Those parameters will also be used for "reverse forwarding".

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

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. Take care to the capital and lowercase letters, the tests take them into account. 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.

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 : means only private
T S : means smaller first (instead of older first)
T 2 : means only less or equal to 2 KB.

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, personnal mail is forwarded first, and then bulletins.

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.

X (command)
Command to be executed by DOS (Selection of a frequency, as for an example). 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 automaticly first sent. That needs some more memory.

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 autimatically 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
```

It is now 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 pu 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 entered 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 tranceiver with a special programm 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.
```

18.8 Specific language files x.TXT, x.HLP, x.ENT, x.INF (SYSTEM\LANG).

These various files have the names of the languages followed by the suffixes TXT, HLP, ENT, or INF. They may content 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 reffered to a word following the command, (ex: ? HELP). Each help block should begin with a line of a special format:

```
@@ 14 S
```

The two @@ characters, without space, in the first column mark the beginning of a help block. They are followed by an identification number for the type of handling (see appendix 3), and of the word which identifies the help, in capital. A block ends at the line beginning with @@ (next block identifier).

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.

INF : File displayed when the command "I" is used. If the file does not exist, the display of the ! command will take place.

18.9 FBBLOG.nn File (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:COMMOD@EU [913] S:SUPPORT
043007034316M< 64097B F:I1ZEU T:ALL@EU [1096] S:ID1V
043007065416SQ *** BBS Quit
```

18.10 DIRMES.SYS (SYSTEM).

This binary file holds the informations 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.

18.11 STATIS.DAT (SYSTEM).

This binary file holds the informations 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.

18.12 WFBID.SYS (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.

18.13 INF.SYS file (SYSTEM).

This binary file holds the informations 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                    */
  char     free[27]  ;  /* 27  Reserved                    */
  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     qra[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.

18.14 HROUTE.SYS (SYSTEM).

This ascii file holds the information on BBS hierarchical routing. Each time a "R:" header is received in a message or bulletin, the hierarchical routing found in the line is added to a list and updates the HROUTE.SYS file.

This is an example of HROUTE.SYS :

```
# SM2LZK.GSTAD.BD.SWE.EU
# VE3JJV.NWON.CAN.NA
# W2NRE.NY.USA
# W6MCV.AZ.USA.NA
# W6MCV.AZ.USA.NOAM
V 3B8FP.MRU.AF
P 4X1RU.ISR.MDLE
V 4Z4SV.ISR.MDLE
V 7J1AAA.10.JNET1.JPN.AS
V 7K1DJL.11.JNET1.JPN.AS
V 7Z1AB.SAU.MDLE
```

The character in first column can be :

The line is ignored (comment, or line disabled).
P The line is persistent and will not change.
V The line is volatile and will change if a new route exists for this BBS.

When a new route is found for a BBS, the old one is commented by changing the V character with a #, and a new line will be appended at the end of the file.

HROUTE.SYS is automaticly updated by the software and does not need any maintenance.

This file can be sorted and comments deleted with a sequence like :

```
COPY HROUTE.SYS TEMP.$$$
TYPE TEMP.$$$ | SORT | FIND /v "# " > HROUTE.SYS
DEL TEMP.$$$
```

This sequence can be included in the APPEL.BAT loop.

This file will be no longer used and updated if the White Pages service is validated, as the WP service helds the same informations.

18.15 YAPLBL.SYS file (SYSTEM).

This binary file holds the descriptions of the FbbDos files.

It is organized with records, each record containing the filename in upcase characters (with the full path including disk name) and the description of this file.

Structure of one record (C language) :

```
typedef struct {
  char  filename[40];  /* Complete path of the filename */
  char  label[80]   ;  /* Description of the file       */
  long  creation    ;  /* Date of creation of the file  */
  long  unused      ;  /* Not used should be 0          */
} YapLbl;
```

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.

18.16 MEMO.SYS (SYSTEM).

This ascii file will substitute a text when typing 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
```

18.17 SATEL.DAT file (SAT).

This binary file holds the informations 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.

18.18 nnnnn.SAT (SAT) (Satellite characteristics).

The nnnnn.SAT files (where nnnnn is the NASA catalog numbe 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.

18.19 WP.SYS file format.

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
  unsigned     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 !

18.20 MESS.WP file format.

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.