title, message, etc..)

A complete description of these numbers would be useless and time consuming. They are mainly used for debugging purpose. The first number is also used to identify the help block out of the x.HLP file. Upon receipt of the "?" or the "H", the software searches the x.HLP file for a line of corresponding to the language in use, and of the format @@ number word in which "number" stands for the current processing level, and "word" stands for the word following the command "?" or "H".

Example : you are inside FBBDOS, and you type in the command "? EDIT", the help block searched for must begin with the line:

```
@@ 9 EDIT
```

It may happen that a block matches several search keywords. It is enough to specify the various words separated by the character "|" (vertical bar), WITH NO SPACE.

```
@@ 9 EDIT|EDITEUR
```

List of the processing levels :

- 0 Connection.
- 2 Qra-Locator.
- 3 Statistics.
- 4 Information.
- 5 Nomenclature.
- 6 Satellite Orbital Computation.
- 9 FbbDos.
- 11 Telephone Modem
- 14 BBS.
- 15 Forward.
- 16 Gateway Sysop page.
- 17 YAPP.
- 18 Conference.

18.7 APPENDIX 7: Recording a message.

A message can be left by a user or within a forwarding connection. The recording mechanism is always the same.

The recording command is always like :

```
Sx desti @ bbs < expd $ ident + filename
```

Only recipient field is mandatory, all other fields are optional.

Appending a filename is a possibility reserved to the sysop. The name must be complete, including logic unit and complete path (C:\FBB\SYSTEM\TEST.TXT).

When receiving the command line, a first test checks if a route exists when a route has been specified, or if the message must be automatically routed when no route was specified.

The title of the message is then asked to the user.

If the title is missing, the message is canceled and the user returns to the main menu.

The text of the message is then asked to the user.

The software checks possible preamble lines. These lines give information on the previous BBS having routed this message. They all begin by R: on first column. The BBS callsign is given behind the @ character within the preamble line. All adjacent BBS mentioned in this preamble will be included in the "already forwarded" list, and will not be concerned by this message. This list specific to each message can be displayed with the \$ or FN command followed by the message number.

When receiving a /EX in first column, or a Ctrl Z, a message number is then assigned, The BID (or MID if private) and the list of adjacent BBS concerned by this message are created. All these information are sent to the user when acknowledging the message.

In case of disconnection before the /EX or Ctrl Z, the whole message will be lost, and the texts already stored are deleted.

All information about the message (sender, recipient, route, MID, title, etc...) are stored in the DIRMES.SYS file. The text of the message is stored in a sub-directory of the MAIL directory. The sub-directory is MAILn where n is the last digit of the message number. The name of the file corresponds to the message number 123 is M_000123.MES, the number is 6 digits wide, in this

case it is in the sub-directory MAIL3.

The message number uses a long integer (32 bits), the number boundary is very far (more than 4 billions !).

18.8: APPENDIX 8: Tricks and tips:

This rubric is yours, more than mine. I'll try to insert there all tricks you will tell me.

Only for DosFBB:
Using DesqView (c).

There is no particular problem when using DesqView. The minimum window size is 500 KB. You MUST use communication drivers, like ESS, COMBIOS or MBBIOS, as the software does not dispose of the whole process time.

Communication errors displaying.

An error counter can be displayed
in WinFBB: after the word "Resync" on the screen
in DosFBB: just right of the date, on the first line of the screen.
If you are using TNC2 with WA8DED software, these errors can be minor, but with PK232, error recovery is more difficult, and the system may reboot.

With a correct operation of your system, this counter will not appear, or exceptionally. If errors are displayed, they can result from :

- Using DOS 4.0 or 5.0 : The keyboard driver of these versions is very slow. You must use the ESS driver for RS232 (or COMBIOS).

- A too high baudrate, or RS232 defective cables. The baudrate can be selected down to 4800 Bds. It is not a good idea to go down 4800 Bds, as the performance of the software should be lower.

- Change the LM324 fitting out some TNC RS232 line drivers by a TL074 or TL084.

- HF detection in the TNC. Errors and resynchronizations will appear when the transmitter is running. There is no real cure, you must investigate.

You can also use communication drivers like ESS, COMBIOS or MBBIOS if you are not still using them.

Repeat the last message number.

The last message number displayed, read, killed, etc... can be utilized again with the # (pound) character. This short-cut allows as for an example to read a message after a list or to suppress it just after its reading.

Example :

```
F6FBB BBS > R 12351
The message is displayed ...
F6FBB BBS > K #
Message #12351 killed.
F6FBB BBS >
```

Problems with BPQ running as a Node ! (OZ1CBQ)

If you are using BPQ running as a node together with FBB you may have problems if your node is linked to other nodesystems as TheNet, TheNetNode, X-net, RMNC, FlexNet, etc.

BPQ seems to have a compatibility problem with these nodesystems causing some hangups during incoming connects via the nodesystem. The problem starts with lowering the available buffers and at last the system is hanging until a reboot has been done. Only cure for the time being is to run BPQ as a SWITCH only without nodeupdate (NODE=0 in BPQCFG.TXT) only allowing level_2 traffic.

This brings another problem:

When FBB runs together with BPQ running as a level_2 switch, it is only possible to have outgoing forward on two channels at the time. One channel using the callsign and one channel using the alias as first call in the forwardfile.

Example 1.st outgoing forward channel:

```
C C SWITCH
C C 1 OZ7BOX-2      ; Callsign of the BPQ-switch
C C ....           ; Other connects
```

Example 2.nd outgoing forward channel:

```
C C SWITCH
C C 1 ESBOX         ; Alias_name of the BPQ-switch
C C ....           ; Other connects
```

PMS forwarding:

In INIT.SRV there is a flag 128 in the FBB-type-forward section. If this flag is set, only PMS-users that have been declared with the F-flag in the EU-command (Edit User) will be able to up/download messages to/from the BBS. The same users should also have the U flag set (in EU-command) to enable them to send unproto resync requests to the BBS.

Display looks wrong (Only WinFBB):

If you get a wrong display of the windows of WinFBB, halt the software, delete windows\winfbb.ini and rerun the software.

18.9 APPENDIX 9: FBB forward protocol:

FBB software includes two forward protocols. The first one is standard with MBL/RLI protocol. The second one was developed to allow efficiency, particularly on long links where propagation time of data are long. The exchange of commands is reduced to a minimum, and not acknowledged to get time. The data transfer direction is changed every block of data, a block of data holding up to five messages. This uses the "pipeline" effect of long links (Nodes and digipeaters), and gain some time over short links (HF...).

FBB protocol is very simple in its principle. It is based on MID/BID usage. The identification is made by the F letter in the SID (system type identifier contained in square brackets). All command lines must start in first column with the 'F' character. All command lines are ended by a return (CR) character.

Suppose I call another BBS to forward some mail. When I connect another BBS using FBB protocol, I will receive the SID followed by a text and the prompt (">"). If the SID contains the F flag, I will send immediately my SID and the first proposal.

Proposals looks like :

```
FB P F6FBB FC1GHV FC1MVP 24657_F6FBB 1345
F> HH
```

FB : Identifies the type of the command (proposal)
P : Type of message (P = Private, B = Bulletin).
F6FBB : Sender (from field).
FC1GHV : BBS of recipient (@field).
FC1MVP : Recipient (to field).
24657_F6FBB : BID ou MID.
1345 : Size of message in bytes.
F> : End of proposal.
HH is optional. It is the checksum of the whole proposal in hexadecimal.

ALL the fields are necessary. This kind of command must hold seven fields. If a field is missing upon receiving, an error message will be send immediately followed by a disconnection.

A proposal can handle up to five FB command lines. If the total size of messages seems to be too important, the proposal can handle less lines. In FBB software, a parameter is defined in INIT.SRV file to tell the maximum size of the message block. It is set by default to 10KB.

Example of proposal :

```
FB P F6FBB FC1GHV.FFPC.FRA.EU FC1MVP 24657_F6FBB 1345
FB P FC1CDC F6ABJ F6AXV 24643_F6FBB 5346
FB B F6FBB FRA FBB 22_456_F6FBB 8548
F> HH
```

This proposal is limited to three FB lines, as the amount of messages overran the 10KB limit.

When receiving the proposal, the other BBS will reject, accept or defer the message. This command is made by a FS line :

```
FS -+=
```

This means :

- I don't want the first message (-).
- I need the second message (+).
- I defer the third message, as I'm still receiving it.

In the new version 1 of FBB protocol there are 3 more responses:
R, E or H:

"FS +R++" means that the second message is rejected. Only works with new version of the protocol.

The information is also written in the LOG like :

```
MJ B:Message_Bid V:Callsign_Rejecting
```


A warning message may be sent to the sending sysop when his message is rejected (see INIT.SRV for more info on warning messages).
The message is not marked as 'F', and still can be forwarded to another BBS

"FS +H++" means that the second message is held. Only works with new version of the protocol.
The information is also written in the LOG like :
MH B:Message_Bid V:Callsign_Rejecting
A warning message may be sent to the sending sysop when his message is held (see INIT.SRV for more info on warning messages).

"FS +E++" means that the second message has a format error. Only works with new version of the protocol.
A warning message may be sent to the sending sysop when his message proposal is wrong (see INIT.SRV for more info on warning messages).

It should interesting to defer a message if you are still receiving it on a other channel, or if you think that the size is to big, or for another reason. The message should be proposed again at the next connection.

FS line MUST have as many +,-,=, R, E, H signs as lines in the proposal.

When receiving the FS lines, I can send the block of messages. Each message is made with the title on the first line, the text, and a Ctrl Z in the last line. The is no blank line between the messages.

Title of 2nd message
Text of 2nd message
.....
^Z

When the other BBS has received all the asked messages, it acknowledges by sending its proposal, and the system is reversed.

If it has no message to send, it only sends a line :

FF

This line must not to be followed by a F>.

If the other hand has no message, it sends a line :

FQ

and asks for the disconnection.

Example :

```
F6FBB                                FC1GHV
-----
Connects FC1GHV

                                Connected

                                [FBB-5.11-FHM$]
                                Bienvenue a Poitiers, Jean-Paul.
                                >
```

[FBB-5.11-FHM\$] (F6FBB has the F flag in the SID)
FB P F6FBB FC1GHV.FFPC.FRA.EU FC1MVP 24657_F6FBB 1345

FB P FC1CDC F6ABJ F6AXV 24643_F6FBB 5346
FB B F6FBB FRA FBB 22_456_F6FBB 8548
F> HH

FS +-+ (accepts the 1st and the 3rd).

Title 1st message
Text 1st message
.....
^Z
Title 3rd message
Text 3rd message
.....
^Z

FB P FC1GHV F6FBB F6FBB 2734_FC1GHV 234
FB B FC1GHV F6FBB FC1CDC 2745_FC1GHV 3524
F> HH

FS -- (Don't need them, and send immediately the proposal).
FB P FC1CDC F6ABJ F6AXV 24754_F6FBB 345
F> HH

FS + (Accepts the message)

Title message
Text message
.....
^Z

FF (no more message)

FB B F6FBB TEST FRA 24654_F6FBB 145
F> HH

FS + (Accepts the message)

Title message
Text message
.....
^Z

FF (still no message)

FQ (No more message)

Disconnection of the link.

In this example, FBB protocol is used as the two BBS were identified by the F flag in the SID. If F6FBB had sent the SID [FBB-5.11-MH\$] when answering FC1GHV, the protocol should be the standard MBL/RLI.

All callsigns are only examples !

18.10 APPENDIX 10: Compressed forward.

FBB has 2 kinds of compressed forward:
Compressed forward FBB.
XFWD Compressed forward.

Extension to the protocol. Compressed forward FBB.

The protocol utilized for the transfer of ascii files compressed is an extension to the existing protocol. The compressed forward is validated by the presence of the letter B in the SID [FBB-5.12-BFHM\$]. The transfer of compressed files can only take place under FBB protocol. The presence of the letter B in the SID without the F letter will remain without effect.

The only difference as regard to the standard protocol is the submit line. It can specify the type of data contained in the compressed message. FA means that the transfer will be an ascii compressed message. FB means that the message will be a binary compressed file (this last possibility is not yet implemented in the version 5.12).

The submission of an ascii message will be in the form :
FA P FC1CDC F6ABJ F6AXV 24754_F6FBB 345

The submission of a binary file will be in the form :
FB P FC1CDC F6ABJ F6AXV 24754_F6FBB 345

The transferred data are of a specific format. The transfer will be done in binary mode. This last one is derived of the YAPP protocol which is very reliable. All transfer is made of a header, a block of data, an end of message and a checksum. Each transfer is equivalent to the transfer of one message of the standard protocol and shall not be followed by a control Z, the end of file specifier is defined in another way.

Format of header for an ascii compressed message (submission FA) :

<SOH> 1 byte = 01 hex
Length of the header 1 byte = Length from the title,
including the two <NUL> characters.
Title of the message 1 to 80 bytes
<NUL> 1 byte = 00 hex
Offset 1 to 6 bytes
<NUL> 1 byte = 00 hex

Format of header for a binary compressed file (submission FB) :

<SOH> 1 byte = 01 hex
Length of the header 1 byte = Length from the filename,
including the two <NUL> characters.
Name of the file 1 to 80 bytes
<NUL> 1 byte = 00 hex
Offset 1 to 6 bytes
<NUL> 1 byte = 00 hex

As to follow the french regulation, the title of the message or the file name are transmitted in ascii, not compressed.

The offset is also transmitted in ascii and specifies the offset at which the data should be inserted in the file (in case of a fragmented file). In the version 5.12, this parameter is not utilized and is always equal to zero.

A data block contains from one to 256 bytes. It begins by two bytes which specify the format.

Data block format :

<STX> 1 byte = 02 hex
Number of data 1 byte = 00 to ff hex. (00 if length = 256 bytes).

Data bytes 1 to 256 bytes

The last data block is followed by the end of file specifier and the checksum.

End of file specifier format :

<EOT> 1 byte = 04 hex

Checksum 1 byte = 00 a ff hex

The checksum is equal to the sum of all the data bytes of the transmitted file, modulo 256 (8 bits) and then two's complemented.

The checking of the checksum is very simple :

The sum of the data from the file and the checksum received modulo 256 (anded with FF) shall be equal to zero.

In case of a checksum error, the message or the file is not taken to account and the system issues a disconnect request after having sent the comment :

*** Checksum error

Extension to the protocol. XFWD compressed forward.

X forwarding Protocol is implemented.

XForwarding now supports re-routing and swapping.

Binary forwarding via telephone modem (FBB or XFWD)

18.11 APPENDIX 11: Format of ACK-messages:

The ACK messages on receiving have a simple and compact format. The aim is to have a message as short as possible in order to avoid an unnecessary usage of the network.

The title of the message is the title of the original message with a leading "ACK:". Example :

ACK:Title of the original message.

These ACK messages are true messages strictly speaking. They carry the origin, the destination, the route and the MID but they are of a particular type, the type A (private are of type P, bulletins of type B, etc...). This difference allows the routing of these messages without the lines "R:". This is done again with the aim of avoiding an excessive load by data which are of no use in this case.

To keep the compatibility with the existing forwarding protocol, the type of these messages is changed to P (private) if the receiving BBS of the forwarding does not know the type of ACK messages (specified in the SID [FBB-5.12-ABFHM\$] by the letter A). In this case, the ack message will continue on its route as a private message.

The ACK messages are of the following form :

ACK:Message test. <-- Title of message

Msg FD1CDC@F6FBB - 22-dec 17:28z <-- Text of message

It tells that the message that you had sent to FD1CDC at F6FBB and whose

title is "Message test" has been received in the BBS F6FBB on 22 dec at 12:28 GMT.

18.12 APPENDIX 12: Wildcards:

Replacement characters or wildcards.

Most of the search commands or list commands and some configuration files as well, accept replacement characters or wildcards.

Character Replaces

@ a letter
? an alphanum character (letter or number)
= a printable character
a numeral character or the # character
* a string of printable characters.
& a dot followed by printable characters. (equiv to .*)

18.13 APPENDIX 13: Programming servers:

Programming technics for servers.

The servers are exec programs (.COM or .EXE). They are compact and fast. They will work as the function of the messages which are addressed to them.

They should be compact because the available memory to run their application is limited (check the information Ok:nnnn in the status window). They should be fast because they are executed in the MsDos environment which is not multi-task.

The programming language can be of any kind provided that it could be compiled and that it is able to read parameters which are given appended in the command line.

I wrote three servers in TurboC but I have no equivalent in TurboPascal or in TurboBasic, since I usually don't write in these languages. The working principle remains always the same whatever language is utilized.

The program is called with the following manner from the MsDos (Example for the REQDIR.COM command) :

```
C> REQDIR.COM TEMP.$$$
```

TEMP.\$\$\$ is the name of the file in which the message addressed to REQDIR is located. It is necessary to read the name of this file in the command line, as the one can change from one call to another.

The file TEMP.\$\$\$ contains the message with the following format :

```
SP REQDIR < F6FBB
Title of message
Text of message line 1
Text of message line 2
...
Text of message last line.
/EX
```

The server should then eventually work as a function of the contents of this message.

The server can read and make use of the configuration file of the BBS software (in particular INIT.SRV) to execute its process.

If the server generates a return message, it should be APPENDED to the incoming mail file to the BBS. The name of this file can be found in INIT.SRV. Take care : it is necessary to open the incoming mail file in APPEND as to add the answer at the end of the file. If it is not done this way, the messages which could be waiting in this file are destroyed.

The incoming mail file is tested each and every minute, except in the case of the usage of a service, where it is tested right after.

The format of the messages in the incoming mail file is identical to the format of the file given to the server. Several messages can be written sequentially in the file. There should not be blank lines or separations between the messages. The routing fields (@ field), and the originator (< field) should mandatory be specified. The originator field is the callsign of the BBS which is taken from the INIT.SRV file.

Example of server REQFIL written in C language.

```
/*
 * REQFIL.C Server example.
 *
 * This server is called with a command line like this :
 *
 * REQFIL.COM FILE
 *
 * FILE is the filename of the message to be answered.
 *
 * This server answers to a message like this :
 *
 * SP REQFIL < FC1EBN
 * TEST.TXT @ F6ABJ
 * Text is not necessary
 * /EX
 *
 * by a message like this
 *
 * # <- This is a local message
 * SP FC1EBN @ F6ABJ < F6FBB <- command line
 * Req File : TEST.TXT <- subject
 * Contents of the file <- text
 * etc.....
 * /EX <- end of text (must be in 1st column)
 *
 * Appent to mail in bbs file.
 *
 * The server receives from FBB software 1 argument :
 *
 * argv[1] = Name of the file including the message received from
 * FBB software.
 *
 * =====
 * The server must APPEND its answer to MAIL.IN
 * file to avoid destroying existing mail.
 * =====
```

```

*
* As this server opens the INIT.SRV file, it must be in the same
* directory.
*
*/

#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>

/* Offsets of parameters from INIT.SRV */

#define BBS_CALL 1
#define USER_DIR 8
#define MAIL_IN 14

main(int argc, char **argv)
{
#define LINE 80
    int end = 0;
    int index = 0;
    FILE *fptr;
    char buffer[LINE];
    char sender[LINE];
    char route[LINE];
    char file[LINE];
    char bbs_call[LINE];
    char base_dir[LINE];
    char mail_in[LINE];

    if (argc != 2) exit(1);                /* Check the number of arguments */

    /* The first task is to open and then read the message */

    fptr = fopen(argv[1], "rt") ;          /* Open the received message */
    if (fptr == NULL) exit(1);

    fgets(buffer, LINE, fptr);              /* Read the command line */
    sscanf(buffer, "%*s %*s %*s %s\n", sender);

    *file = *route = '\0';
    fgets(buffer, LINE, fptr);              /* Read the subject */
    strupr(buffer);                        /* Capitalize */

    /* Scan dir and route */
    sscanf(buffer, "%[^@\n]%[^^\n]", file, route);

    fclose(fptr);                          /* All needed is read in the message */

    /* We must get some informations from the INIT.SRV file */

    fptr = fopen("INIT.SRV", "rt"); /* Open the file */
    if (fptr == NULL) exit(1);

    /* Scan the file to get the requested lines. */
    while (!end) {
        fgets(buffer, LINE, fptr) ;
        if (*buffer == '#') continue; /* Comments ignored */

        switch (++index) {

            case BBS_CALL:
                sscanf(buffer, "%[0-9A-Za-z]", bbs_call);

```

```

        break; /* Callsign */

case USER_DIR:
    sscanf(buffer,"%s\n", base_dir);
    break; /* Users directory */

case MAIL_IN :
    sscanf(buffer,"%s\n", mail_in);
    end = 1; /* Mail in file */
    break;

    }
}

fclose(fp_ptr);

/* Append the answer to mail in file */
/* Mail in file is opened in append text mode */

if (fp_ptr = fopen(mail_in, "at")) {

    /* Tell that this is a message from this BBS */

    fprintf(fp_ptr, "#\n");

    /* Send command line */
    fprintf(fp_ptr, "SP %s %s < %s\n",
sender, route, bbs_call);

    /* Send subject and requested file */
    send_file(fp_ptr, base_dir, file);

    /* Send end of message */
    fprintf(fp_ptr, "/EX\n");

    /* That's all ! */
    fclose(fp_ptr);
}
exit(0); /* Tell BBS all is correct */
}

int points(char *ptr) /* Looks for a ".." sequence in the path */
{
    while (*ptr) {
        if ((*ptr == '.') && (*(ptr+1) == '.')) return(1);
        ++ptr;
    }
    return(0); /* ".." not found ! */
}

send_file(FILE *fp_ptr, char *base_dir, char *filename)
{
#define BUF_SIZE 1000
    int fd;
    int nb;
    char path[256];
    char buffer[BUF_SIZE];
    char last_char;

    sprintf(path, "%s%s", base_dir, filename); /* Complete path */

    fprintf(fp_ptr, "ReqFil 1.1 : %s\n", filename); /* Subject */

```



```

if ((!points(path)) && ((fd = open(path, O_RDONLY | O_TEXT)) != -1)) {
    while (nb = read(fd, buffer, BUF_SIZE)) {
        fwrite(buffer, nb, 1, fptr);
        last_char = buffer[nb-1];
    }
    close(fd);

    /* Be sure /EX will be in first column */
    if (last_char != '\n') fputc('\n', fptr);
}
else fprintf(fptr, "File not found !\n");
}

```

18.14 APPENDIX 14: Contents of the directories:

These files and directories are mandatory. Other files may reside but will be created by the system or by the sysop.

Directory FBB :

[BIN]		BAT, COM and EXE files.
[BINMAIL]		Compressed messages.
[DOC]		Documentation-files for the BBS.
[DOCS]		Files used for DOCS command.
[EXAMPLES.700]		Backup-copy of system-files for this version.
[MAIL]		Ascii messages.
[OLDMAIL]		Archives of messages.
[PG]		COM and EXE files for PG command.
[STATS]		Statistics for the BBS.
[SYSTEM]		System files.
[USERS]		Area for users to upload/download files.
[UTIL]		Different utilities for the BBS.
DEFAULT	MNU	Configuration-file for FBBSETUP-program. DosFBB only.
DEFAULT	OLH	Configuration-file for FBBSETUP-program. DosFBB only.
EPURMESS	INI	Init file for EPURMESS.
FBBSETUP	EXE	Configure INIT.SRV and PORT.SYS. DosFBB only.
INIT	SRV	Init file for the BBS.
MODLABEL	EXE	Program to update old YAPPLBL.SYS from older FBB-versions.
TF18	.ROM	Binary ROM-files for making EPROMs for TNCs for use in HOST-mode with this BBS-software
	etc	

Directory BIN :

APPEL	BAT	Batch file running the software. For DosFBB only.
BC453RTL	DLL	Part of main-program, WinFBB only.
BIDS47	DLL	Part of main-program, WinFBB only.
CLEANUP	COM	Program to clean MAIL and BINMAIL directory.
CLR_USER	COM	Resets a users quota for downloaded files
CONNECT	WAV	Sound-file, sounds when a user connects. WinFBB only.
CUT	COM	Copy/paste utility.
DSZ	BAT	DSZ software for up/download of binary files. Only WinFBB.
DSZ	COM	DSZ software for up/download of binary files. Only WinFBB.
DSZ	PIF	DSZ software for up/download of binary files. Only WinFBB.
EPURMESS	COM	Message processing.
EPURWP	COM	White Pages processing.
ESS	COM	RS232 driver. Only DosFBB.
ESSKAM	COM	KAM RS232 driver. Only DosFBB.
FBBIOS	COM	MODEM driver. Only DosFBB.

FBB_SVC	EXE	Program for starting servers. For WinFBB only.
FBB_SVC	PIF	Program for starting servers. For WinFBB only.
FV	COM	FV shareware used for LIST command in FbbDos.
INSTALL	EXE	Program for installing the BBS-software. For WinFBB only.
INSTWFBB	EXE	Configure INIT.SRV and PORT.SYS. Only for WinFBB.
ISDAY	COM	Utility for triggering certain actions as specific day.
MAINTINF	COM	INF.SYS maintenance program.
MULTI	EXE	For sending multiple copies of personal messages.
NEWDOC	COM	NEWDOC server.
OWL253A	DLL	Part of main-program, WinFBB only.
REQDIR	COM	REQDIR server.
REQFIL	COM	REQFIL server.
RUNFBB	BAT	Batch file running the software. For WinFBB only.
RUNFBB	EXE	Batch file running the software. For WinFBB only.
RUNFBB	PIF	Batch file running the software. For WinFBB only.
SATUPDAT	EXE	Automatic update of satellite database.
SERV	EXE	FBB software EXE file. For DosFBB only.
SETUSER	COM	set a user owner of a FbbDos file.
SLEEP	EXE	waits for some seconds.
SYSCALL	WAV	Sound-file, sounds when a user wants to "talk". WinFBB only.
WFBB	EXE	FBB software EXE file. For WinFBB only.
WFBB	HLP	FBB software Help-file. For WinFBB only.

Directory BINMAIL :

[MAIL0]	[MAIL1]	[MAIL2]	[MAIL3]	[MAIL4]
[MAIL5]	[MAIL6]	[MAIL7]	[MAIL8]	[MAIL9]

Directory DOC :

CHANGES	DOC	Shows all changes in the software during development.
FBBMUX	DOC	Doc for the Port Multiplexer
MULTI	DOC	Doc for sending multiple copies of personal messages.
WA8DED21	DOC	DED-host documentation.
DSZ	DOC	Doc for the DSZ-program. Only for WinFBB.
FV	DOC	Doc for the FV-program, used with WHERE command.

Directory DOCS :

This is the section for easy-to-use documentations. Numerical commands for reading the docs.

Directory EXAMPLES.700 :

Backup of most of the new text-files and configuration-files for this version of the software.

Directory MAIL :

[MAIL0]	[MAIL1]	[MAIL2]	[MAIL3]	[MAIL4]
[MAIL5]	[MAIL6]	[MAIL7]	[MAIL8]	[MAIL9]

Directory OLDMAIL :

Directory for backup of messages (bulletins and personal messages).

Directory PG :

Directory for special-made programs that the user can run. The filename is the command-name.

Directory STATS :

Directory for statistical programs for the BBS.

LISEZMOI	DOC
LOG	DOC
LOGCALL	EXE
LOGFBB	CNF
LOGSTATF	EXE
LOGSTATU	EXE

Directory SYSTEM :

[FWD]		Directory for FORWARD.SYS include-files.
[LOG]		Directory for FBBLOG.nn files.
[LANG]		Directory for .TXT .HLP .ENT language files.
[SAT]		Directory for satellite files
[WP]		Directory for White pages database
BEACONn	SYS	Beacon file.
BBS	SYS	BBS callsigns
CRON	SYS	Hour work.
INITTNCn	SYS	Initialization of TNC.
INITTNC	DED	Initialization of TNC, example for DED-host.
INITTNC	PK	Initialization of TNC, example for PK-232.
FORWARD	SYS	Forwarding description.
LANGUE	SYS	Language description
MAINTn	SYS	Housekeeping text.
MEMO	SYS	CtrlF0 to CtrlF9 aliases
MULTI	TXT	Configuration of languages for MULTI-server.
PASSWD	SYS	Password file.
PORT	SYS	Port description.
PROTECT	SYS	Directories protection.
REDIST	SYS	Configuration-file for the REDIST-server.
REJECT	SYS	Messages reject.
SWAPP	SYS	Swap file.
THEMES	SYS	Configuration-file for the thematic-zone.

Directory USERS :

Directory for the users to upload their own files.

Director USERS\YAPP :

Directory for users upload with YAPP-protocol and YU/YD commands.

Directory UTIL :

Directory for special utility-programs that may be used with this BBS:

ICOM	DLL	Program for controlling ICOM-radios.
ICOM	COM	Program for controlling ICOM-radios.
M_FILTER	DLL	Message-filtering example-program.
PROGTX	DLL	Program for sending commands to radios (QSY etc).

18.15 APPENDIX 15: Connections filtering:

FBB software allows filtering on connection. Filtering is not done by the BBS software but by external programs developed by users.

When the BBS starts, the C_FILTER does not really need to be there. But at the first connect it must exist. If it does not exist at THAT time, it will not be checked again. So if a C_FILTER is added after that, the BBS must be rebooted for the filter to take effect.

Connection filter may be interactive and allows to incorporate some features like dedicated information for predefined callsigns, password filtering, etc...

The C_FILTER program must be found by the PATH of MsDos. Its extension can be COM or EXE, and it must be little and fast as multitasking is stopped during the activity of this program. If this program is not found, it will not be called until the BBS is rebooted.

The C_FILTER may also be created as a DLL. Both for WinFBB and DosFBB (!!). The filter must be installed in the path (\FBB\BIN) of Dos.

When receiving the connection, the C_FILTER program (if found) is called with some arguments including a level number. This number is incremented each time the program is called in the same connection session. The first time the level number will be 0.

The line arguments given to the C_FILTER program are :

- Callsign (format as F6FBB-8).
- Level number (0 is the first time, up to 99).
- Flags of the user (binary number as user's mask of INIT.SRV).
- New : Flag indicating if the user is unknown in the BBS database.
- Record number of the user in INF.SYS.
- New: one more parameter before the optional text. It indicates the port where was connected the user.
- Received data (in one argument).

The C_FILTER program ends with an exit value. This value is very important and tells the BBS what to do :

Return value (for C_FILTER):

- 0 : Connection is accepted
- 1 : C_FILTER will called again, level is incremented
- 2 : Connection is refused, user is disconnected
- 3 : Connection is accepted, but in read-only mode
- 4 : Connection is accepted, but messages will be hold.
- 100 and up
: C_FILTER will called again, next level is equal to the return value.

Example of C_FILTERs. First C_FILTER.EXE, next C_FILTER.DLL:

Example C_FILTER.EXE:

```
/*  
 * C_FILTER.C  
 *  
 * Connection filter for FBB BBS software. (C) F6FBB 1991.
```

```

*
*/

#include <stdio.h>

/*
* Connexion filter called for each connection.
* All datas sent to stdout will be sent to the user.
*
* Filter is called with some arguments on the command line :
* C_FILTER CallSign Level Flags New Record ReceivedData....
*
* The return value tells the BBS if C_FILTER must be called again or not
* 0 if the BBS can go on,
* 1 if the C_FILTER must be called again
* 2 if the user must be disconnected.
*
* Callsign is in the FORM CALLSIGN-SSID (F6FBB-0).
*
* The first time C_FILTER is called Level=0, and then will be incremented
* each time it is called for the same connection.
*
* Flags give the flags of the user (MGPBSXLE) in a bit field. as defined
* in the INIT.SRV user's mask. (0x80=Modem, 0x40=Guest, etc...)
*
* If New=1, then this is the first connection of the user on the BBS.
* Record is the record number in the INF.SYS file.
*
* All other arguments are the words sent by the user
* (password for instance).
*
* The number of arguments is variable and depends of the number of words
* in the answer of the user.
*
*/

/*
* This is only a little example to test the system. It will be called
* four times and will give the list of arguments.
*
* The fourth time, the hand will be given back to the BBS.
*/

main(int ac, char **av)
{
    int i;
    int level = atoi(av[2]);           /* Get level from argument list */
                                      /* and transform it to integer */

    if (level == 0) {                 /* Is level equal to 0 ? */
        printf("Connection line :\n"); /* This is the first call */
        for (i = 0 ; i < ac ; i++)    /* List line arguments */
            printf("%s ", av[i]);
        putchar('\n');
        return(1);                    /* C_FILTER must be called again */
    }
    else {
        printf("Following line :\n"); /* These are other lines */
        for (i = 0 ; i < ac ; i++)    /* List line arguments */
            printf("%s ", av[i]);
        putchar('\n');
        if (level == 4)               /* Is it the last time ? */

```

```

        return(0);                                /* Yes, go on BBS          */
    else
        return(1);                                /* No, call once more C_FILTER */
}
}

```

Next example, C_FILTER.DLL:

```

#define STRICT
#include <stdio.h>
#include <windows.h>

/*
 *   Code (C) F6FBB 1995-1996
 *
 *   C_FILTER example
 *
 *   svc_main is the only exported function. It must exist as WFBB will
 *   look for it. All filters are in the same format. If not NULL,
 *   r_buf allow to give a direct text back to the BBS
 *   The size of the buffer is given by the parameter len.
 *
 *   Answers may also go to stdout (slower...).
 *
 *   This code is only an example and was not fully tested. It could
 *   give problems as I am not an expert !
 *
 *   The used compiler is a Borland C++ 4.5
 *
 *   The DLL is not linked with the code, but checked and loaded when
 *   needed by WFBB.
 *
 *   This is an example of the C_FILTER.DEF file :
 *
 *   LIBRARY          C_FILTER
 *   DESCRIPTION      'C_FILTER DLL for FBB'
 *   EXETYPE          WINDOWS
 *   CODE             PRELOAD MOVEABLE DISCARDABLE
 *   DATA            PRELOAD MOVEABLE SINGLE
 *   HEAPSIZ          1024
 *
 *
 *   Parameters are (for C_FILTER):
 *   ac[0] : The name of the filter (C_FILTER)
 *   ac[1] : The callsign and SSID
 *   ac[2] : Level number (from 0)
 *   ac[3] : Flags
 *   ac[4] : Boolean for new user
 *   ac[5] : Recordnumber in INF.SYS
 *   ac[6] : Port number
 *   ac[7] : Optionnal Command received
 *   r_buf : The buffer to put an answer
 *   len   : The size of the answer buffer
 *
 *   Return value (for C_FILTER):
 *   0 : Connection is accepted
 *   1 : C_FILTER will called again, level is incremented
 *   2 : Connection is refused, user is disconnected
 *   3 : Connection is accepted, but in read-only mode
 *   4 : Connection is accepted, but messages will be hold.
 * 100 and up
 *   : C_FILTER will called again, next level is equal to the return value.

```

```

*
*/

int _export FAR PASCAL svc_main (int ac, char FAR ** av, char FAR * r_buf, int
len)
{
    if (len > 20)
    {
        sprintf (r_buf, "Inside the C_FILTER\r");
    }
    return (0);
}

```

18.16 APPENDIX 16: Message filtering:

FBB software allows filtering messages. Filtering is not done by the BBS software but by external programs developed by users.

When the BBS starts, the M_FILTER does not really need to be there. But at the first message, it must exist. If it does not exist at THAT time, it will not be checked again. So if a M_FILTER is added after that, the BBS must be rebooted for the filter to take effect.

M_FILTER may be interactive and allows to incorporate some features like dedicated information for predefined callsigns, password filtering, etc...

I did not develop such programs, but this is an open door to many applications.

The M_FILTER program must be found by the PATH of MsDos. Its extension can be COM or EXE, and it must be little and fast as multitasking is stopped during the activity of this program. If this program is not found, it will not be called until the BBS is rebooted.

The M_FILTER may also be created as a DLL. Both for WinFBB and DosFBB (!!). The filter must be installed in the path (\FBB\BIN) of Dos.

The message filter is called (if found) each time a message is ready to be recorded (when Ctrl Z or /EX is received). The decision to validate or not the message is function of the exit value of the M_FILTER program.

The M_FILTER program (if found) is called with some arguments including a level number. This number is incremented each time the program is called in the same connection session. The first time the level number will be 0.

The line arguments given to the M_FILTER program are :

- File name including the text of the message.
- Type of the message (P, B, T).
- Sender.
- "To" field.
- Record number of DIRMES.SYS file.

The M_FILTER program ends with an exit value. This value is very important and tells the BBS what to do :

- 0 : Message is recorded.
- 1 : Message is killed (status = K).
- 2 : Message is archived (status = A).
- 3 : Message is held (status = H).

```

/*
 * M_FILTER.C
 *
 * The message filter MUST be named M_FILTER (COM or EXE).
 *
 * This example only writes its call arguments in the TEST.MES file.
 *
 * It is called with 5 arguments :
 *   File name of the message.
 *   Type .
 *   Sender.
 *   To.
 *   Number of the record in the DIRMES.SYS file.
 *
 * If it returns 0 : The message is accepted.
 *               1 : The message is killed (status K).
 *               2 : The message is archived (status A).
 */

```

```

#include <stdio.h>
main(argc, argv)
int  argc;
char **argv;
{
    int  i;
    FILE * fptr = fopen("TEST.MES", "at");

    for (i = 0 ; i < argc ; fprintf(fptr, "%s ", argv[i++]));
    fputc('\n', fptr);

    fclose(fptr);

    return(0);
}

```

18.17 APPENDIX 17: PG programs:

PG programs development.

PG programs are in the PG subdirectory. They are small programs allowing interactivity with the user.

COM or EXE programs can be called.

PG programs must be little as the amount of memory is limited and fast because the multitasking is stopped during its activity.

To run a PG program and start a session, the user must type the command PG followed by the name of the program. The PG command alone gives the content of the PG subdirectory. The PG program is particularly developed for FBB software but can be an interface to a standard program.

Each time a complete line (up to the return character) is received, the PG program is called with some arguments including a level number. This number is incremented each time the program is called in the same PG session. The first time the level number will be 0.

The line arguments given to the PG program are :

- Callsign (format as F6FBB-8).
- Level number (0 is the first time, up to 99).
- Flags of the user (binary number as user's mask of INIT.SRV).
- Record number of the user in INF.SYS.
- Received data (each word is a new argument).

The PG program ends with an exit value. This value is very important and tells the BBS what to do :

- 0 : end of session and return to the BBS menu.
- 1 : the program will be called again and the level number is incremented.
- 2 : the user will be disconnected.
- 3 : the receive data will be sent as a BBS command and return to BBS.
- 4 : the receive data will be sent as a BBS command, level incremented.
- 5 : the program will be called again, but the level is not incremented.

The data sent by the PG program to the standard output will be sent to the user. This allows a real interactivity between the user and the PG program.

Here is an example of a small program :

```
/*
 * TST_PG.C
 *
 * Little test program of "PG" command for FBB BBS software.
 *
 * (C) F6FBB 1991.
 *
 * FBB software 5.14 and up.
 *
 * This program echoes to the user what he types
 * or executes a BBS command preceded by "CMD"
 * until "BYE" is received
 */

#include <stdio.h>

main(int argc, char **argv)
{
    int i;
    int level = atoi(argv[2]);          /* Get level from argument list */

                                        /* and transform it to integer */
    if (level == 0) {                  /* Is level equal to 0 ? */
                                        /* This is the first call */
        printf("Hello %s, type BYE when you want to stop !\n", argv[1]);
        return(1);                    /* program will be called again */
    }
    else {
       strupr(argv[5]);               /* Capitalise the first word */
        if (strcmp(argv[5], "BYE") == 0) { /* is BYE received ? */
            printf("Ok, bye-bye\n");
            return(0);                 /* Yes, go on BBS */
        }
        else if (strcmp(argv[5], "CMD") == 0) { /* is CMD received ? */
            for (i = 6 ; i < argc ; i++) /* List line arguments */
                printf("%s ", argv[i]); /* sent by user */
            putchar('\n');
            for (i = 6 ; i < argc ; i++) /* List line arguments */
                printf("%s ", argv[i]); /* sent by user */
            putchar('\n');
            return(4);                 /* Yes, send command */
        }
    }
}
```

```

    }
    else {
        printf("You told me : ");          /* These are other lines          */
        for (i = 5 ; i < argc ; i++)      /* List line arguments          */
            printf("%s ", argv[i]);        /* sent by user                  */
        putchar('\n');
        return(1);                        /* No, call again program        */
    }
}
}
}

```

18.18 APPENDIX 18: Unproto message lists:

FBB software allows sending unproto lists of message. This is validated separately on each port (letter L in PORT.SYS). Unproto address is FBB with the following header :

fm F6FBB-1 to FBB ctl UI

An unproto list line is sent on every validated port each time a message is recorded. The line is in the form :

12345 B 2053 TEST@ALL F6FBB 920325 This is the subject

If a message number is missing or does not exist the line will be :

12346 #

This allows a system listening to the UI packets on a frequency to create a list identical to the one of the BBS, and then the user will not have to connect the BBS to know the list of messages and bulletins.

A control can be done on the number of the messages to check if a line is missing.

If the remote system receives a new line, and a line is missing, it only has to send an unproto frame addressed to the BBS callsign like this :

fm FC1EBN-3 to F6FBB-1 ctl UI
? 00002EE00E

This will be taken in account only if the user has the U flag validated (EU command).

If the user has not his flag validated in the BBS, he will receive a line like :

fm F6FBB-1 to FBB ctl UI
12200 / callsign

In this case, the the remote software MUST stop asking unprotoes.

The first 8 digits are the hexadecimal number of the requested start of the list (here 00002EE0 -> 12000) and the last two digits are the sum of the four bytes anded with FF (0E).

The BBS will then starts sending lines from the requested number up to the last message number.

If the number requested seems to be too far from the current line, the BBS can readjust the request of "callsign" while sending :

```

fm F6FBB-1 to FBB ctl UI
12200 ! CALLSIGN
12201 B 2040     TEST@FRA     F6FBB  920325 This is a bulletin
12202 #
12203 P 206      F6ABJ@F6ABJ  F6FBB  920325 Hello Remy.
etc...

```

and then starts sending lines from 12201. The remote system must change its base number to 12201.

If the number requested is greater than the last message received in the BBS, the BBS will send a line like :

```
12300 !!
```

This indicates that the list in the remote system is up to date. The last received message in the BBS is 12300.

The remote system can also connect the BBS and ask for messages in binary compressed mode using the following sequence :

BBS	Remote system

	Connection request
Connection.	
[FBB-5.14-ABFHM\$]	
Welcome in Toulouse, Gerard.	
F6FBB BBS>	
	[TPK-1.80-\$]
1>	
	F< #Msg
Binary compressed message #Msg	
is sent using format described	
in appendix 7	
1>	
	Disconnect.

From TPK version 1.80 (packet communication program developed by FC1EBN), this protocol is implemented.

18.19 APPENDIX 19: Extensions to YAPP-protocol:

These extensions are used in TPK 1.65a (and up) and FBB 5.14 and up.

HEADER extension.

The header now carries the DATE and TIME of the file being transmitted.
[SOH] [Len] [Filename] [NUL] [File Size] [NUL] [Date] [Time] [NUL]

The Date and Time are the values reported by DOS, coded in 4 hexadecimal digits and are sent in ASCII (8 characters).

The receiver has the choice of using either extended Yapp with checksum or normal Yapp.

The normal Yapp reply is RF, as before and the receiver can keep the date and time information .

The extended Yapp reply is : RT Receive_TPK and is coded : [ACK] [ACK]

If the receiver reply is RT the protocol used will be what I have called YappC for Yapp with checksum. When the sender gets this packet he MUST use YappC.

Data Packets extension.

If the receivers reply is RT the protocol used will be YappC. The checksum allows detection of packets corrupted along the link, particularly on the RS232 lines where there is no error control or correction (or it's very poor!)

Data packets : [STX] [Len] [Data] [Checksum]

Checksum is the sum of all data bytes anded with FF in 8 bits like Xmodem.

If the checksum is bad then the receiver must send a Cancel and enters CW state.

Crash Recovery.

A new field has been added to the resume reply to tell the sender if the receiver can use YappC or not. Resume is sent instead of RF (or RT).

Resume reply for Yapp: (as used before by TPK and FBB)

```
[NAK] [Len] [R] [NUL] [Received Length] [NUL]
      I      I      I
      I      I      +-- in ASCII as in the header
      I      +-- as Resume !
      I
      +----- len of the following bytes
```

Resume reply for YappC:

```
[NAK] [Len] [R] [NUL] [Received Length] [NUL] [C] [NUL]
                                         I
      Tells sender I can use YappC -----+
```

When the sender gets this packet then he must also use YappC.

18.20 APPENDIX 20: Telephone-modem:

The FBB software allows the connection of a telephone modem on a serial port.

This section is split into these parts:

Driver

Type of modem

Protocol

Wiring for external modem

Usage

Example of PORT.SYS for DosFBB

Example of PORT.SYS for WinFBB

Example of PORT.SYS for LinFBB

Example of INITTNCx.SYS for modem

Example of FORWARD.SYS

Driver:

The modem must be declared in PORT.SYS.

In WinFBB the necessary driver is in FBBCOMM.DRV. This driver replaces the original COMM.DRV supplied by Windows.

In DosFBB it is necessary to use the FBBIOS.COM program. This driver performs BBS to Modem interfacing. It is therefore necessary to load FBBIOS prior to BBS start up. A line FBBIOS will be added ahead of APPEL.BAT file. See below an example of the APPEL.BAT file.
Of course, the FBBIOS used will have to match the serial port on which the modem is to be connected (address and IRQ).

In LinFBB it is necessary to compile the Linux Kernel with SLIP (serial line).

The modem-driver will modify the RS232 speed according to the CONNECT info received from the modem :

```
CONNECT 1200 set the driver to 1200 Baud.  
CONNECT 2400 set the driver to 2400 Baud.
```

The driver is reset to the default speed rate (specified in PORT.SYS) at disconnect.

At connect, the text ; "xxxx BBS. Phone Access" is sent by default. This can be modified by creating a file MODEM.ENT in the SYSTEM directory.

You can specify in the PORT.SYS file that you accept connections, even if the user is not registered. In this case, the connection will be "read-only", the user has only a read only-access, he cannot download files or write messages, except to the sysop.

When a non-registered user is connected, he will be signified that he only has a "read only access", and will be prompted to go on. If he answers "Y", he will go to the BBS menu, else he will be asked again for a callsign.

Type of modem:

Various experiments have been carried out with several different kinds of (external, or internal cards) modems. If possible, modems at 14.400 or better 28.800 bps should be used; the modem speed is to be declared in PORT.SYS. Some modems can work different speeds, with an internal speed conversion between the modem and the computer. For example, the modem can transmit and receive on the telephone line at 14400 bps and talk to the PC at 19200 bps.

If the modem uses the HAYES protocol, the INITTNCx.SYS file must hold the correct HAYES init lines for the modem.

Protocol:

The configuration used is 1 start bit, 8 data bits, 0 parity, 1 stop bit, that is to say 10 bits words during the transmission.

The Minitel standard is not supported (7 bits, even parity).

The software being parametered by default in 8 bits, it was not possible to modify this protocol, as the binary file transfers are carried out on 8 bits.

NB.: FBBIOS/FBBCOMM.DRV handles XON/XOFF protocol during the transmissions.

Wiring for external modems:

The modem must be able to tell the BBS that a call has been detected with the remote modem carrier detection. This is the reason why all the RS232 signals are used by the BBS. The RS232 cable has to be wired up pin to pin with no inversion: pins 2, 3, 4, 5, 6, 7, 8, 20, and 22 of the DB25 connector.

Usage:

BBS users are not authorized to connect the BBS through a modem as per the default settings.

The SysOp must declare the modem users. To do it, perform a user edition (EU call) and put M (for Modem) then W pass_word. A password made of the three last letters of the callsign can be attributed for the first connect. Then the user will be able to change it through the NP command. The authorized callsigns can be listed by the DM command.

When a modem call occurs, once the modulations have been detected, the modem advises the BBS that a connection is effective on the telephone line. Then the BBS asks for the caller's call sign and password.

The user is granted three tries before being disconnected. If everything is all right, the user can access the same commands like connected on a radio channel, he will have to disconnect using the B command.

The log file is maintained for the connections through the modem channel. Keep in mind to add a line in the LOGFBB.CNF file for the modem calls to be processed in the stats analysis by LOGSTAT.

For WinFBB:

FBBCOMM.DRV is already installed when Windows is started. So there is no more driver to install for the modem.

For LinFBB:

Make sure you have compiled the Linux kernel with SLIP (serial line), or if you compiled SLIP as module you must load this.

For DosFBB:

Example of APPEL.BAT with FBBIOS for a modem connected to COM3/IRQ4 :

```
echo off
break off
echo Press ^C to interrupt !
sleep 3
fbbios 3 03E8 4
serv %1
etc.....
```

Example of PORT.SYS file with 3 serial port used:

COM1 for a TNC, COM2 for a TNC, COM3 for the modem.

Example for DosFBB:

FBBIOS will have been previously configured for COM3 (3E8, IRQ 4).

File for programming of channels and TNCs.

#

Ports : How many ports (COM1, COM2, Etc...)

TNCs : How many TNCs and modems in use. With multiplexer

```

# there can be up to 4 TNCs per port.
#
#Ports TNCs
3      3
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#             as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS and FBBIOS.
#             7 = TCP/IP.  Needs WINSOCK.DLL.  Put port-address as 17.
#             TNC-emulation is T (see below)
#             8 = TFWin.dll
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB.  Interface 6
#         replaces both.  (FBBCOMM.DRV).  Neither ESS nor FBBIOS can be
#         used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux.  Can work via serial port (D), via AX25 domain
#               socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#             as stated in INIT.SRV, if any..
# Address    : Address of port in hexadecimal (Needed for multiplexer).
#               In LinFBB:
#               Address is the device name (/dev/cua0).
#               Be sure you have the rights to access to the device (rw-rw-rw-).
#               When using kernel AF_AX25 socket, address is not used.
#               When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud       : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (Hex) Baud
1      1      3F8      9600
2      1      2F8      9600
3      3      3E8      1200
#
# TNC       : Number on TNC in use.  Use 0 for file-forward !
# NbCh      : Number of channels I want to use in the TNC.
#             Maximum available channels depend on firmware.
# Com       : Number of the COM-port. Com1, Com2 etc.
# MultCh    : Number of channel if port-multiplexer is used, otherwise 1.
#             In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
# PacLen    : PACLEN on this TNC.
# Maxframe  : The maximum nb of frames the TNC will send at a time.
# NbFwd     : Number of channels for OUTGOING forward at same time.
# MxBloc    : Size of forward-block in kb.
# M/P-Fwd   : Minute of the hour for start of forward, and period
#             (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED

```

```

#           K : KAM hostmode.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.      : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacIn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1     5   1   1     230   2     1     10  10/60  UDW  144.675
2     4   2   1     250   3     2     5   12/30  UDYW  430.675
3     1   3   1     128   2     0     5   00/60  UM   MODEM
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1     1   LA1B-1      B
#
# End of file.
#

```

Same example for WinFBB:

FBBCOMM.DRV is already loaded by Windows.

```

# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
3      3
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS and FBBIOS.
#             7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
# replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
# used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#               socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#

```



```

#           4 = DRSI card with driver
#           5 = TFPCR/TFPCX interface.  Interrupt MUST be 0xFD or the same
#           as stated in INIT.SRV, if any..
# Address   : Address of port in hexadecimal (Needed for multiplexer).
#           In LinFBB:
#           Address is the device name (/dev/cua0).
#           Be sure you have the rights to access to the device (rw-rw-rw-).
#           When using kernel AF_AX25 socket, address is not used.
#           When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (Hex) Baud
1      6      3F8      9600
2      6      2F8      9600
3      6      3E8      1200
#
# TNC       : Number on TNC in use.  Use 0 for file-forward !
# NbCh      : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com       : Number of the COM-port. Com1, Com2 etc.
# MultCh    : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
# PacLen    : PACLEN on this TNC.
# Maxframe  : The maximum nb of frames the TNC will send at a time.
# NbFwd     : Number of channels for OUTGOING forward at same time.
# MxBloc    : Size of forward-block in kb.
# M/P-Fwd   : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.     : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacLn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1      5      1      1      230      2      1      10    10/60    UDW  144.675
2      4      2      1      250      3      2       5    12/30    UDYW  430.675
3      1      3      1      128      2      0       5     00/60     UM   MODEM
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1      1    LA1B-1      B
#
# End of file.

```

#

Same example for LinFBB:

Example of PORT.SYS file with 3 serial port used:
COM1 (/dev/cua0) for a TNC, COM2 (/dev/cua1) for a TNC,
COM3 (/dev/cua2) for the modem.

File for programming of channels and TNCs.

#

Ports : How many ports (COM1, COM2, Etc...)

TNCs : How many TNCs and modems in use. With multiplexer
there can be up to 4 TNCs per port.

#

#Ports TNCs

3 3

#

#In WinFBB ONLY THESE interfaces are available:

Interface : 2 = BPQ-node (BPQ in AA4RE-mode)

4 = DRSI

5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
as stated in INIT.SRV, if any..

6 = Windows-driver, replaces both ESS and FBBIOS.

7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.

TNC-emulation is T (see below)

8 = TFWin.dll

BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6

replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be

used with WinFBB !

#

#In LinFBB ONLY this interface is available:

Interface 9 = Linux. Can work via serial port (D), via AX25 domain

socket (X) or via Telnet port (T).

#

#In DosFBB ONLY THESE interfaces are available:

Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)

2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)

3 = Telephone-modem with FBBIOS

4 = DRSI card with driver

5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
as stated in INIT.SRV, if any..

#

Address : Address of port in hexadecimal (Needed for multiplexer).

In LinFBB:

Address is the device name (/dev/cua0).

Be sure you have the rights to access to the device (rw-rw-rw-).

When using kernel AF_AX25 socket, address is not used.

When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)

Baud : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.

#

Use same number of lines as number of ports.

#

#Com Interface Address (Hex) Baud

1 9 /dev/cua0 9600

2 9 /dev/cua1 9600

3 9 /dev/cua2 1200

#

TNC : Number on TNC in use. Use 0 for file-forward !

NbCh : Number of channels I want to use in the TNC.

Maximum available channels depend on firmware.

Com : Number of the COM-port. Com1, Com2 etc.

MultCh : Number of channel if port-multiplexer is used, otherwise 1.

```

#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
# Paclen   : PACLEN on this TNC.
# Maxframe: The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1    5    1    1      230    2      1      10    10/60   UDW  144.675
2    4    2    1      250    3      2       5    12/30  UDYW  430.675
3    1    3    1      128    2      0       5    00/60   UM   MODEM
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
1    1    LA1B-1      B
#
# End of file.
#

```

Example of INITTNCx.SYS file for HAYES modem :

```

#
ATZ
ATS0=1

```

Example of FORWARD.SYS file to forward via modem. A protocol with correction (like MNP) must be used to avoid any modifications of texts. The HAYES command ATDT dials the line, and when the connection is established, the callsign (preceded with a "." to avoid echoing) is sent twice and then the password. The callsign is sent twice in case of a problem the first time. ^M is the <return> variable.

```

A LA1B
#
P C                                     <-- Modem port.
#
C C F6FBB ATDT19~~~12345678^M          <-- Connects F6FBB using HAYES.
V .F6FBB^M~~~.F6FBB^M~~~PASSWORD^M    <--sends callsigns and password.
#
B LA1B

```

F LA6CU

In the PORT.SYS file, the number of forward channels for the TNC (modem) must be 1.

18.21 APPENDIX 21: File filtering: -----

There is a possibility to filter files that are uploaded to the BBS. This is done by a F_FILTER program.

The F_FILTER may also be created as a DLL. Both for WinFBB and DosFBB (!!). The filter must be installed in the path (\FBB\BIN) of Dos.

If the F_FILTER exists, the temporary file is no longer deleted by WinFBB (unless return value from F_FILTER is 0), the filter must decide what to do.

F_FILTER receives as parameter a file in which are given information such as asked filename and label. If F_FILTER returns 0, then the file is normally recorded (and the temporary file is deleted), else nothing is done by FBB. The text returned by F_FILTER will be sent to the user. So an archive may be compressed, scanned, and a report may be sent to the user. This can be done in real time or delayed.

18.22 APPENDIX 22: MAIL.IN: -----

This file is used for importing messages directly to the BBS. FBB checks every minute for the existence of this file (filename and path may be changed in INIT.SRV, but the default is \FBB\MAIL.IN). If the file exists, the messages in the file will be imported directly into the BBS.

The format of each message is identical to the format of any message normally sent to the BBS. Example of short MAIL.IN with 2 messages:

SP F6FBB < LA6CU	(S-line)
Testmessage	(Title)
This is only a test	(Message)
/EX	(/EX)
SB ALL @ EU < LA6CU \$12345ABCD	
Test Message	
This is just another test	
/EX	

Some sevrvers include a #-sign between each message to force the from-bbs to become the callsign of this BBS. This is still allowed, but should maybe not be necessary.

After the file has been processed (the messages imported) the file is automatically deleted.

18.23 APPENDIX 23: Installation-help for Linux: -----

Why use FBB with Linux ?

Info on AF_AX25, AF_NETROM and AF_ROSE.

Install AF_AX25, AF_NETROM and AF_ROSE.

Config AF_AX25, AF_NETROM and AF_ROSE for KISS TNC and SCC card.

Telnet installation (Appendix 24).

Why use FBB with Linux ?

LINUX is a true multitasking system where many applications may run concurrently. When Linux (and other UNIX systems) has finished booting it is possible to run one or more applications running concurrently (NODE, BBS, CLUSTER, TCPIP, etc...) on the same CPU and above all very fast and secure.

Another good thing for linux is that the AX25 is included in the kernel, and then you need no TNC, no BPQ or other. You only need hardware. (think of a DOS including Netrom, Rose, AX25, TCPIP and other software....)

Info on AF_AX25, AF_NETROM and AF_ROSE.

Standard kernel AF_AX25, AF_NETROM and AF_ROSE sockets
AX.25, NET/ROM and ROSE protocol are built into the Linux kernel.

I recommend to use these interfaces, because it will give you more flexibility to build a system with many futures..

This choice may be little difficult to set up, because the functions is depended of lots of things.. eg:

- Linux kernel version
- ax25-utils version
- z8530drv version
- net-tools version
- ... and more

If you desire to use this/these interfaces,
i strongly recommend to read at least AX25-HOWTO!

The AX25-HOWTO located at:
<http://sunsite.unc.edu/mdw/HOWTO/AX25-HOWTO.html>

More HOWTO's located at:
<http://sunsite.unc.edu/mdw/HOWTO/HOWTO-INDEX-3.html>

The Linux Documentation Project Homepage is located at
<http://sunsite.unc.edu/mdw/linux.html>

Install AF_AX25, AF_NETROM and AF_ROSE.

How to set up Linux kernel AF_AX25, AF_NETROM, AF_ROSE socket.

These examples are for Linux kernel 2.0.27 patched with ax25-module-7.

First, get these files (or newer files).

Linux kernel 2.0.27
available @ <ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/>

ax25-utils-2.0.12c.tar.gz
ax25-module-7.tar.gz
ax25-utils-module-7.tar.gz
available @ <http://www.cs.nott.ac.uk/~jsn/>

Unpack kernel 2.0.27 in `~/usr/src` with `'tar zxvf linux-2.0.27.tar.gz'`
this will unpack kernel 2.0.27 to a directory called `'linux'`,
beware if you already have a directory named `'linux'` in `~/usr/src`.

Tip:

I rename my kernel source directory to kernel version with `'mv linux linux-2.0.27'`

then i do a symbolic link to this directory with `'ln -s linux-2.0.27 linux'`

Before i install a new kernel, i remove the symbolic link (eg: `rm linux`), this allows me to have more kernel sources installed at same time.. :-

Okay, now it's time to compile the `'ax25-utils-2.0.12c.tar.gz'` , unpack this file in `'/usr/src'` with `'tar zxvf ax25-utils-2.0.12c.tar.gz'`.

I can't compile `'ax25-utils-2.0.12c.tar.gz'` after patching the kernel with `'ax25-module-7.tar.gz'`... :-/

Read the `'INSTALL'` file how to compile, and install this package.

Now it's time to patch the kernel with `'ax25-module-7.tar.gz'`, unpack this file in `'/usr/src'` with `'tar zxvf ax25-module-7.tar.gz'`, then type `'cat ax25-module-7/ax25-2.0.27-2.1.19.diff | patch -p0'`

Okay, do `'cd linux'` now, read the file README now, to know how to set up the kernel!

Okay, now it's time to run `'make config'` , be sure that you enable

- Amateur Radio AX.25 Level 2 (CONFIG_AX25)
- Amateur Radio NET/ROM (CONFIG_NETROM)

If you want to use ROSE, enable this..

- Amateur Radio X.25 PLP (Rose) (CONFIG_ROSE)

You can make these as loadable modules, or to be compiled into the kernel.

After you finished with `'make config'` , then type `'make dep; make clean; make zImage'` to compile the new kernel with AX.25 and NET/ROM support (and ROSE if defined).

After compilation, install the new kernel, as in the `/usr/src/linux/README` file.

Okay, you should have rebooted and started up the new kernel now..

Now it's time to compile `'ax25-utils-module-7.tar.gz'` , unpack this file in `'/usr/src'` with `'tar zxvf ax25-utils-module-7.tar.gz'`, do `'cd ax25-utils.module-7'` and read the INSTALL file how to compile and install!

If all steps has gone without problem, you are now ready to set up your Linux system

to work with kernel AX.25, NET/ROM (and ROSE if defined), CONGRATULATIONS!

Now you should go to `'/usr/src/ax25-utils-2.0.12c'` and read the README file, to find out what files you should need to change for suite you.

All configuration files are in `'/etc/ax25'`.

Tip:

Read the AX25-HOWTO @ <http://sunsite.unc.edu/mdw/HOWTO/AX25-HOWTO.html>

Config AF_AX25, AF_NETROM and AF_ROSE for KISS TNC and SCC card.

How to setup LinFBB to use kernel AF_AX25, AF_NETROM and AF_ROSE sockets.

Here is a example with LinFBB and one KISS TNC at COM1, 8 channels.

```
# /etc/ax25/axports
#
# The format of this file is:
#
#name callsign      speed paclen      window      description
#
ax0    SM6TKY-3      9600  256    7      Tx: 437.975 Rx: 434.000 MHz 9600bps
```

```
# /usr/local/fbb/system/port.sys
#
# Comments is deleted to save space
#
#Ports TNCs
 1      1
#
#
#Com Interface Adress (device) Baud
 1    9      ****      9600
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
 1    8    1  ax0    250    7    2    10    30/01  XUWYL  ax0
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
#1    2    SK6BA-0      B
#
# End of file.
#
```

```
# /etc/rc.d/rc.ax25
#
# This file is called from /etc/rc.d/rc.local
#
# attach a TNC2 with KISS
#
# (/dev/cua0 = COM1, /dev/cua1 = COM2, /dev/cua2 = COM3, /dev/cua3 = COM4)
#
/usr/sbin/kissattach -i 44.140.208.129 /dev/cua0 ax0
#
# set up default IP routings
# This is not needed if you don't want to run TCP/IP on Radio
#
/sbin/ifconfig ax0 44.140.208.129 netmask 255.255.255.0 broadcast 44.140.208.255
arp mtu 256
/sbin/route add -host sk6ba.ampr.org ax0
/sbin/route add -net 44.140.208.0 gw sk6ba.ampr.org netmask 255.255.255.0 ax0
/sbin/route add default gw sk6ba.ampr.org ax0
#
# Send ID every 10 min
#
/usr/sbin/beacon -t 10 ax0 "This Packet Radio Node is Powered with Linux Unix" &
#
```

```
# If you want to run LinFBB in X-Windows,
# comment out this below...
#
# Start up LinFBB daemon
#
/home/xfbb/xfbb.sh -d &
#
```

Here is one example with one KISS TNC2 on COM2, NET/ROM interface, 16 channels.

```
# /etc/ax25/axports
#
# The format of this file is:
#
#name callsign      speed paclen      window      description
#
ax0    SM6TKY-3      9600  256    7      Tx: 437.975 Rx: 434.000 MHz 9600bps
```

```
# /etc/ax25/nrports
#
nr0 SM6TKY-13 #IP 236 Radio 9600bps
nr1 SM6TKY-0  XFBB 236 NET/ROM<->XFBB
```

```
# /etc/ax25/nrbroadcast
ax0  1      192  10    1
```

```
# /usr/local/fbb/port.sys
#
# Comments is deleted to save space
#
#Ports TNCs
1      2
#
#
#Com Interface Adress (device) Baud
1      9      ****      9600
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1    8    1  ax0    250    7     2     10    30/01  XUWYL  ax0
2    8    1  nr1    236    7     2     10    30/01  XUY    netrom
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
#1    2    SK6BA-0      B
#
# End of file.
#
```

```
# /etc/rc.d/rc.ax25
```



```

#
# This file is called from /etc/rc.d/rc.local
#
# attach a TNC2 with KISS
#
# (/dev/cua0 = COM1, /dev/cua1 = COM2, /dev/cua2 = COM3, /dev/cua3 = COM4)
#
/usr/sbin/kissattach -i 44.140.208.129 /dev/cua1 ax0
#
# attach NET/ROM interface
/usr/sbin/nrattach -i 44.140.208.129 nr0
#
# This is for LinFBB NET/ROM
/usr/sbin/nrattach -i 44.140.208.129 nr1
#
# set up default IP routings
# This is not needed if you don't want to run TCP/IP on Radio
#
/sbin/ifconfig ax0 44.140.208.129 netmask 255.255.255.0 broadcast 44.140.208.255
arp mtu 256
/sbin/route add -host sk6ba.ampr.org ax0
/sbin/route add -net 44.140.208.0 gw sk6ba.ampr.org netmask 255.255.255.0 ax0
/sbin/route add default gw sk6ba.ampr.org ax0
#
# Start up NET/ROM deamon
/usr/sbin/netromd
/usr/sbin/nrparms -parms -qual 192
#
# Send ID every 10 min
/usr/sbin/beacon -t 10 ax0 "This Packet Radio Node is Powered with Linux Unix" &
#
# If you want to run LinFBB in X-Windows,
# comment out this below...
#
# Start up LinFBB deamon
/home/xfbb/xfbb.sh -d &
#

```

Here is a example with 2 ports SCC (PA0HZP compatibelt) card installed, 16 channels, and SCC driver is a loadable module.

```

# /etc/z8530drv.conf
#
# Lines must not be longer than 255 bytes!
#
# Hardware section
#
# SCC chip 1
#
chip 1
data_a 0x153          # data port A
ctrl_a 0x152          # control port A
data_b 0x151          # data port B
ctrl_b 0x150          # control port B
irq 5                 # IRQ No. 5 (#)
pclock 4915200        # clock (+)
# board PA0HZP
# escc yes
# escc no              # enhanced SCC chip? (8580/85180/85280) (*)

```

```

# vector 0          # latch for interrupt vector ($)
# special no        # address of special function register (*)
# option 0          # option to set via sfr (*)
#
# (*) option, default value is 0
# (+) option, default is 4915200
#
# SCC chip 2
#
# chip      2
# data_a    0x157
# ctrl_a    0x156
# data_b    0x155
# ctrl_b    0x154
# irq       5          # IRQ No. 5 (#)
# pclock    4915200    # clock (+)
# board     PA0HZP
# escc      no
# vector     0
# special    no
# option     0
#
# (#) all chips can share one interrupt, or you may use multiple INTs,
#      but there must be at least one IRQ defined in this system.
# ($) only one Vector Latch for all Chips is allowed.
#
#
# Software section
#
# The order of entries IS important!
#
#      first      : DEVICE
#      then       : MODEM section
#      after that : Level 1 parameters
#
# Definition of /dev/scc0
#
device      /dev/scc0
speed 9600
clock divider
mode nrzi
# rxbuffers 8
# txbuffers 16
# bufsize 1152
#
txdelay 30
persist 250
slot 8
tail 8
fulldup 0
wait 12
min 3
maxkey 7
idle 3
maxdef 120
group 0
txoff off
softdcd off          # use software dcd
#
# Definition of /dev/sccl
#
device      /dev/sccl
speed 1200

```

```
clock divider
mode nrzi
# rxbuffers 8
# txbuffers 16
# bufsize 1152
#
txdelay 30
persist 250
slot 8
tail 8
fulldup 0
wait 12
min 3
maxkey 7
idle 3
maxdef 120
group 0
txoff off
softdcd off
```

```
# /etc/ax25/axports
#
# The format of this file is:
#
#name callsign      speed paclen      window      description
#
scc0 SM6TKY-3      9600 256 7      Tx: 437.975 Rx: 434.000 MHz 9600bps
scc1 SM6TKY-1      1200 256 7      MARKnet Lan 144.800 MHz 1200bps
```

```
# /usr/local/fbb/port.sys
#
# Comments is deleted to save space
#
#Ports TNCs
1 2
#
#
#Com Interface Adress (device) Baud
1 9 ***** 9600
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1 8 1 scc0 250 7 2 10 30/01 XUWYL scc0
2 8 1 scc1 250 7 2 10 30/01 XUWYL scc1
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
#1 2 SK6BA-0 B
#
# End of file.
#
```

```
# /etc/rc.d/rc.ax25
#
# This file is called from /etc/rc.d/rc.local
```

```

#
# Load the SCC module and init
/sbin/insmod scc
/sbin/sccinit
#
# attach the SCC card
#
/sbin/ifconfig scc0 44.140.208.129 hw ax25 sm6tky-3 up
/sbin/ifconfig scc1 44.140.208.129 hw ax25 sm6tky-1 up
#
# set up default IP routings
# This is not needed if you don't want to run TCP/IP on Radio
#
/sbin/ifconfig scc0 44.140.208.129 netmask 255.255.0.0 broadcast 44.140.255.255
arp mtu 256
/sbin/ifconfig scc1 44.140.208.129 netmask 255.255.255.0 broadcast
44.140.208.255 arp mtu 256
/sbin/route add -host sk6ba.ampr.org scc1
/sbin/route add -net 44.140.208.0 gw sk6ba.ampr.org netmask 255.255.255.0 scc1
/sbin/route add default gw sk6ba.ampr.org scc1
#
#
# Send ID every 10 min
/usr/sbin/beacon -t 10 scc0 "This Packet Radio Node is Powered with Linux Unix"
&
/usr/sbin/beacon -t 10 scc1 "This Packet Radio Node is Powered with Linux Unix"
&
#
# If you want to run LinFBB in X-Windows,
# comment out this below...
#
# Start up LinFBB daemon
/home/xfbb/xfbb.sh -d &
#

```

18.24 APPENDIX 24: TELNET interface:

<Image>

TELNET interface.

(Only for WinFBB and LinFBB)

The contents of this file is:
 Info about Telnet interface
 Example of PORT.SYS for LinFBB
 Example of PORT.SYS for WinFBB
 Example of FORWARD.SYS

<Image>Info about Telnet interface

Telnet is a communication tool like a terminal, but using TCP-IP. That is you can use it on a network like the web. You MUST validate your local echo first.

It's possible to use Telnet with LinFBB and WinFBB to be used in User contacts or in Forward contacts.

The address to use may be 189C (189C hex = 6300 decimal).

But, it's also possible to define another port address,
for example if you want Telnet access to LinFBB on port 23,
(port 23 is standard for Telnet, but 23 is also used by inetd in Linux!)
then change the 'address' of COM2 to 17 (hex 17 = 23 decimalt)

Tip:

You may put the R letter (Read only default access) which gives a read-only
if the user is unknown or not authorized for Modem/Telnet (M Flag in EU
command).

<Image>Example of PORT.SYS for LinFBB

Standard PORT.SYS with 1 TNC (COM1) with WA8DED-hostmode,
4 channels, 1 channel for outgoing forward,
and one Telnet interface listening on port 6300,
8 channels, 1 channel for outgoing forward.

```
# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
2      2
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS and FBBIOS.
#             7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
#          replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
#          used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#               socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#             2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#             3 = Telephone-modem with FBBIOS
#             4 = DRSI card with driver
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#
# Address    : Address of port in hexadecimal (Needed for multiplexer).
#              In LinFBB:
#              Address is the device name (/dev/cua0).
#              Be sure you have the rights to access to the device (rw-rw-rw-).
#              When using kernel AF_AX25 socket, address is not used.
#              When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud       : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
```

```

# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1    9          /dev/cua0      9600
2    9          189C           0
#
# TNC      : Number on TNC in use. Use 0 for file-forward !
# NbCh     : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com      : Number of the COM-port. Com1, Com2 etc.
# MultCh   : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#           When using kernel AF_AX25 socket in Linux, MultCh is the
#           interface name (eg: ax0)
# Paclen   : PACLEN on this TNC.
# Maxframe : The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode. Must use ESSKAM driver.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
0    0    0  0      0      0      0      0      00/01  ---  File-fwd.
1    4    1  1     230     4      1     10     30/60  UDYL  433.650
2    8    2  0     250     2      1     10     00/60  TUWR  Telnet
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
#1    1    LA1B-1      B
#
# End of file.
#

```

<Image>Example of PORT.SYS for WinFBB

Standard PORT.SYS with 1 TNC (COM1) with WA8DED-hostmode,
4 channels, 1 channel for outgoing forward,
and one Telnet interface listening on port 6300,
8 channels, 1 channel for outgoing forward.

FBB7.00

```

#
#
#Ports TNCs
2      2
#
# Com      : COM-number (1,2,...8)
#
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#             4 = DRSI
#             5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#               as stated in INIT.SRV, if any..
#             6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#             7 = TCP/IP. Needs WINSOCK.DLL.
#               TNC-emulation is T (see below)
#             8 = TFWin.dll
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
#         replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
#         used with WinFBB !
##
#Com Interface Address (Hex) Baud
1      6      3F8      9600
2      7      189C      0
#
# TNC      : Number on TNC in use. Use 0 for file-forward !
# NbCh     : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com      : Number of the COM-port. Com1, Com2 etc.
# MultCh   : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#           When using kernel AF_AX25 socket in Linux,
#           MultCh is the interface name (eg: ax0)
# Paclen   : PACLEN on this TNC.
# Maxframe : The maximum nb of frames the TNC will send at a time.
# NbFwd    : Number of channels for OUTGOING forward at same time.
# MxBloc   : Size of forward-block in kb.
# M/P-Fwd  : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.    : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
0      0      0      0      0      0      0      0      00/01 ---- File-fwd.
1      4      1      1      230      4      1      10      30/60 UDYL  433.650
2      8      2      0      250      2      1      10      00/60 TUWR  Telnet
#

```

```
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
# 1      1      LA1B-1      B
#
# End of file.
#
```

<Image>Example of FORWARD.SYS

It's syntax is common to both LinFBB and WinFBB.

```
A      SK6BA
      P A
#      C C <callsign> <ip address> <ip port>
      C C SK6BA 44.140.208.134 6300
#      V <login>$W<password>$W
      V SM6TKY$WPASSWORD$W
#
# the rest of fwd block...
```

<Image>

<Image>

This page was last updated

18.25 APPENDIX 25: TFPCX / TFPCR / TFKISS (Hostmode KISS-driver):

TFPCX / TFPCR / TFKISS (Hostmode KISS-driver).

It is possible to link FBB to a TheNetNode or similar systems running on a separat PC with a simple 0-modem cable.

For this TF-KISS, TFPCR or TFPCX can be used as com-driver, and the link must be defined as a KISSLINK in the nodesystem.

Another possibility is to use the driver with tnc's in kiss-mode etc..

The only restriction is that baycom modem do not work, but SCC cards do work.

It must be called with the same interrupt as stated in INIT.SRV as TFPC-interrupt, or OxFD by default (if the INIT.SRV does not specify it).

It can be used with DosFBB or WinFBB running under WIN3.xx or WIN95.

IMPORTANT : TFPCX, TFPCR or TFKISS always must be called before starting WINDOWS and/or FBB.

Better add the call into your AUTOEXEC.BAT with a line like this (example using TFKISS) :

```
TFKISS -I:FD -B:9600
```

Comport 1 is default, but other ports as well as other parameters can be specified (look in your manual).

Place the driver itself in your \FBB\BIN directory.

In PORT.SYS you must use 5 in INTERFACE, and D in type host mode.

The interface behaves in many ways as WA8DED hostmode.

If you intend to use higher speed than 9600 baud you must use UART 16550 in your serial ports and PCs not slower than 386 40 MHz. Otherwise you probably will get errors during binary transfer or forward.

Example of PORT.SYS:

```
# FBB7.00
#
#Ports TNCs
1      1
#
#Com Interface Address (device)  Baud
1      5      3F8      9600
#
#TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode  Freq
0      0      0      0      0      0      0      0      00/01  ----- File-fwd.
1     10      1      1     236      7      4     30     00/15      UDLY VHF/UHF/SHF
#
#TNC Nbs Callsign-SSID Mode
#1      1     LA1B-1      B
#
# End of file.
#
```

Example of INITTNC1.SYS:

```
I OZ7BOX      # Set callsign
M IUS         # Monitor I, U and S-frames - only necessary with UNPROTO
function active
@T2 150       # T2 timer settings
W 10          # Slottime
P 255         # P-persistence
O 7           # Maxframe
Y 10          # Number of channels
```

Other parameters can be set - look in your manual.
Parameters can also be set with F7 - TNC commands.

If you have outgoing on more than one channel at the time, you have to use SSID's in your forward connects for each channel.
With hostmode SSID's can be set for a channel using the L-command (send tnc-command) together with the I-command (hostmode tnc-command - SET CALLSIGN).

Examples (only part of the connect strings showed) :

Forward to the first call:

```
L I OZ7BOX-10  # Use SSID -10 as your outgoing call
C C ...       # The connect strings
```

Forward to the second call:

```
L I OZ7BOX-9   # Use SSID -9 as your outgoing call
C C ...       # The connect strings
```

etc....

SSID's from -0 to -15 can be used, but NOT SSID's already in use.

18.26 APPENDIX 26: EMS/XMS memory:

EMS/XMS is no longer used in any version. But you need to put "1" to allow memory cache in the BBS. It now means "memory cache" and then allows to put the list of options which follow this line in memory to be faster.

18.27 APPENDIX 27: Using FBB with PE1CHL NET and the BPQ Hostmode Emulator:

PE1CHL NET and FBB work together under DESQview (or other Multi-Tasker's) with the "G8BPQ Hostmode Emulator" package. The setup is very similar to the method for using the regular BPQ code. Setup your .SYS files in the same and read the appendix on using BPQ mode as well for details.

The following differences should be noted.

In INITTNCx.SYS, monitoring is available but you may prefer to use the more comprehensive monitoring within NET instead.

In FORWARD.SYS, "C C SWITCH" will connect the BBS to the NET/ROM in NET, after this point the syntax is specific to NET and the options are :-

```
A LA2D
#
P B
#
C C SWITCH      Connect to the switch (Net/Rom in NET)
V P 70cm        Select the "70cm" port
C LA2D          Connect LA2D on the port we have just defined
#
B LA2D
F LA2D
#
-----
```

Alternately, you could predefine the downlink to LA2D in AUTOEXEC.NET with the command :-

```
netrom downlink LA2D 70cm
```

You would then omit the "port select" line in the above example and the "downlink" would be selected by NET on whatever port you have predefined. This could be any valid NET portname, whether it is an Ethernet or AX/IP encapsulated portname or just another regular port.

Setting up the G8BPQ Hostmode Emulator with FBB

You must edit and use BPQCFG.TXT from the G8BPQEMU package and compile it with BPQCFG.EXE which should result in a 3072 byte BPQCFG.BIN file.

You should load G8BPQ.COM (from the emulator) from AUTOEXEC.BAT and this can be loaded high with your favourite memory manager.

You will need to exercise the usual tricks to get the maximum memory from each DESQview window. NET should be started first, to setup the G8BPQ Emulator port mappings, so that Unproto (for TPK users) broadcasts are setup before FBB is booted.

In the autoexec.net file, the emulator will need to be started and the port mapping setup. You should choose port names that differ from the internal names of FBB or BPQ types to avoid confusion, an example is given below.

FREQ	USAGE	NET-PORT	BPQ-PORT	FBB-PORT
RS232	38400bd	A1	1	A
432.675	4800bd	70f	2	B
432.675	1200bd	70s	3	C
144.625	1200bd	2ms	4	D
144.625	4800bd	2mf	5	E
70.3125	1200bd	4m	6	F
50.650	1200bd	6m	7	G
Exp.	76800bd	exp	8	H

Then you would use the following in autoexec.net :-

```
ax25 start g8bpq
g8bpq ports A1 70f 70s 2ms 2mf 4m 6m exp
```

This will then map all the above ports to FBB and UI broadcasts will be available on those ports. If you want only selected ports, you could have an INITTNCx.SYS for them and specify if you want UI broadcasts or not. Alternately, an easier method is to specify only those ports you want to map to UI broadcasts as in the above example for NET and only ports setup this way will broadcast UI frames, ports not specified will not broadcast.

DESQview Window Management

The object is to get the maximum memory for FBB and this will always be available in the first DV window. The trick here is to open a dummy window first, as a place holder (a regular DOS one will do fine). Now you should start NET (which will automatically open in window 2) and once this has booted, you will need to switch to the 1st window, close it and then boot FBB. This will release the place holder and give the 1st DV window slot to FBB.

A script that does this automatically is called DESQVIEW.DVS and should be made with the script tools (see DV manual) from this ASCII file :-

{Learn {Shift-F12} " !Startup"}	Name the startup script
obd{Enter}	Open BIG DOS Window
{DESQ}ont	Open NET window
{Delay 5}	Wait 5 seconds
{DESQ}s1	Switch to the 1st window
{DESQ}cy	Close that window
{DESQ}ofb	Open FBB window
{Finish}	End of script

This assumes you have the following two letter assignments for windows in DESQview :-

```
BD = Big DOS
NT = PE1CHL NET
```

FB = FBB

Note there is a delay of 5 seconds to allow NET to boot fully before switching windows and starting FBB, this may need adjusting to suit the speed of your machine.

You will need to make some .PIF files for NET and FBB and the following examples will give you a good starting point :-

You will also set your DMPI allocation for FBB here too. Make sure you allocate at least 2048KB, otherwise FBB may load it's own and waste memory.

Setting up PIF file options in DESQview for FBB and NET Standard Options

Program Name.....: FBB 7.00b41

Keys to Use on Open Menu: FB

Memory Size (in K): 536

Program...: c:\FBB\APPEL.BAT

Parameters:

Directory.: C:\FBB

Options:

Writes text directly to screen.....: [Y]
Displays graphics information.....: [Y]
Virtualize text/graphics (Y,N,T).....: [N]
Uses serial ports (Y,N,1,2).....: [N]
Requires floppy diskette.....: [N]

Advanced Options

System Memory (in K).....: 0 Maximum Program Memory Size (in K): 600

Script Buffer Size.....: 1000 Maximum EMS/XMS/VCPI/DPMI (in K): 2048

Text Pages: 4

Graphics Pages: 1

Initial Video Mode:

Window Position:

Maximum Height: 25

Starting Height: 25

Starting Row...: 0

Maximum Width.: 80

Starting Width.: 80

Starting Column: 0

Shared Program

Pathname...:

Data.....:

Close on exit (Y,N,blank).....: [] Uses its own colors.....: [Y]
Allow Close Window command.....: [Y] Runs in background (Y,N,blank)...: [Y]
Uses math coprocessor.....: [Y] Keyboard conflict (0-F).....: [0]
Share CPU when foreground.....: [Y] Share EGA when foreground/zoomed.: [N]
Can be swapped out (Y,N,blank).: [N] Protection level (0-3).....: [0]

Standard Options

Program Name.....: NET

Keys to Use on Open Menu: NT

Memory Size (in K): 230

Program...: net.bat

Parameters:

Directory.: C:\

Options:

Writes text directly to screen.....: [Y]
Displays graphics information.....: [N]
Virtualize text/graphics (Y,N,T).....: [Y]
Uses serial ports (Y,N,1,2).....: [N]
Requires floppy diskette.....: [N]

Advanced Options

System Memory (in K).....: 0 Maximum Program Memory Size (in K): 700

Script Buffer Size.....: 1000 Maximum EMS/XMS/VCPI/DPMI (in K):

Text Pages: 1 Graphics Pages: 0 Initial Video Mode:

Window Position:

Maximum Height:	50	Starting Height:	50	Starting Row...:	0
Maximum Width.:	80	Starting Width.:	80	Starting Column:	0

Shared Program

Pathname...:

Data.....:

Close on exit (Y,N,blank).....:	[Y]	Uses its own colors.....:	[Y]
Allow Close Window command.....:	[N]	Runs in background (Y,N,blank)...:	[Y]
Uses math coprocessor.....:	[Y]	Keyboard conflict (0-F).....:	[0]
Share CPU when foreground.....:	[Y]	Share EGA when foreground/zoomed.:	[N]
Can be swapped out (Y,N,blank).:	[N]	Protection level (0-3).....:	[0]

Adjusting memory size window for NET

1. Open the NET window and then check from the command line to see how much memory is available in NET with the "mem" command.

2. Close NET and allocate another 10KB to the window repeat step 1.

When the amount of free memory does NOT increase, you have reached the maximum and should backoff until you reach the exact point where the free memory starts to drop and add a few KB to that value. (Note this may need adjusting as you upgrade to new NET releases).

Adjusting memory size window for FBB

1. Open FBB with a large value, say 600KB and reduce in 10KB increments until APPEL.BAT starts to run. Cancel and close the window.

2. Increase the memory by 1KB and repeat step 1 until APPEL.BAT will no longer run then back it off to the previous value.

BPQ type Ethernet frames

As with the regular BPQ switch, NET can be setup to handle BPQ ethernet frame mapping. This is achieved through the normal packet driver (for your card) and you need to attach this mapping BEFORE any other mappings to the packet driver in AUTOEXEC.NET :-

```
attach packet 0x60 g8bpq bpq 8 256 GB7OSP-14 ff:ff:ff:ff:ff:ff
attach packet 0x60 0 eth0 8 1500 anytype
```

18.28 APPENDIX 28: CONFERENCE COMMANDS:

Any text typed while in the conference is sent to all the parties involved. A header before the text shows its origin. The header is not repeated if the next text has the same origin.

In the conference, the commands should begin with a semi-colon in the first column, so as to be interpreted as commands and not text.

Commands in the Conference mode:

.C[PORT] CALLSIGN : Connect to a callsign on a port. The port indication is optional; the port of the connected user is otherwise chosen as a default.

.D CALLSIGN : Disconnects the callsign from the conference.

.H : Help

.W : Shows a list of the participants in the conference.

.Q : Exit the conference.

18.29 APPENDIX 29: GATEWAY COMMANDS:

B : Exit the Gateway

K : Enters Converse mode.

C : Connect.

D : Disconnect.

H : Help

J : Lists the 20 last stations heard on the port.

P : Link from one port to another.

Once the selection of an authorized port has been made, a channel will be granted, depending upon availability. This channel will use the users own callsign. Switching to converse mode is automatic when the corresponding station is connected; switching back to command mode takes place on a disconnection, or if the Esc character or a > is sent in the first column.

Switching to converse mode can also be made with the command K.

The C command must be followed by the destination callsign, and optionnaly

by the letter V followed by a string of digipeater callsigns. Like:
C Destination call V repeat1 repeat2 ...

18.30 APPENDIX 30: TFWIN.DLL.

The TFWIN.INI must be edited to match the comport configuration ie port speed etc.

TFWIN.INI can also define things like txdelay for hf or highspeed ports. It is possible to program most of the PORT specific parameters in a INITTNCn.SYS (first TFWIN port) with "DRSI-like" P line (look at the FBB doc for DRSI).

This is my port.sys for TfWin DLL... Think that Multch (as usual) must match with the port number of TFWIN.INI

The TFWIN.INI is using COM1 as "port 0 (MultCh 0)"

As it is a Windows application, the com1 MUST be known (and available) from Windows !

```
# FBB7.00
# File for programming of channels and TNCs.
#
# Ports : How many ports (COM1, COM2, Etc...)
# TNCs : How many TNCs and modems in use. With multiplexer
# there can be up to 4 TNCs per port.
#
#Ports TNCs
 1      1
#
#In WinFBB ONLY THESE interfaces are available:
# Interface : 2 = BPQ-node (BPQ in AA4RE-mode)
#              4 = DRSI
#              5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#                  as stated in INIT.SRV, if any..
#              6 = Windows-driver, replaces both ESS, ESSKAM and FBBIOS.
#              7 = TCP/IP. Needs WINSOCK.DLL. Put port-address as 17.
#                  TNC-emulation is T (see below)
#              8 = TfWin.dll
# BEWARE: The old interface 1 and 3 will NOT be used in WinFBB. Interface 6
#          replaces both. (FBBCOMM.DRV). Neither ESS nor FBBIOS can be
#          used with WinFBB !
#
#In LinFBB ONLY this interface is available:
# Interface   9 = Linux. Can work via serial port (D), via AX25 domain
#                  socket (X) or via Telnet port (T).
#
#In DosFBB ONLY THESE interfaces are available:
# Interface : 1 = Use external COMBIOS-driver (MBBIOS, ESS etc)
#              2 = BPQ-node v 4.05 and up (BPQ in AA4RE-mode)
#              3 = Telephone-modem with FBBIOS
#              4 = DRSI card with driver
#              5 = TFPCR/TFPCX interface. Interrupt MUST be 0xFD or the same
#                  as stated in INIT.SRV, if any..
#
# Address    : Address of port in hexadecimal (Needed for multiplexer).
#              In LinFBB:
#              Address is the device name (/dev/cua0).
#              Be sure you have the rights to access to the device (rw-rw-rw-).
```

```

#           When using kernel AF_AX25 socket, address is not used.
#           When using Telnet, address is the Telnet port in Hex (Hex 17 =
Telnet port 23)
# Baud      : Ports baud rate. Ignored by BPQ, kernel AF_AX25 socket and Telnet.
#
# Use same number of lines as number of ports.
#
#Com Interface Address (device)  Baud
1   8           3f8           9600
#
# TNC       : Number on TNC in use. Use 0 for file-forward !
# NbCh      : Number of channels I want to use in the TNC.
#           Maximum available channels depend on firmware.
# Com       : Number of the COM-port. Com1, Com2 etc.
# MultCh    : Number of channel if port-multiplexer is used, otherwise 1.
#           In DRSI use values from 0 to 7, by KAM use 1/VHF and 2/HF.
#           When using kernel AF_AX25 socket in Linux, MultCh is the
#           interface name (eg: ax0)
# PacLen    : PACLEN on this TNC.
# Maxframe  : The maximum nb of frames the TNC will send at a time.
# NbFwd     : Number of channels for OUTGOING forward at same time.
# MxBloc    : Size of forward-block in kb.
# M/P-Fwd   : Minute of the hour for start of forward, and period
#           (how many minutes between each forward-start).
# Port mode, one of these:
#           B : BBS-mode.
#           G : "Guest"-mode.
#           U : Normal-mode.
# Type host-mode, one of these:
#           D : WA8DED
#           K : KAM hostmode. Must use ESSKAM driver.
#           P : PK-232
#           Q : BPQ v 4.x
#           T : Ethernet/TCP-IP
#           X : AX25 domain socket (for Linux)
# Addition: One or more of these letters can be used too:
#           L : Send unproto beacon after each arriving mail.
#           M : Telephone-modem.
#           Y : Yapp allowed on this QRG.
#           W : Gateway allowed TO this QRG.
#           R : Modem port allowed in Read-only mode.
# Freq.     : Text to describe this port (max 9 characters, no space)
#
# Same number of lines as TNCs:
#
#TNC NbCh Com MultCh PacLn Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
0   0   0   0       0       0       0       0       00/01 ---- File-fwd.
1   7   1   0      230      4       1      10      30/60 DUWY  433.650
#
# Special callsigns and modes for some channels.
#
#TNC Nbs Callsign-SSID Mode
#1   1   LA1B-1      B
#
# End of file.
#

```