

1.	Architecture	4
1.1.	APIs	4
1.1.1.	Command APIs	4
1.1.2.	High language APIs	4
1.2.	Usable files	4
1.2.1.	Physical files	4
1.2.2.	Spoolfiles	4
1.3.	Network target	5
1.3.1.	Logical address	5
1.3.2.	Physical address	5
1.3.3.	Search address	5
1.4.	Other transmitting functionality's	6
1.4.1.	Spool files gateways	6
1.4.2.	Office vision Gateway	6
2.	Messages transmitting and tracking	7
2.1.	Preliminary	7
2.2.	Application	7
2.3.	Queue	8
2.4.	Processing program	8
2.5.	Processing program Structure	9
2.6.	Diagram	10
3.	IPSNDRGRAPH command	11
3.1.	Syntax	11
3.2.	Keywords	11
3.2.1.	APPEME: Transmitting application name	11
3.2.2.	KEYUSR : use key of message	11
3.2.3.	OBJLIB: name of the file library	11
3.2.4.	OBJFIL: file name	11
3.2.5.	OBJMBR: member name	11
3.2.6.	SPLNAM: Spoolfile name	11
3.2.7.	SPLJOB: job name	12
3.2.8.	SPLNUM: Spoolfile name	12
3.2.9.	NOMLOG: logical name of the correspondent	12

3.2.10.	REFMSG: message reference _____	12
3.2.11.	NUMFAX: fax number of the recipient _____	12
3.2.12.	IDTERM: terminal identifier _____	12
3.2.13.	NUMTLX: telex number of the recipient _____	12
3.2.14.	DEBDEM: request for beginning function _____	12
3.2.15.	FINDEM: request for end function _____	13
3.2.16.	EXCDEM: request for Exception function _____	13
3.2.17.	TRADEM: request for API trace _____	13
3.2.18.	DUPDEM: duplication request _____	13
3.2.19.	APPDES: Receiving application name _____	13
3.2.20.	IDNRES: Network identifier _____	13
3.2.21.	LIGTBT: input/output line of TBT/400 _____	13
3.2.22.	DATDIF: Date of postponed message send _____	14
3.2.23.	HORDIF: Deferred transmission hour _____	14
3.2.24.	DATPER: Time-limitation date of message _____	14
3.2.25.	HORPER: Time-limitation hour of a message _____	14
3.2.26.	ACKDEM: TBT forwarding acknowledgment request _____	14
3.2.27.	SUPDEM: General suppression of files _____	14
3.2.28.	PUIDEM: Chaining mode _____	15
3.2.29.	IMPDEM: Printing requested _____	15
3.2.30.	BRKDEM: Break message requested _____	15
3.2.31.	COMUSR: User comment _____	15
3.2.32.	SPLSUP: Spoolfile suppression _____	15
3.2.33.	HAUPAG: Height of page _____	15
3.2.34.	PRIRES: Message priority _____	16
3.2.35.	AUTHOR: Author name _____	16
3.2.36.	OBJECT: Message subject _____	16
3.2.37.	ATTENT: For the attention of... _____	16
3.2.38.	AVIDIS: Broadcast notice request _____	16
3.2.39.	SYNDEM: Synchronous mode _____	16
3.2.40.	TYPTRN: Type of message forwarding _____	16
3.2.41.	GPHSIZ: Sheet size _____	16
3.2.42.	GPHCOL: Columns per line _____	17
3.2.43.	GPHSPA: Spacing _____	17
3.2.44.	GPHFON: Font used _____	17
3.2.45.	GPHLMA: Left border _____	18
3.2.46.	GPHRMA: Right border _____	18
3.2.47.	GPHDIS: End page note _____	18
3.2.48.	GPHDIF: Form for note _____	18

3.2.49.	GPHLOT: Logo at the top of the page _____	18
3.2.50.	GPHLOB: Logo at the bottom of the page _____	18
3.2.51.	GPHSIG: Insertion of a signature _____	19
3.2.52.	GPHNNF: Suppression of information field _____	19
3.2.53.	GPHLAN: Landscape mode _____	19
3.2.54.	GPHCOV: Isolated front page _____	19
3.2.55.	GPHNCO: Absence of front page _____	19
3.2.56.	GPHNHE: Absence of "header" _____	19
3.2.57.	GPHNFT: Absence of "footer" _____	19
3.2.58.	GPHFOR: Form used _____	20
3.2.59.	GPHTOP: Upper overlay margin _____	20
3.2.60.	GPHLEF: Left overlay margin _____	20
3.2.61.	GPHLOG : Logo _____	20
3.2.62.	GPHDBC: Invoicing code _____	20
3.2.63.	GPHCUR: User reference _____	20
3.2.64.	GPHMIN: Minicode _____	20
3.2.65.	GPHUSR: Graphnet subscriber _____	20
3.2.66.	GPHFIN: Fine resolution _____	20
3.2.67.	GPHISQ: Number of sequence _____	20
3.2.68.	GPHCV: Bilingual Header _____	21
3.2.69.	GPHSZA: Page adjustment _____	21

1. Architecture

1.1. APIs

Application programs can interface TBT/400. APIs are accessible at two levels.

1.1.1. Command APIs

This is the first level of usable APIs. These APIs are the simplest way to interface TBT/400. They are directly usable in CLP programming. All commands are located in the *IPLSP* library (*IPLSP* is the default installation name and can be changed). ¹

IPSNDGRAPH This Api is used for sending a message to network ². With this command you can send an OS/400 file (physical or spool) to a network target (fax or telex).

IPSRCVTBT This Api is used for receiving later network acknowledgments

1.1.2. High language APIs

This second level of interface allows all **RPG**, **C** or **COBOL** processes to access, using a single syntax, all functionality's of TBT/400. The possibilities are the same, but usage is some more complex than **Command APIs** (writing, compiling and debugging...). In fact, Command APIs use themselves this second level. In this way of programming, user programs have to supply information in communication elements provided in structures or copy clauses. The used fields in these copy clauses use the same names than keywords in commands.

1.2. Usable files

The message text to be sent can be found in OS/400 physical files or OS/400 spoolfiles.

1.2.1. Physical files

TBT/400 can send OS/400 physical files. These files are qualified by:

- **Library Name** ³ *Keyword OBLIB*
- **File Name** ⁴ *Keyword OBJFIL*
- **Member Name** ⁵ *Keyword OBJMBR*

Physical files can also be sourcefiles , but it is not mandatory. In fact, TBT/400 can also send logical and join files.

1.2.2. Spoolfiles

TBT/400 can send OS/400 spoolfiles. These files are qualified by :

- Spool creating job ⁶ *Keyword SPLJOB*
- Spool number ⁷ *Keyword SPLNUM*

¹ To write and compile a applicative program, you must input IPLSP in the library list (ADDLIBLE IPLSP) or in the commands of qualification of TBT/400 (IPLSP/IPSNDGRAPH).

² Obligatory API.

³ Optional field, the value default is library.

⁴ Obligatory field

⁵ Optional field ; by default the first member is used

⁶ By default current job

- Spool name⁸ *Keyword SPLNAM*

1.3. Network target

You can define the network target either with logical address, or with physical address. These two ways are exclusive. TBT/400 can also search address within message text.

1.3.1. Logical address

TBT/400 has a directory. This directory is interactively updated with the **IPS** command⁹. A network target can be defined in this directory with a logical name, and a physical address (fax or telex). In the application programs, you must use the **NOMLOG**¹⁰ keyword.

```
IPLSP/IPSNDGRAPH OBJLIB (IPLSP) OBJFIL (IPSSAMPLES)
                   OBJMBR (IPZIGBAN) NOMLOG (IPLS) 11
```

1.3.2. Physical address

The application program can directly give a telex or fax address :

- For telex use **NUMTLX**¹², and optionally **IDTERM**¹³ keywords
- For fax use **NUMFAX**¹⁴ keywords

```
IPLSP/IPSNDGRAPH OBJLIB (IPLSP) OBJFIL (IPSSAMPLES)
                   OBJMBR (IPZIGBAN) NUMFAX (33+130157091)
                   15
```

1.3.3. Search address

TBT/400 can search for the target address within message text (either from physical files or from spoolfiles). This addressing mode is used by gateways and defined for them. It can be also directly used by application programs.

It is mandatory to find the following syntax within the **32000** first characters of the file one of the following items:

- (**D : FAX =**, [**PRI =**]) Transmit to **Fax**
- (**D : TLX =**, [**IDT =**] , [**PRI =**]) Transmit to **Telex**
- (**D : LOG =**, [**PRI=.....**]) Transmit to **Partner**
- (**D : EDI =**, [**PRI=.....**]) Transmit to **EDI**

FAX= Fax address with syntax **FAX=130157091** for a national fax

⁷ *By default, the last and only spoolfile*

⁸ *By default TBT400*

⁹ *Entry point of TBT/400 menus.*

¹⁰ *Defines the logical name of the partner.*

¹¹ *This command send a message to **IPLS** company. The text is read into **IPZIGBAN** member of **IPSSAMPLES** file of the **IPLSP** library. **IPLS** recipient is defined in **TBT/400** directory.*

¹² *Defines a telex number.*

¹³ *It defines the telex code awaited*

¹⁴ *Defines a fax number.*

¹⁵ *This command send a message to the **IPLS** fax. The text is read into **IPZIGBAN** member of **IPSSAMPLES** file of the **IPLSP** library*

FAX=33+130157091 for a international fax
TLX= Telex address with syntax **TLX=123456** for a national telex
TLX=42+123456 for a international telex
IDT= Telex answerback
LOG= Partner's name defined in TBT/400 directory with type \$\$\$\$ANNUAI ¹⁶
EDI= Partner's name defined in TBT/400 directories ¹⁷
PRI= N,U or H Normal, urgent

1.4. Other transmitting functionality's

1.4.1. Spool files gateways

An Output queue (IP\$\$\$\$PLF within the *IPLSC* ¹⁸ library) can be periodically scanned for outgoing messages. Each present spoolfile is sent over network through search address mechanism. See 'Spoolfiles gateway' for more information. This method is not the best one for message tracking. Application programs have a lot of difficulties to follow the various acknowledgments given by TBT/400 and networks. It seems to be preferable to use this mechanism with already existent programs with no modifications wished or possible.

1.4.2. Office vision Gateway

TBT/400 offers an interface with OV/400. See 'Office gateway' for more information. The search mechanism is also used in this gateway.

¹⁶ *Telex or fax library*

¹⁷ *EDI libraries*

¹⁸ *The name of this library can be changed during the installation.*

2. Messages transmitting and tracking

2.1. Preliminary

With TBT/400, it's very easy to transmit messages over the network. TBT/400 archives all transmissions. It is so possible to supervise the traffic, and have immediate access to the list of erroneous transfers via the supervisory menus, through the **IPS** command.

TBT/400 can also inform application programs from network acknowledgments. This mechanism, mandatory asynchronous, is a little more complex, and some concepts must be explained.

This chapter must be used only for acknowledgments process by application programs, in addition to the standard TBT/400 tracking.

2.2. Application

This is the main notion for TBT/400. It is a logical entity representing the access point to TBT/400. It is your identification, your 'handle' to TBT/400. For TBT/400, there is always a sending application, and a receiving application. Some technical applications are delivered with TBT/400.

- **\$INTERNA** means internal application, it's the default for sending application.
- **\$EXTERNA** means external application (in fact the network driver), it's the default for receiving application.

For message tracking, you must define a new application ('MYAPPLI' for sample), define it in the 'Applications configuration' panel, and use it in the sending application for the **IPSNDRGRAPH** command. You have also to give a 'user key'. This key is a literal, which will appear in TBT/400 supervision, and will be later be transmitted to the application program which will process acknowledgments (a correlation key). At last, you have to require for TBT/400 acknowledgments.

```
IPLSP/IPSNDRGRAPH APPEME (MONAPPLI) 19 KEYUSR (MACLE) 20
ACKDEM (O) 21
OBJLIB (IPLSP) OBJFIL (IPSSAMPLES)
OBJMBR (IPZIGBAN) NUMFAX (33+130157091)
```

19 *Transmission application. It must be defined by the menu of the applications configuration*

20 *User key. It is a correlation key for the next treatment of the acknowledgements.*

21 *Acknowledgement request : O = always*

2.3. Queue

For each Application definition, **TBT/400** associates three logical queues: a message queue, a transmission acknowledgment queue, and a rejection queue.

Message and Rejection queues are used for incoming messages, and are not usable here.

Acknowledgment is the **TBT/400** where are put networks acknowledgments. This is the good one for message tracking. The default name is the application name prefixed by the literal 'A'. We have to define it through the 'Queues configuration' panel. To this queue must be associated a processing program (the application program which will be called for each acknowledgment), and an operating mode (the best one is the mode 2, which means queue processed in 'real time').

In our sample, we have to create the queue "AMONAPPLI", in mode **2**, associated to the program "MONPROG" within library "MALIB".

2.4. Processing program

By default, the processing program is 'IPSPADUMMY' within library *IPLSP*. This program is given in source form in file *IPSSAMPLES* within library *IPLSP*.

The simplest way to create a processing program, is to:

- duplicate the source member **IPSPADUMMY** from file *IPSSAMPLES* within library *IPLSP* to source file '*MYSOURCE*' within library '*MYLIB*', with the name of '*MYPROG*',
- to compile it without modifications (input *IPLSP* in library list: **ADDLIBLE *IPLSP***),
 - and after creating the object program, define the queue (a control is made for the existence of the program when configuring the queue).

1. Define the application
2. Create the program (without modifications)
3. Define the queue
4. Alter the program to satisfy your requirements

2.5. Processing program Structure

```

PGM
DCL          VAR (&DEBDEM)          TYPE (*CHAR)      LEN (1)      VALUE (0)
DCL          VAR (&RTNCDP)          TYPE (*DEC)       LEN (11)
DCL          VAR (&KEYTBT)          TYPE (*CHAR)      LEN (16)
DCL          VAR (&KEYUSR)          TYPE (*CHAR)      LEN (16)
DCL          VAR (&COMUSR)          TYPE (*CHAR)      LEN (128)
DCL          VAR (&ACKTBT)          TYPE (*CHAR)      LEN (2)
DCL          VAR (&LIBTBT)          TYPE (*CHAR)      LEN (128)

MONMSG      MSGID (CPF0000)        EXEC (GOTO        CMDLBL (CPF0000) )

LOOP: IPSRCVTBT  FNCDEM (R)  DEBDEM (&DEBDEM)  FINDEM (C)      + 22
      EXCDDEM (N)  RTNCDP (&RTNCDP)  KEYTBT (&KEYTBT)  +
      KEYUSR (&KEYUSR)  ACKTBT (&ACKTBT)      +
      LIBTBT (&LIBTBT)                                COMUSR (&COMUSR)

CHGVAR      VAR (&DEBDEM)                                VALUE (N)

IF          COND (&RTNCDP      *NE          0          )      THEN (DO)
GOTO        CMDLBL (ENDPGM) 23
ENDDO

      SNDPGMMSG  MSG ('KEYUSR='          *CAT          &KEYUSR) 24
      SNDPGMMSG  MSG ('COMUSR='          *CAT          &COMUSR)
      SNDPGMMSG  MSG ('ACKTBT='          *CAT          &ACKTBT) 25
      SNDPGMMSG  MSG ('LIBTBT='          *CAT          &LIBTBT)

IPSRCVTBT  FNCDEM (P)  DEBDEM (N)  FINDEM (C)  EXCDDEM (O)  +
      KEYTBT (&KEYTBT) 26
GOTO        CMDLBL (LOOP) 27

CPF0000:  SNDPGMMSG  MSGID (CPF9898)  MSGF (QSYS/QCPFMSG)      +
      MSGDTA ('ERROR IN COMMAND')      +
      MSGTYPE (*ESCAPE)

ENDPGM:  ENDPGM

```

22 The first acknowledgement is asked to **TBT/400**

23 The last event has been read: End of program

24 The application program can treat the values provided by **TBT/400**. **KEYUSR** was provided by the transmission program and constitutes the correlation key. (**COMUSR** is in the same case). Considerable other values are provided by this command (acceptance date, distribution date, etc..).

25 Network acknowledgement '**PC**' means network acceptance, ' ' (**SPACE**) means positive distribution, the other values mean negative distribution.

26 Informs **TBT/400** that the event was treated, and removes it from **TBT/400**.

27 It loops to treat the following event

2.6. Diagram

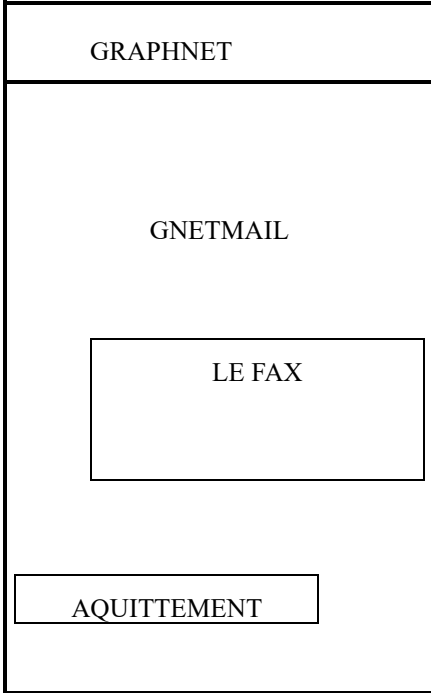
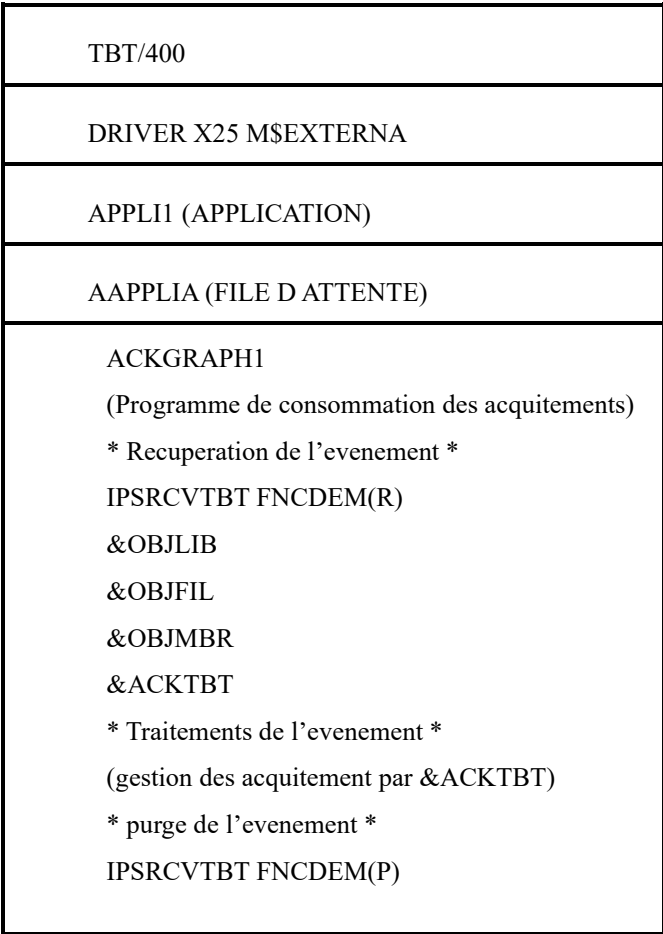
Applicatif qui fait une requête d'émission

```

Programme d'emission EMIGRAPHTBT
*****

PGM
PARM(&NUMFAX &OBJLIB &OBJFIL &OBJMBR.....)
ILS410P/IPSNDGRAPH APPEME(APPLI1) KEYUSR('MAILING
+      NOEL') OBJLIB(&OBJLIB) OBJFIL(&OBJFIL) +
OBJMBR(&OBJMBR) NUMFAX(&NUMFAX) ACKDEM('O')

ENDPGM
*****
    
```



3. IPSNDGRAPH command

3.1. Syntax

IPSNDGRAPH APPEME(\$INTERNA) KEYUSR() OBJLIB(*LIBL) OBJFIL() OBJMBR(*FIRST) SPLNAM(TBT400) SPLJOB() SPLNUM(0) NOMLOG() REFMSG() NUMFAX() IDTERM() NUMTLX() DEBDEM(O) FINDEM(O) EXCDEM(O) TRADEM(O) DUPDEM(N) APPDES(\$EXTERNA) IDNRES(BAL0000001) LIGTBT() DATDIF() HORDIF() DATPER() HORPER() ACKDEM() SUPDEM() PUIDEM() IMPDEM() BRKDEM() COMUSR() SPLSUP() HAUPAG() PRIRES() AUTHOR() OBJECT() ATTENT() AVIDIS() SYNDEM() TYPTRN() GPHSIZ() GPHCOL() GPHSPA() GPHFON() GPHLMA() GPHRMA() GPHDIS() GPHDIF() GPHLOT() GPHLOB() GPHSIG() GPHNNF() GPHLAN() GPHCOV() GPHNCO() GPHNHE() GPHNFT() GPHFOR() GPHTOP() GPHLEF() GPHLOG() GPHDBC() GPHCUR() GPHMIN() GPHUSR() GPHFIN() GPHCV() GPHSZA(O nnn mm)

3.2. Keywords

3.2.1. APPEME: Transmitting application name

It specifies the name of application that transmits the message. In case of transmission supervision, notices will be available for this application name.

The default value is \$INTERNA.

3.2.2. KEYUSR : use key of message

It specifies the message identification for transmitting application, to act as correlation to applicative program.

This field is considered by TBT/400 as a narrative. However, it appears in "list" of the supervision type menus, and can form the subject of a selective research in this list. Then, its use is recommended.

For a program treating of the entering events, this heading can be developed.

3.2.3. OBJLIB: name of the file library

It specifies the name of the file library in transmission as in reception. For spoolfiles, this field is ignored.

Default value is *LIBL.

3.2.4. OBJFIL: file name

It specifies the file name in transmission or in reception.

The special value *DUMMY represents a false request of transmission. It is reserved for the scanning mode (clearing box).

The special value *SPLF means spoolfile is to be sent.

3.2.5. OBJMBR: member name

It specifies the member name in transmission or in reception. For spoolfiles, this field is ignored.

Default value is *FIRST.

3.2.6. SPLNAM: Spoolfile name

It specifies the spoolfile name to be sent on the network. The default value is TBT400.

To transmit a spoolfile, OBJFIL item must contain the special value "*SPLF"; SPLNAM, SPLJOB, SPLNUM items must be fed.

3.2.7. SPLJOB: job name

It specifies the job name that created the spoolfile to be sent on network. The default value is “*”, which means current job.

To transmit a spoolfile, **OBJFIL** item must contain the special value “*SPLF”; **SPLNAM**, **SPLJOB**, **SPLNUM** items must be fed.

3.2.8. SPLNUM: Spoolfile name

It specifies the number of the spoolfile to be sent on the network. **0** value specifies that the spoolfile is single. **-1** value requests transmission of the last spoolfile.

To transmit a spoolfile, **OBJFIL** item must contain special value *SPLF; **SPLNAM**, **SPLJOB**, **SPLNUM** items must be fed.

3.2.9. NOMLOG: logical name of the correspondent

Logical name or alias setting a physical address described in your directory.

If you wish to use an only numerical logical name, it is necessary to input a name which is formed **Unnnn**, with **nnnn** numerical. Input control will be done only on **nnnn**.

This use must be exclusive from **NUMFAX** or **NUMTLX** fields. The special value *SEARCH requires that **TBT/400** must search the network address within the text of the message.

3.2.10. REFMSG: message reference

It specifies reference of message, peculiar to user.

3.2.11. NUMFAX: fax number of the recipient

It specifies the fax number of the recipient, under the form:

telephonic country code + national number (you must code the +).

For national transmission, you can omit the national code and the character +.

- 33+130157091 - Fax **IPLS**
- 130157091 - Fax **IPLS** (Paris)

As part of the new plan of numbering of **France Telecom**, you mustn't dial the 0 included in the number with ten digits.

3.2.12. IDTERM: terminal identifier

It specifies the dialing code of the terminal of the message recipient

3.2.13. NUMTLX: telex number of the recipient

Specifies the telex number of the receiver under the form:

telegraphic country code + national number (you must input the +).

For national, you can omit the national code and the character +.

3.2.14. DEBDEM: request for beginning function

This parameter is interesting only for **APIs** programming.

It is used to ask the execution of the **APIs** beginning process (**IPSSGDEB**) before the execution of another function specified in **FNCDEM**.

The default value is **Non**.

TBT/400 builds an environment to satisfy an **API** request if this environment is not already present. When this parameter has the 'Yes' value, **TBT/400** is instructed to systematically build a new environment for this request. This is a good precaution, and is the default value. (The **No** value must be reserved for batch sending with many messages, to make a 'resident' environment and save **CPU** process time).

3.2.15. FINDEM: request for end function

After process of an **API** request, **TBT/400** can break the build environment. This parameter has three possible values: **Yes**, **No**, or **Conditional** (**Yes** if error in **API**). The **Yes** value is the default value. (The other values must be reserved for batch transmitting with many messages, to make a 'resident' environment and save **CPU** process time).

3.2.16. EXCDEM: request for Exception function

At the end of processing an **API** request, **TBT/400** sends always an **OS/400** ending message if the process is accepted. If an error is encountered, **TBT/400** can send an **OS/400** exception message (**IPS9999**) to inform the transmitter of the incident. This exception message is sent if **EXCDEM** has the **Yes** value (which is the default). In the other case, a diagnostic message will be sent. If message mode is chosen (implicitly or explicitly), the application program must monitor the exception (**MONMSG IPS9999**)

3.2.17. TRADEM: request for API trace

Specifies if application program wants to trace its **TBT/400** **API** calls.

This parameter can be overridden by **TBT/400** configuration menu to have a trace without compilation. Avoid **3** value which is **CPU** consuming.

- '0' = no trace (default value),
- '1' = level 1 trace requested (passage runs),
- '2' = level 2 trace requested (passage runs and blocks snaps).

3.2.18. DUPDEM: duplication request

Specifies whether the application wants to duplicate the file during deposit.

In **N** case, the process forbids itself to modify the file between the transmission request and the real broadcast on the network.

In **Y** case, a duplication in **TBT/400** operation library (**IPLSE**) is effected, releasing then the original file. The duplication is implicit if the original library is **QTEMP**. The duplication allows also historisation of messages text. For spoolfiles, this value has no meaning.

- **N** = no duplication (default value)
- **O** = duplication into **IPLSE**

3.2.19. APPDES: Receiving application name

Specifies the name of application that will receive the message. The default value **\$EXTERNA** means transmission on the network.

3.2.20. IDNRES: Network identifier

This name must be defined in 'Networks identifiers configuration' and defines the used mailbox. (**TBT/400** can use several mailboxes). The default value is **BAL0000000001**.

3.2.21. LIGTBT: input/output line of TBT/400

Specifies logical name of input/output lines of the used network. This name is defined in the networks identifiers. By default **TBT/400** use all **X25** lines defined in the configuration lines. It's possible to require the use of a specific line.

3.2.22. DATDIF: Date of postponed message send

It specifies date of postponed message transmission, which size is AAAAMMJJ. The default value is immediate start. This item is linked to **HORDIF**.

In sizing, +nn means current date + nn days.

The **DATPER** + **HORPER** couple can't be lower to **DATDIF** + **HORDIF**.

3.2.23. HORDIF: Deferred transmission hour

It specifies deferred transmission hour of message, in the format HHMMSSCC.

The default value is immediately. This item is bound to **DATDIF**.

In seizing, +nn means current hour + nn minutes

3.2.24. DATPER: Time-limitation date of message

Specifies time-limitation date of message, in AAAAMMJJ format. Default value is generated from the number of minutes specified in **PERTLX** item for telex type messages, from the number of minutes specified in **PERFAX** item for fax type messages, or from the number of days specified in **PEREMJ** item in other cases. These three items are in system timers configuration menu (**PEREMJ** item). This item is linked to **HORPER**.

In sizing, +nn means current date + nn days.

3.2.25. HORPER: Time-limitation hour of a message

Specifies time-limitation hour of the message, in the format HHMMSSCC. Default value is current hour. System configuration (**PEREMJ**) enables only to define a time-limitation in days. This item is bound to **DATPER**.

In seizing, +nn means current hour + nn minutes.

The couple **DATPER** + **HORPER** can't be inferior to **DATDIF** + **HORDIF**.

3.2.26. ACKDEM: TBT forwarding acknowledgment request

Specifies if the transmitting process wishes to receive a forwarding acknowledgment from **TBT/400**.

- Y = Yes in any case,
- N = No in any case (default value),
- C = Conditional, only in error case.

3.2.27. SUPDEM: General suppression of files

Specifies the general rule of file suppression :

- Y = immediate suppression after transmission,
- N = no suppression, then "housework" will have to be done manually,
- H = suppression during time limitation of log.
- C = conditional, that means, if valid transmission.

Member suppression is done. If there is no more members, the file is suppressed too. If you don't wish it, create in the file a member with the same name than file. Then, the automatism will be stop.

Members names must be dynamically created (that means by **TBT/400**) in order for suppression to be carried out by use of this parameter.

IPSCRTMEMB is at disposal to create file members whose name is compatible with **TBT/400** syntactic rules, and then participating to purges at time limitation on objects of **IPLSE (LIBEXP)** and **IPLSM (LIBMES)** libraries.

3.2.28. PUIDEM: Chaining mode

Enables the use of chaining function for message transmission.

- **Y** = if several messages for the same receiver, if need be chaining of messages during the same communication (default value),
- **N** = no chaining, message will be transmitted separately.

3.2.29. IMPDEM: Printing requested

Enables to demand a printing after message processing. Printing will be done on the printer associated with the Job having which has effecting the transmitting request.

The default value depends on receiver type; **IMPDET** for telex, **IMPDEF** for fax, **IMPDEA** for all others.

- **Y** = systematic printing,
- **N** = no printing,
- **C** = conditional printing (if error),
- **B** = printing if good forwarding.

3.2.30. BRKDEM: Break message requested

Enables to demand a break message after message processing. This **OS/400** message will be sent to the message queue of the original **OS/400** user.

The default value depends on receiver type; **BRKDET** for telex, **BRKDEF** for fax, **BRKDEA** for all others.

- **Y** = systematic printing,
- **N** = no printing,
- **C** = conditional printing (if error),
- **B** = printing if good forwarding.

3.2.31. COMUSR: User comment

Contains a user comment available for a possible use by application program. This field is not sent over the network.

It is possible, for a program which treats entering events, to develop this heading.

3.2.32. SPLSUP: Spoolfile suppression

Specifies suppression rule of spoolfile after deposit a transmission request in the **TBT/400** queue.

- **Y** = immediate suppression after deposit
- **N** = no suppression, then the housework will have to be done manually,
- **C** = conditional, that means, if deposit is accepted by **TBT/400**

3.2.33. HAUPAG: Height of page

Contains, for a fax, the number of lines contained in one page.

TBT/400 can, in **POSTSCRIPT** transmission, lessen this height.

TBT/400, in spoolfile transmission, sets it.

The height of a standard page of an **AS/400 "printer file"** is 66 lines. If network adds a "**header**" or a "**trailer**", the height of the page really printed can reach 68 lines; in fact the capacity of a standard A4 page is easily over topped. It's then better to inhibit, if possible, this add-ons, or to use maximal **62 lines "printer files"**.

You can ask for a page break anywhere in the text while informing, in the same text, the command `/PAGE '`.

3.2.34. PRIRES: Message priority

Specifies message priority managed by network.

- U = urgent,
- N = normal (default value),
- H = off peak hours of network.

According to protocols, this field can really be emitted or can only be with documentary type. It is only used on Megafax+ (heading OP:)

3.2.35. AUTHOR: Author name

Indicates author's name of message. On **GraphPAK**, it feeds the heading **/FROM**, on **Megafax+** the heading **FR:**.

3.2.36. OBJECT: Message subject

Specifies the subject of the message, i.e. enables to indicate succinctly the content of the message. On **GraphPAK**, it feeds the heading **/RE**, on **Megafax+** it is unused.

3.2.37. ATTENT: For the attention of...

Specifies the attention that designates the aimed user of the message. On **GraphPAK**, it feeds the heading **/ATTN**, on **Megafax+** heading **AT:**.

3.2.38. AVIDIS: Broadcast notice request

Indicates whether the network broadcast notice is requested.

- Y = broadcast notice for all messages,
- N = notice only for impossible deliveries.

In the current state, **TBT/400** enforces this field to the Y value.

3.2.39. SYNDEM: Synchronous mode

Specifies a transmission request with a synchronous or asynchronous return code. This feature isn't used yet.

- Y = Synchronous mode requested
- N = No synchronous mode : default

3.2.40. TYPTRN: Type of message forwarding

Determines a forwarding mode of message.

For Fax transmission, transparent mode is in fact a "Postscript" management of message. It improves presentation, and enables in particular to make "horizontal" fax. This mode is forced if record length is upper than 170.

- Y = transmission in transparent mode (Binary)
- N = transmission in normal mode or text mode (default value).

3.2.41. GPHSIZ: Sheet size

This item specifies size of sheet used. It is used by **GraphPAK** and it is sent such as.

/LETT	- 8,5 x 11 inches	(default in USA)
/A4	- A4	(default in France)
/LEGL	- 8,5 x 14 inches	

3.2.42. GPHCOL: Columns per line

This item specifies the number of columns per line. It is used by **GraphPAK**. **TBT/400** adjust this item to the immediately higher value. For spoolfiles, the value is set by **TBT/400**.

Three values are possible in vertical mode:

80

96

132

In landscape mode, possible values are:

90

100

145

170

171 à 250

3.2.43. GPHSPA: Spacing

This item specifies spacing between each line (from 0 to 3) It is used by **GraphPAK** and feeds the option **/SP =**.

3.2.44. GPHFON: Font used

This item specifies the font used. It is used by **GraphPAK** and feeds the option **/FONT =**. "Complete Ebcidic" fonts are more complete. "Wide PC" fonts lose some special characters EBCDIC.

The others lose all the accentuated characters. The number of lines is the maximum number, without "disclaimer". Use if possible the font ***, and for special cases the font **, or *.

Vertical mode:

AEC	- ALHEBEC	132 characters	75 lines		
AES	- ALHEBES	96 characters	75 lines		
APC	- ALHEBPC	132 characters	75 lines		
APS	- ALHEBPS	80 characters	75 lines		
C1618	- ATI1618	96 characters	57 lines	Wide PC	**
ECT	- EPECSCAL	132 characters	89 lines		
EEC	- EPSONEC	132 characters	75 lines		
EES	- EPSONES	96 characters	75 lines		
EPC	- EPSONPC	132 characters	75 lines		
EPS	- EPSONPS	80 characters	75 lines		
EST	- EPSSCAL	96 characters	89 lines		
IES	- IBMPCPS	96 characters	75 lines	Wide PC	
IPC	- IBMPCPS	132 characters	75 lines	Wide PC	
IPS	- IBMPCPS	80 characters	75 lines	Wide PC	
IST	- IBMPCPS	80 characters	89 lines	Wide PC	**
I92	- IBM0912	132 characters	89 lines	Wide PC	**
I94	- IBM0914	132 characters	75 lines	Wide PC	
PCT	- EPPCSCAL	132 characters	89 lines		
PST	- EPPSSCAL	80 characters	89 lines		
SPC	- SEPSONPC	132 characters	108 lines	Wide PC	**
273EC	- 273EC	132 characters	75 lines	Complete Ebcidic	***
273ES	- 273ES	96 characters	75 lines	Complete Ebcidic	***
273PS	- 273PS	80 characters	75 lines	Complete Ebcidic	***
297EC	- 297EC	132 characters	75 lines	Complete Ebcidic	
297ES	- 297ES	96 characters	75 lines	Complete Ebcidic	
297PS	- 297PS	80 characters	75 lines	Complete Ebcidic	

Landscape mode:

ECT	- LS25EEC	170 characters 60 lines		*
EEC	- LS28EEC	170 characters 53 lines		*
EES	- LS28EES	100 characters 53 lines		
EPC	- LS28EPC	145 characters 53 lines		
EPS	- LS28EPS	90 characters 53 lines		
EST	- LS25EES	100 characters 60 lines		*
IES	- LS28IES	100 characters 53 lines	Wide PC	**
IPC	- LS28IPC	145 characters 53 lines	Wide PC	**
IPS	- LS28IPS	90 characters 53 lines	Wide PC	**
IST	- LS25IPS	90 characters 60 lines	Wide PC	**
PCT	- LS25EPC	145 characters 60 lines		*
PST	- LS25EPS	90 characters 60 lines		*

3.2.45. GPHLMA: Left border

This item specifies the left border in 1/10 of inch (1 inch = approximately 25 millimeters). This field can take values from 0 to 10. It is used by **GraphPAK** and conditions the transmission of the command **/LMARG =**.

In all cases, TBT/400 adjusts this value with the one of heading **GPHRMA**.

3.2.46. GPHRMA: Right border

This item specifies the right border in 1/10 of inch (1 inch = approximately 25 millimeters). This field can take values from 0 to 10. It is used by **GraphPAK** and conditions the transmission of the command **/RMARG =**.

In all cases, TBT/400 adjusts this value with the one of heading **GPHLMA**.

3.2.47. GPHDIS: End page note

This item specifies a printed note at each end of page (above the footer). In portrait mode, the font used is specified by **GPHDIF** item. It is used by **GraphPAK** and conditions the transmission of the command **/DISCLMR={.....}**.

3.2.48. GPHDIF: Form for note

This item specifies the form used by end page note (**GPHDIS**). It is used by **GraphPAK**, and is useful only for "landscape" mode.

IBMPCPS 80 characters per ligne (/DISCLSTD command is sent)

EPSONPC 132 characters per ligne (/DISCLCMP command is sent)

3.2.49. GPHLOT: Logo at the top of the page

This item specifies the use of a logo at the top of a page, **if there is no front page**. It is used by **GraphPAK** and conditions the transmission of the command **/LOGT**.

'Y' - Yes

'N' - No (default)

3.2.50. GPHLOB: Logo at the bottom of the page

This item specifies the use of a logo at the bottom of the page, **if there is no front page**. It is used by **GraphPAK** and conditions the transmission of the command **/LOGB**.

'Y' - Yes

'N' - No (default)

3.2.51. GPHSIG: Insertion of a signature

This item specifies insertion of a signature at the end of a message. It is used by **GraphPAK** and conditions the transmission of the command **/SIGN**.

'Y' - Yes

'N' - No (default)

3.2.52. GPHNNF: Suppression of information field

This item specifies suppression of information field on every pages. It is used by **GraphPAK** and conditions the transmission of the command **/NNFO**.

'Y' - Yes

'N' - No (default)

3.2.53. GPHLAN: Landscape mode

This item specifies the use of the landscape mode. TBT/400 rocks automatically in landscape mode beyond 132 columns.

It is used by **GraphPAK** and conditions the transmission of the command **/LAND**.

'Y' - Yes

'N' - No (default)

3.2.54. GPHCOV: Isolated front page

This item specifies the use of an isolated front page.

It is used by **GraphPAK** and conditions the transmission of the command **/COV**. This option is forced if the height of the page is imposed (**GPHLIG**).

'Y' - Yes

'N' - No (default)

3.2.55. GPHNCO: Absence of front page

This item specifies absence of front page. It is used by **GraphPAK** and conditions the transmission of the command **/NCOV**.

'Y' - Yes

'N' - No (default)

3.2.56. GPHNHE: Absence of "header"

This item specifies absence of header automatically generated. It is used by **GraphPAK** and conditions the transmission of the command **/NHED**.

'Y' - Yes

'N' - No (default)

3.2.57. GPHNFT: Absence of "footer"

This item specifies absence of a footer automatically generated. It is used by **GraphPAK** and conditions the transmission of the command **/NFTR**.

'Y' - Yes

'N' - No (default)

3.2.58. GPHFOR: Form used

This item specifies the form used (use of overlay). It is used by **GraphPAK** and conditions the transmission of the command **/FORM={ option1 = parameter 1, ...}**.

The options of the command are:

ID=xxxxx which is the number of form in 5 positions
 SP=n which is the spacing requested in heading GPHSPA
 FONT=nnn which is the font requested in heading GPHFON
 LN=xx which is the number of lines per page, requested in heading HAUPAG
 C=xxx which is the number of characters per line, requested in heading GPHCOL
 SET=ALL which asks for the form on all the pages of the message
 LEFTC=n which creates a left margin of N characters, asked in heading GPHLEF
 TOP=n which creates a high margin of N pixels, asked in heading GPHTOP

3.2.59. GPHTOP: Upper overlay margin

This heading specifies the high margin in lines fax (pixels) for the transmissions with overlay. It is used by **GraphPAK**. It must not exceed **999**. It conditions the transmission of the **TOP parameter** of the command/**FORM**.

3.2.60. GPHLEF: Left overlay margin

This heading specifies the left margin in characters for the transmissions with overlay. It is used by **GraphPAK**. It must not exceed **99**. It conditions the transmission of **LEFTC parameter** of the command/**FORM**.

3.2.61. GPHLOG : Logo

This item specifies the use of a logo different from the one associated to **Graphnet** mailbox. It is under **LG(xxxx)** parameter of the command/**SVC**.

3.2.62. GPHDBC: Invoicing code

This item specifies the use of an invoicing code (billing code). Its use is controlled by **GPHFAC** item. It is under **DB** parameter (invoicing code) of the command/**SVC**.

3.2.63. GPHCUR: User reference

This item specifies a user reference. The default value is given by the **REFMSG** item. It is under **CR** parameter (customer reference) of the command/**SVC**.

3.2.64. GPHMIN: Minicode

This item specifies the use of a Minicode.

3.2.65. GPHUSR: Graphnet subscriber

This item specifies the use of a Graphnet subscriber using Gnetmail.

3.2.66. GPHFIN: Fine resolution

This item requires the use of the fine resolution (200x100 points per inch). The default resolution is 100x100 points per inch. It is used by **GraphPAK** and conditions the transmission of the command **/FINERES**.

'Y' - yes
 'N' - No (default)

3.2.67. GPHISQ: Number of sequence

This item specifies a number of applicative sequence to the message. It is used by Gnetmail. This number must be from 1 to 9999 characters. It conditions the transmission of the command **/ISQ**.

3.2.68. GPHCV: Bilingual Header

This item specifies the use of bilingual header.

It is used by **GraphPAK** and conditions the transmission of the command **/CV**

'Y' - Yes

'N' - No (default)

3.2.69. GPHSZA: Page adjustment

This item requires the adjustment of the page for the small faxes **GPHSZA**=(' O' 60 20).

It is used by **GraphPAK**. The first parameter means 'Y' or 'N'. The second means the maximum size (60 by default), the third means the size of the header (20 by default).

In vertical mode, if the adjustment is required, **TBT/400** will calculate the minimal number of lines for the fax: number of recordings + size of the prefix (heading **GPHSZH**). If this number is higher than the maximum size (**GPHSZM**), the standard size will be used (**GPHSIZ**).

At the opposite, the **/LN** command = will be used, adjusting at least the reception fax.

IPLS

176, Bureaux de la Colline
92210 Saint-Cloud
FRANCE

Téléphone +33 (0)1 80 41 00 60

Site de l'éditeur : www.lpls.fr

E-mail : lpls@lpls.fr

E-mail : Commercial@lpls.fr

E-mail : Technic@lpls.fr

Site du progiciel : www.tbt400.com