

1.	Utilisation des API	3
1.1.	Concepts	3
1.2.	Mode d'utilisation	4
1.3.	Logique d'utilisation	4
1.4.	Schéma de fonctionnement	5
2.	BLOCS DE COMMUNICATION	6
2.1.	Logique d'utilisation	6
2.2.	Bloc P0 de dialogue	7
2.3.	Bloc P1 de dialogue	11
3.	Exemples de programmes	20
3.1.	Programme d'émission Telex X25	20
3.2.	Programmes Atlas 400 ou Graphnet	24
3.2.1.	Programme d'émission de Fax	24
3.2.2.	Programme d'émission de message	28
3.2.3.	Programme de réception de message	30
3.2.4.	Programme de réception d'acquittement	32
3.2.5.	Programme de réception par CL	34
4.	Annexe : Blocs de communication P0 et P1	39
4.1.	Structure C (IPSILCAP)	39
5.	ANNEXE : INDEXE DES NOMS DE CHAMPS DES BLOCS DE COMMUNICATION	56
6.	Annexe : Statut du message	58
6.1.	En émission	58
6.2.	En réception	58
7.	Annexe : Numérotation Téléphonique, Fax et Téléx	59
7.1.	Numérotation Téléphonique ou Fax	59
7.2.	Numérotation Téléx	59
7.2.1.	Numéro Téléx	59
7.2.2.	Indicatif Téléx	59
7.2.3.	Codes des incidents Téléx	60

1. Utilisation des API

1.1. Concepts

TBT/400 assure le transfert des messages ou fichiers entre des destinataires de divers types, et peut donc être utilisé pour faire de l'EDI.

La couche **API** de **TBT/400** permet à un programme utilisateur d'envoyer à une autre application, ou à un destinataire de type Terminal (télécopie, télex, etc...) ou ordinateur, des messages ou fichiers.

Il peut de la même manière recevoir des messages ou fichiers pour les mettre à la disposition de vos applicatifs.

Vos applicatifs sont identifiables par **TBT/400** sous la forme (au travers) d'un nom logique ou nom d'application (8 caractères). Votre correspondant est lui aussi identifiable par un nom d'application qui est obligatoirement \$EXTERNA pour le driver non serveur (ou \$EXTERNB ou \$EXTERNC selon le réseau; le nom d'application dépendant du driver utilisé) dans le cas d'une communication vers un réseau extérieur. L'identification réelle de votre correspondant s'exécutera par l'identification du réseau d'accès et son adresse réelle (conforme au réseau duquel il dépend).

Par exemple, si votre destinataire est une télécopie: il s'identifiera par le réseau d'accès, le Réseau à Valeur Ajoutée ATLAS 400 par exemple, puis par le numéro du fax 12345678. Et, l'application destinatrice (représentant tous vos correspondants joignables par réseau externe, donc votre destinataire y compris) du message qui lui est destiné sera \$EXTERNA

Chaque application (APPLICAT) ainsi identifiée, aussi bien utilisatrice que réseau, se voit attribuer trois files d'attente:

- Une file d'attente pour les messages entrants dans l'application (MAPPLICAT).
- Une file d'attente pour les acquittements des messages envoyés par l'application (AAPPLICAT).
- Une file d'attente pour les rejets à destination de l'applicatif (RAPPLICAT). Elle contient les messages en provenance du réseau et considérés comme erronés par le réseau.

Nota: ces trois files d'attente peuvent être le même objet (par le menu de définition des files d'attente). Mais dans ce cas, le suivi des transmissions et la supervision de l'état des files d'attente perdront de leur clarté.

Ainsi, pour émettre un message sur le réseau ATLAS 400, l'applicatif utilisateur va déposer son message via les API dans la file d'attente des messages entrants de l'applicatif driver ATLAS 400, et recevra dans sa propre file d'attente d'acquittements les avis de distribution offerts par le RVA.

1.2. Mode d'utilisation

Maintenant l'objectif de vos applications est d'accéder à ces files d'attente par l'intermédiaire du nom logique d'application et via des fonctions ou commandes.

Pour cela, nous recommandons fortement d'utiliser les API dans les conditions suivantes:

- Pour une commande CMD ou un CL, utilisation de l'API par mot clef: cela assure la compatibilité ascendante de vos applicatifs avec les releases ultérieures de **TBT/400**.
- Pour un programme applicatif C, COBOL, RPG, ..., utilisation de l'API par les clauses copy, sans utiliser de paramètres positionnels, pour permettre toutes les évolutions futures.

Il est fourni trois accès différents à l'utilisation de **TBT/400** :

- accès à partir de vos applicatifs développés en Cobol, RPG/400, C, ...
- accès à partir de commandes spécifiques à chaque réseau d'accès.
Exemples: IPSNDATLAS, IPSNDFAX, ... pour le réseau ATLAS 400.
- accès à partir d'un écran utilisateur disponible dans la gestion du progiciel ou supervision **TBT/400**.

Les principales différences de ces niveaux d'API sont liées à la connaissance à acquérir pour les mettre en oeuvre, et aux services applicatifs qu'elles peuvent rendre.

Par exemple, la couche utilisateur accessible par CMD ou CL permet uniquement la constitution d'une adresse extérieure pour la transmission. Par contre, la couche accessible via vos applicatifs RPG/400 peut émettre, recevoir, et suivre la distribution de votre message (ou fichier).

1.3. Logique d'utilisation

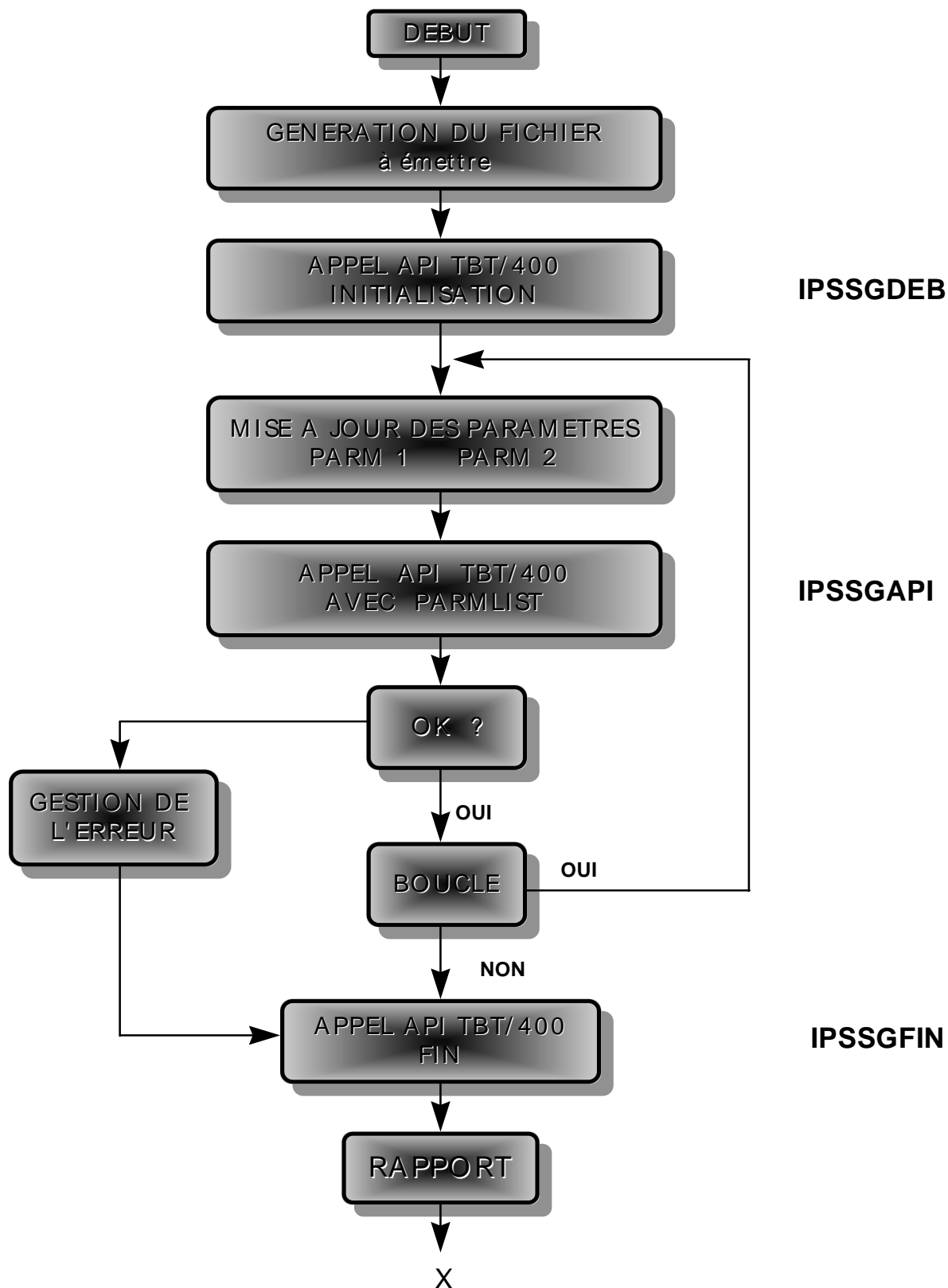
A partir de vos applicatifs C, RPG/400, Cobol/400, la logique d'accès aux API est la suivante:

- Appel des API afin d'initialiser l'environnement de travail propre à **TBT/400**.
IPSSGDEB
- Appel des API avec passage de paramètres qui vont indiquer: la fonction à réaliser, le fichier de travail et diverses identifications (émetteur, destinataire, ...).
IPSSGAPI Parm1 Parm2 ...
avec Parm1 = P0
avec Parm2 = P1
- Appel des API pour libérer l'environnement de travail de **TBT/400**.
IPSSGFIN

ATTENTION: les appels à IPSSGDEB sont implicites. Votre applicatif peut passer directement à l'appel de IPSSGAPI s'il n'y a pas de boucle de traitement des messages. Sinon, leur utilisation est fortement recommandée, pour des raisons d'intégrité pour IPSSGDEB en cas de plantage, et pour des raisons de libération mémoire pour IPSSGFIN.

Remarque: Pour la fonction d'initialisation IPSSGDEB, il est conseillé de passer le bloc de communication P0 (voir ci-après) pour que l'applicatif soit alerté d'une quelconque erreur système, et puisse reprendre l'initiative.

1.4. Schéma de fonctionnement



2. BLOCS DE COMMUNICATION

Les blocs de communication sont fournis en standard dans la bibliothèque IPLSP fichier IPSSAMPLES (voir Annexe).

2.1. Logique d'utilisation

Pour les fonctions d'initialisation et de terminaison des API de **TBT/400**, aucun paramètre n'est obligatoire. Seul l'appel véritable de l'action souhaitée nécessite l'information des blocs de communication. Ils sont au nombre de cinq:

- un bloc P0 de dialogue, commun à toutes les API.
- un bloc P1 de définition d'adresse externe et d'options, commun à tous les réseaux.
- un bloc P2 de type options, spécifique au réseau utilisé.
- un bloc P3 de type destinataire, spécifique au réseau utilisé.
- un bloc P4 de type texte, pour travailler avec le texte en mémoire et non par fichier.

Seuls les blocs **P0 et P1 sont obligatoires**, et ils sont **prioritaires** par rapport aux blocs P2 et P3.

En effet, certains champs du bloc P1 peuvent également exister (et porter le même nom) dans les blocs P2 ou P3. Dans ce cas, quand le même champ est renseigné dans le bloc P1 et dans l'un des blocs P2 ou P3, la valeur du champ du bloc P1 écrase la valeur du même champ dans l'autre bloc P2 ou P3. Par contre, si le même champ n'est pas renseigné dans P1, mais renseigné dans P2 ou P3, la valeur du champ dans P2 ou P3 renseigne le champ P1 correspondant.

Le programme applicatif renseigne uniquement les champs qui lui sont utiles, en utilisant au maximum les options par défaut. Lors de rappel de l'API, par exemple pour les émissions multiples, il est nécessaire de réinitialiser les blocs pour éviter la prise en compte de valeurs renseignées en retour.

Par CMD, seuls les deux premiers blocs P0 et P1 sont utilisés.

Pour une émission simple, seuls les champs code fonction (FNCDEM), nom du fichier à émettre (OBJFIL) et adresse du destinataire (par exemple NUMFAX) sont à renseigner, tous les autres champs étant facultatifs ou de valeur par défaut. L'utilisation des API est donc élémentaire pour les fonctions courantes.

CONVENTIONS DE PRESENTATION DES CHAMPS:

1. L'ordre de présentation des champs est celui des structures. Un index alphabétique est fourni en Annexe.
2. Dans les listes qui suivent, chaque champ est repéré par son nom, suivi du mode d'alimentation (A pour un champ à alimenter par le programme d'application, R pour un champ retourné par l'API), suivi de sa longueur, et enfin d'un qualificatif précisant s'il est obligatoire (O), facultatif (F), ou disposant d'une valeur par défaut (D).
3. Les champs existants dans les structures ou clauses copy fournies et non documentés dans les pages suivantes ne doivent en aucun cas être utilisés.
4. Pour la programmation en RPG, afin d'éviter des noms en double pour les champs communs à P1 et P2 ou P3, les noms communs sont modifiés dans le P2 et P3 en remplaçant le sixième caractère du nom par le numéro du bloc. Ceci ne complique ni ne change cependant pas la programmation, les noms corrects du bloc P1 pouvant être utilisés.

2.2. Bloc P0 de dialogue

	Nom	A / R	O / F	D
----- -----				
Fonction a réaliser	FNCDEM	A	O	
Début demandé	DEBDEM	A	---	D
Fin demandée	FINDEM	A	---	D
Exception demandée	EXCDEM	A	---	D
Trace demandée	TRADEM	A	---	D
Duplication demandée	DUPDEM	A	---	D
Annuaire local demandé	LOCDEM	A	---	D
Mode synchrone demandé	SYNDEM	A	---	D
Flag fin de texte	FLGEOF	R		
Nombre d'enregistrements	RECNBR	R		
Taille d'un enregistrement	RECSIZ	R		
Nombre de lignes	NBRLIG	A/R	---	F
Nombre de colonnes	NBRCOL	A/R	---	F
Code retour en Binaire	RTNCDB	---	---	R
Code retour en Packé	RTNCDP	---	---	R
Numéro de message émis	MSGNUB	---	---	R
Numéro de message émis en Packé	MSGNUP	---	---	R
Ident. du message émis	MSGIDN	---	---	R
Texte du message émis	MSGTXT	---	---	R
Nom du spool	SPLNAM	A	---	D
Job de creation	SPLJOB	A	---	D
Numero de spool	SPLNUM	A	---	D
Suppression du spoolfile	SPLSUP	A	---	D

FNCDEM	A	<i>1 caractère alphabétique</i>	O
---------------	---	---------------------------------	---

- Ce symbole précise la **fonction à réaliser**. Il doit être renseigné avec une des fonctions suivantes:
 - S = pour émettre,
 - R = pour recevoir, et aucun autre champ renseigné, sauf APPDES si mode manuel,
 - E = pour l'état d'un message, sélectionné par KEYTBT ou l'ensemble APPEME et KEYUSR,
 - P = pour purger un message sélectionné par KEYTBT ou l'ensemble APPEME et KEYUSR, en renseignant éventuellement ACKTBT, LIBTBT et SUPDEM. N'agit que sur les messages en cours d'émission, avec code retour à l'applicatif et historisation.
 - F = pour forcer la purge d'un message (idem P), mais agit ou que soit le message, sans code retour et sans historisation. Donc à n'employer qu'avec la plus grande prudence. KEYUSR, en renseignant éventuellement ACKTBT, LIBTBT et SUPDEM.
 - V = pour lire et purger un message dans le même appel (équivalent à R+P).
 - T = pour lire un texte dans un buffer applicatif (bloc P4), le fichier étant au préalable sélectionné par une fonction R, ou éventuellement V ou E.
 - I = pour ignorer, c'est à dire pas de fonction, aucun contrôle n'est effectué. La suppression est physique, et ne donne pas lieu à d'acquiescement.
 - C = pour contrôler sans émission (équivalent à S sans émission).
 - L = pour obtenir la copie du dernier bloc de contrôle (utile uniquement pour une émission par CL pour obtenir le code retour).
- Utiliser de préférence les équivalences (equats) fournies dans les clauses copy pour faciliter les évolutions.

DEBDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Demande l'**exécution de la procédure de début** de l'API (IPSSGDEB) avant la fonction demandée FNCDEM. La valeur par défaut est Non.

FINDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Demande l'**exécution de la procédure de terminaison** de l'API (IPSSGFIN) après la fonction demandée:
 - O = Oui,
 - N = Non (valeur par défaut),
 - C = Conditionnel, c'est à dire oui si le code retour RTNCDB est non nul (utile dans des boucles de programme).

EXCDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- En retour de l'API, **TBT/400** émet un message de type INFO, un message de type COMPLETION si bonne réalisation, et un message de type ESCAPE ou DIAG en cas de mauvaise réalisation. Si Oui, ce dernier message est de type COMPLETION, si Non il est de type DIAG. La valeur par défaut est Non, sauf si l'appel de l'API est effectué sans fournir de bloc P0 (par exemple en utilisant IPSSGDEB), ou si l'appel se fait en CLP, auquel cas la valeur par défaut est Oui.

TRADEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise si l'application désire **tracer ses appels** de l'API de **TBT/400**:
 - '0' = pas de trace (valeur par défaut),
 - '1' = trace demandée de niveau 1 (points de passage),
 - '2' = trace demandée de niveau 2 (points de passage et snaps des blocs).

DUPDEM	A	<i>1 caractère alphabétique</i>	D
<ul style="list-style-type: none"> Précise si l'application désire dupliquer le fichier lors du dépôt: <ul style="list-style-type: none"> - N = Non, l'applicatif s'interdit alors de modifier le fichier entre la demande d'émission et l'envoi réel sur le réseau (valeur par défaut). - O = Oui, une duplication est effectuée dans la bibliothèque IPLSE, libérant ainsi le fichier d'origine pour une autre émission. Cette option est implicite si la bibliothèque d'origine est QTEMP. 			
LOCDEM	A	<i>1 caractère alphabétique</i>	D
<ul style="list-style-type: none"> Précise si l'application désire utiliser l'annuaire local pour la recherche du nom logique du correspondant: <ul style="list-style-type: none"> - N = Non, la recherche est effectuée uniquement sur l'annuaire global (valeur par défaut). - O = Oui, la recherche est effectuée sur l'annuaire local et sur l'annuaire global. 			
SYNDEM	A	<i>1 caractère alphabétique</i>	D
<ul style="list-style-type: none"> Précise si l'application désire utiliser le mode synchrone, c'est à dire si l'applicatif doit rester en wait jusqu'à l'obtention du compte-rendu d'émission: <ul style="list-style-type: none"> - N = Non (valeur par défaut), - O = Oui. Non opérationnel dans cette version. 			
FLGEOF	R	<i>1 caractère alphabétique</i>	
<ul style="list-style-type: none"> Contient l'indicateur de fin de fichier suite à une fonction Texte: <ul style="list-style-type: none"> - O = plus de texte à récupérer, - N = texte à suivre. 			
RECNBR	R	<i>1 mot binaire</i>	
<ul style="list-style-type: none"> Contient le nombre d'enregistrements du fichier suite à une fonction Réception avec expansion du texte, c'est à dire si l'appel a été effectué en précisant le bloc P4 indiquant ainsi l'intention de lire le fichier. 			
RECSIZ	R	<i>1 mot binaire</i>	
<ul style="list-style-type: none"> Contient la taille des enregistrements du fichier, en octets, suite à une fonction Réception avec expansion du texte, c'est à dire si l'appel a été effectué en précisant le bloc P4 indiquant ainsi l'intention de lire le fichier. Pour la fonction Texte qui suit, il est recommandé d'avoir NBRCOL=RECSIZ, sinon il y a risque de troncature du texte dans un sens, ou perte de place mémoire dans l'autre sens. 			
NBRLIG	AR	<i>1 mot binaire</i>	F
<ul style="list-style-type: none"> Contient le nombre de lignes du buffer d'émission lorsque l'applicatif n'envoie pas un fichier, mais un texte formaté en mémoire et renseigné dans le bloc P4. TBT/400 crée et gère alors lui-même le fichier. Est utilisé conjointement avec NBRCOL. Egalement utilisé en réception lors d'une fonction Texte pour l'expansion du texte; spécifie le nombre de lignes réservées dans le buffer, et contient au retour le nombre de lignes réellement utilisées. 			
NBRCOL	AR	<i>1 mot binaire</i>	F
<ul style="list-style-type: none"> Contient le nombre de colonnes du buffer d'émission lorsque l'applicatif n'envoie pas un fichier, mais un texte formaté en mémoire et renseigné dans le bloc P4. TBT/400 crée et gère alors lui-même le fichier. Est utilisé conjointement avec NBRLIG. Egalement utilisé en réception lors d'une fonction Texte pour l'expansion du texte: spécifie le nombre de colonnes réservées dans le buffer, et contient au retour le nombre de colonnes réellement alimentées. 			

RTNCDB	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Contient le **code retour** renvoyé par l'interface de **bonne ou mauvaise réalisation** de la fonction demandée:
 - 0 = bonne réalisation,
 - autre valeur = mauvaise réalisation.

RTNCDP	R	<i>6 caractères packés</i>
---------------	---	----------------------------

- Contient le **code retour** en packé renvoyé par l'interface de **bonne ou mauvaise réalisation** de la fonction demandée.
-

Ces deux derniers champs, identiques en contenu, sont utilisés indifféremment selon le langage de programmation.

MSGNUB	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Contient le **numéro TBT/400** du message émis.

MSGNUP	R	<i>6 caractères packés</i>
---------------	---	----------------------------

- Contient le **numéro TBT/400** en packé du message émis.

Ces deux derniers champs, identiques en contenu, sont utilisés indifféremment selon le langage de programmation.

MSGIDN	R	<i>7 caractères alphanumériques</i>
---------------	---	-------------------------------------

- Contient l'**identification TBT/400** du message émis.

MSGTXT	R	<i>256 caractères alphanumériques</i>
---------------	---	---------------------------------------

- Contient le **libellé du compte-rendu** de la fonction réalisée, associé au code retour RTNCDB.

SPLNAM	A	<i>10 caractères alphanumériques</i>	D
---------------	---	--------------------------------------	---

- Contient le **nom du spoolfile** à envoyer sur le réseau.
Le nom par défaut est « TBT/400 ». Pour envoyer un spoolfile, la rubrique OBJFIL doit contenir la valeur spéciale « *SPLF », les rubriques SPLNAM, SPLJOB, SPLNUM doivent être renseignées.

SPLJOB	A	<i>26 caractères alphanumériques</i>	D
---------------	---	--------------------------------------	---

- Contient le **nom du job** ayant créé le spoolfile à envoyer sur le réseau.
Le nom par défaut est « * », ce qui signifie le job en cours.

SPLNUM	A	<i>1 mot binaire</i>	D
---------------	---	----------------------	---

- Contient le **numéro du spoolfile** à envoyer sur le réseau.
La valeur '0' précise que le spoolfile est unique.
La valeur '-1' demande l'émission du dernier spoolfile.

SPLSUP	A	<i>1 caractères alphanumérique</i>	D
---------------	---	------------------------------------	---

- Précise la règle de **suppression du spoolfile** après dépôt dans la file d'attente de TBT/400 d'une demande d'émission.
 - O = suppression immédiate après le dépôt
 - N = pas de suppression, donc le "ménage" devra être fait manuellement,
 - C = conditionnel, c'est à dire si le dépôt est accepté par TBT/400

2.3. Bloc P1 de dialogue

TYPOBJ	R	<i>1 caractère alphabétique</i>
---------------	---	---------------------------------

- Contient le **type d'objet entrant**:
 - M = message entrant
 - R = message de type rejet
 - A = message de type acquittement.
- Utiliser de préférence les équivalences (equats) fournies dans les clauses copy pour faciliter les évolutions.

TYPMSG	R	<i>1 caractère alphabétique</i>
---------------	---	---------------------------------

- Contient le **type de message entrant**:
 - T = compte rendu **TBT/400**,
 - M = courrier entrant,
 - P = prise en compte Réseau à Valeur Ajoutée,
 - I = prise en compte impression du message,
 - D = avis de distribution Réseau à Valeur Ajoutée.
- Utiliser de préférence les équivalences (equats) fournies dans les clauses copy pour faciliter les évolutions.

PRITBT	A	<i>1 demi mot binaire</i>	D
---------------	---	---------------------------	---

- Détermine la **priorité** interne affectée à votre message pour son traitement par **TBT/400**. Les valeurs possibles sont 0 à 255, la valeur par défaut étant 0.

STATBT	R	<i>1 caractère alphanumérique</i>
---------------	---	-----------------------------------

- Contient le **statut** interne du message dans **TBT/400**. Les valeurs possibles sont Non pour non traité, et Oui pour traité par **TBT/400** (dans ce cas le message est historisé).
- Utiliser de préférence les équivalences (equats) fournies dans les clauses copy pour faciliter les évolutions.

RECEMI	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre de records émis** du message.

NBREMI	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre de tentatives d'émission** du message.

FATDES	R	<i>9 caractères alphabétiques</i>
---------------	---	-----------------------------------

- Précise le **nom de la file d'attente destinatrice** du message.

FATAACK	R	<i>9 caractères alphabétiques</i>
----------------	---	-----------------------------------

- Précise le **nom de la file d'attente d'acquitements** utilisée pour la réception de l'acquitements.

APPEME	AR	<i>8 caractères alphanumériques</i>	D
---------------	----	-------------------------------------	---

- Précise le **nom de l'application émettrice**. En cas de suivi de transmission, les avis seront disponibles pour ce nom d'application. La valeur par défaut est \$INTERNA.

APPDES	AR	8 caractères alphanumériques	D
---------------	----	------------------------------	---

- Précise le **nom de l'application destinatrice**. Dans le cas d'une émission ou TBT/400 ne joue pas le rôle de serveur (par exemple ATLAS 400), la valeur est obligatoirement \$EXTERNA (valeur par défaut).

KEYTBT	AR	16 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Ne pas utiliser. Clé interne du message, réservée EXCLUSIVEMENT à l'usage de **TBT/400**.

KEYUSR	AR	16 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Précise l'**identification** du message pour l'application émettrice, pour servir de corrélation à l'applicatif.

KEYEXT	R	32 caractères alphanumériques	
---------------	---	-------------------------------	--

- Contient la **clé externe de corrélation** du réseau. Par exemple pour ATLAS 400, c'est la référence attribuée par ATLAS.

LIGTBT	AR	10 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Précise le nom logique de la **ligne d'entrée / sortie** du réseau utilisée. Ce champ, inutile pour les RVA, doit être renseigné uniquement pour les réseaux à accès direct (comme Téléx direct).

TYPRES	AR	10 caractères alphanumériques	D
---------------	----	-------------------------------	---

- Précise le **type de réseau** utilisé pour acheminer votre fichier, utilisable uniquement en fonction des options de **TBT/400** souscrites, par exemple (liste non limitative):
 - \$\$\$\$ATLAS = RVA ATLAS 400 (valeur par défaut),
 - \$\$\$TLXX25 = Sorties Banalisées Téléx de Transpac,
 - \$\$\$ETEBAS = serveur protocole ETEBAC 1-2 ou 3,
 - \$\$\$ETEBAC = remote protocole ETEBAC 3,
 - \$\$\$ODETTE = échanges en protocole ODETTE,
 - \$\$\$TBT = échanges en protocole **TBT**.

IDNRES	AR	32 caractères alphanumériques	D
---------------	----	-------------------------------	---

- **Précise l'identifiant du réseau, c'est à dire un nom de boîte ou signature demandé par le réseau pour que TBT/400 effectue une connexion.** Par exemple, pour ATLAS 400, c'est le nom de votre boîte aux lettres ATLAS (BAL0000001 par défaut pour le type de réseau \$\$\$\$ATLAS).

LECJOB	R	26 caractères alphanumériques	
---------------	---	-------------------------------	--

- Contient l'**identifiant AS/400 du job de consommation** de la file d'attente, composé du Jobname, du Username et du Jobnumber.

REFMSG	AR	32 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Précise la **référence** du message, propre à l'utilisateur.

DATPER	A	8 caractères numériques	D
---------------	---	-------------------------	---

- Précise la **date de péremption** pour **TBT/400**, de format AAAAMMJJ. Il est également possible de travailler avec un nombre de jours à partir de la date du jour, en saisissant +n, n étant ce nombre de jours. La valeur par défaut est celle générée à partir du nombre de jours précisé dans le menu de paramétrage des timers du système.

HORPER	A	8 caractères numériques	D
<ul style="list-style-type: none"> Précise l'heure de péremption pour TBT/400, de format HHMMSSCC. Il est également possible de travailler avec un nombre de minutes à partir de l'heure de traitement, en saisissant +m, m étant ce nombre de minutes. La valeur par défaut est celle générée à partir du nombre de secondes précisé dans le menu de paramétrage des timers du système. 			
DATDIF	A	8 caractères numériques	D
<ul style="list-style-type: none"> Précise la date d'envoi différé par TBT/400, de format AAAAMMJJ. Il est également possible de travailler avec un nombre de jours à partir de la date du jour, en saisissant +n, n étant ce nombre de jours. La valeur par défaut est départ immédiat. 			
HORDIF	A	8 caractères numériques	D
<ul style="list-style-type: none"> Précise l'heure d'envoi différé par TBT/400, de format HHMMSSCC. Il est également possible de travailler avec un nombre de minutes à partir de l'heure de traitement, en saisissant +m, m étant ce nombre de minutes. La valeur par défaut est départ immédiat. 			
DATFPC	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise la date de dépôt (d'insertion) du message par l'API TBT/400, c'est à dire son écriture dans la file d'attente, de format AAAAMMJJ. 			
HORFPC	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise l'heure de dépôt (d'insertion) du message par l'API TBT/400, c'est à dire son écriture dans la file d'attente, de format HHMMSSCC. 			
DATFTR	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise la date de traitement du message par TBT/400, de format AAAAMMJJ. 			
HORFTR	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise l'heure de traitement du message par TBT/400, de format HHMMSSCC. 			
DATRPC	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise la date de prise en compte envoyée par le réseau, de format AAAAMMJJ (uniquement valable pour les Réseaux à Valeur Ajoutée). 			
HORRPC	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise l'heure de prise en compte envoyée par le réseau, de format HHMMSSCC (uniquement valable pour les Réseaux à Valeur Ajoutée). 			
DATRTR	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise la date de remise envoyée par le réseau, de format AAAAMMJJ (uniquement valable pour les Réseaux à Valeur Ajoutée). 			
HORRTR	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise l'heure de remise envoyée par le réseau, de format HHMMSSCC (uniquement valable pour les Réseaux à Valeur Ajoutée). 			

CAREMI	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre de caractères émis** sur le réseau, en octets.

CARREC	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre de caractères reçus** du réseau, en octets.

DURCOM	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise la **durée de la communication** sur le réseau, en secondes.

NBRREC	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre d'enregistrements reçus** du réseau.

LNGREC	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise la **longueur des enregistrements reçus** du réseau, en octets.

FILSIZ	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise la **taille du fichier reçu** du réseau, en K octets.

HAUPAG	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise la **hauteur de page** retenue pour l'émission du message.

RECSEG	R	<i>1 mot binaire</i>
---------------	---	----------------------

- Précise le **nombre d'enregistrements par segment** du fichier reçu du réseau.

TYPTRN	AR	<i>1 caractère alphabétique</i>	D
---------------	----	---------------------------------	---

- Précise si l'expédition est en **mode transparent**:
 - O = mode transparent,
 - N = mode normal ou mode texte (valeur par défaut).
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

CRLDEM	AR	<i>1 caractère alphabétique</i>	D
---------------	----	---------------------------------	---

- Permet d'ajouter les **caractères de séparation de ligne CR/LF** (Carriage Return/Line Feed) des messages en émission. Cette facilité permet notamment d'envoyer des messages ou fichiers en format EDI (par exemple avec les segments UNB, UNA, ...). Les valeurs possibles sont:
 - O = ajout des caractères de séparation en émission,
 - N = message non modifié (cas de l'EDI).
- La valeur par défaut dépend de la valeur du champ TYPTRN, pour automatiser le process (Texte ou EDI):
 - si TYPTRN = N (mode texte), alors la valeur par défaut est Oui,
 - si TYPTRN = O (mode transparent), alors la valeur par défaut est Non.
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

SPADEM	AR	<i>1 caractère alphabétique</i>	D
---------------	----	---------------------------------	---

- Permet de supprimer les **caractères blancs en fin de ligne** en émission. Les valeurs possibles sont:
 - O = suppression des caractères blancs de fin de ligne en émission,
 - N = message non modifié.
- La valeur par défaut dépend de la valeur du champ TYPTRN, pour automatiser le process (Texte ou EDI):
 - si TYPTRN = N (mode texte), alors la valeur par défaut est Oui,
 - si TYPTRN = O (mode transparent), alors la valeur par défaut est Non.
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

ASCDEM	AR	<i>1 caractère alphabétique</i>	D
---------------	----	---------------------------------	---

- Si Oui, permet d'effectuer la **conversion des caractères EBCDIC en caractères ASCII**.
- La valeur par défaut dépend de la valeur du champ TYPTRN, pour automatiser le process (Texte ou EDI):
 - si TYPTRN = N (mode texte), alors la valeur par défaut est Oui,
 - si TYPTRN = O (mode transparent), alors la valeur par défaut est Non.
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

ACKDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise si l'application émettrice désire recevoir un **accusé d'acheminement** de TBT/400:
 - O = oui dans tous les cas,
 - N = non dans tous les cas (valeur par défaut),
 - C = oui seulement dans le cas d'erreur.
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

SUPDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise la **règle de suppression** du message après son émission:
 - O = suppression immédiate après l'émission,
 - N = pas de suppression, le "ménage" devra donc être manuelle (valeur par défaut),
 - H = suppression lors de la réception de l'historique,
 - C = conditionnel, c'est à dire si l'émission est correcte. Dans le cas d'Atlas, si l'avis est demandé, c'est l'avis positif qui entraîne la suppression, sinon dans tous les autres cas c'est lors de la réception de l'historique.

PUIDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise l'utilisation de la **fonction puits** pour l'envoi du message:
 - O = si plusieurs messages pour le même destinataire, s'il y a lieu chaînage des messages lors de la même communication (valeur par défaut),
 - N = pas de chaînage, le message sera émis isolément.
 L'utilisation effective de cette fonction dépend du réseau utilisé.
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

IMPDEM	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise si l'**impression du message** après bonne émission est demandée sur l'OUTPUT QUEUE du USER:
 - O = impression systématique (valeur par défaut pour les réseaux Fax et Téléx),
 - N = pas d'impression (valeur par défaut pour tous les autres réseaux),
 - C = conditionnelle, c'est à dire uniquement en cas d'erreur,,
 - B = bon, c'est à dire uniquement en cas de bon acheminement.

OUTLIB	R	<i>10 caractères alphanumériques</i>	
---------------	---	--------------------------------------	--

- Contient le **nom de la bibliothèque** utilisée pour l'impression du message.

OUTFIL	R	<i>10 caractères alphanumériques</i>	
---------------	---	--------------------------------------	--

- Contient le **nom de l'OUTPUT QUEUE** associée au JOB du USER, utilisée pour l'impression du message.

AVIDIS	A	<i>1 caractère alphabétique</i>	D
---------------	---	---------------------------------	---

- Précise si l'**avis de distribution RVA** est demandé:
 - O = avis de distribution pour tous les messages,
 - N = avis de distribution uniquement pour les remises impossibles (valeur par défaut).
- Nota: Vous pouvez préciser la valeur de ce champ en fonction du correspondant existant dans l'annuaire. Dans ce cas, la valeur par défaut prise est la valeur renseignée dans l'annuaire, si elle existe. Sinon, la valeur par défaut reste la valeur standard précisée ci-dessus.

COMUSR	AR	<i>128 caractères alphanumériques</i>	F
---------------	----	---------------------------------------	---

- **Commentaire libre** pour usage éventuel par l'applicatif.

USRPRF	R	<i>10 caractères alphanumériques</i>	
---------------	---	--------------------------------------	--

- Contient le **userid** utilisé par l'émetteur du message.

IDNJOB	R	<i>26 caractères alphanumériques</i>	
---------------	---	--------------------------------------	--

- Contient l'**identifiant AS/400 du job** utilisé par l'émetteur ou l'application émettrice, composé du Jobname, du Username et du Jobnumber.

ACKTBT	R	<i>2 caractères alphanumériques</i>	
---------------	---	-------------------------------------	--

- Contient l'**accusé d'acheminement** de **TBT/400**:
 - low value = en attente de traitement par **TBT/400**,
 - PC = pris en compte par le réseau,
 - blanc = distribué par le réseau,
 - ED = non distribution par le réseau,
 - autre valeur = incident répertorié dans la table des erreurs réseau correspondante.

LIBTBT	R	128 caractères alphanumériques	
<ul style="list-style-type: none"> Contient le libellé relatif à l'accusé d'acheminement de TBT/400, provenant du menu des erreurs, complété éventuellement pour plus de précision. L'applicatif ne doit donc pas tester ce champ. 			
OBJLIB	AR	10 caractères alphabétiques	D
<ul style="list-style-type: none"> Précise le nom de la bibliothèque du fichier reçu ou à transmettre. La valeur par défaut est *LIBL. 			
OBJFIL	AR	10 caractères alphabétiques	O
<ul style="list-style-type: none"> Précise le nom du fichier reçu ou à transmettre. 			
OBJMBR	AR	10 caractères alphabétiques	D
<ul style="list-style-type: none"> Précise le nom du membre reçu ou à transmettre. La valeur par défaut est *FIRST. 			
Ces trois derniers champs contiennent, après une fonction Send, les noms réels utilisés.			
GRPPRF	AR	10 caractères alphanumériques	F
<ul style="list-style-type: none"> Précise le groupe d'appartenance entré dans l'annuaire. 			
DATRTN	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise la date de dernière tentative d'envoi sur le réseau, de format AAAAMMJJ. 			
HORRTN	R	8 caractères numériques	
<ul style="list-style-type: none"> Précise l'heure de dernière tentative d'envoi sur le réseau, de format HHMMSSCC. 			
LIBANN	R	30 caractères alphanumériques	
<ul style="list-style-type: none"> Précise le libellé du correspondant entré dans l'annuaire. 			
TYPANN	AR	10 caractères alphanumériques	D
<ul style="list-style-type: none"> Précise le type d'annuaire du correspondant. La valeur par défaut est \$ANNUAI si le nom logique est renseigné. 			
TYPLOC	R	10 caractères alphanumériques	
<ul style="list-style-type: none"> Précise la portée de l'annuaire ou le nom du correspondant a été trouvé (global ou user particulier). 			
NOMLOG	AR	20 caractères alphanumériques	F
<ul style="list-style-type: none"> Précise le nom logique du correspondant tel que défini dans l'annuaire. 			
CUDLNG	AR	1 caractère alphanumérique	F
<ul style="list-style-type: none"> Précise la longueur en digits du Common User Data quant il est nécessaire à votre correspondant. 			
CUDEM	AR	16 caractères alphanumériques	F
<ul style="list-style-type: none"> Contient le Common User Data en hexadécimal quant il est nécessaire à votre correspondant. 			
FACLNG	AR	1 caractère alphanumérique	F
<ul style="list-style-type: none"> Précise la longueur du champ services complémentaires quant il est nécessaire à votre correspondant. 			

FACDEM	AR	16 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Contient le **champ services complémentaires** quant il est nécessaire à votre correspondant.

TADDEM	AR	1 caractère alphanumérique	F
---------------	----	----------------------------	---

- Précise si la **taxation au demandé** est acceptée par votre correspondant (en émission), ou a été acceptée (en réception).

DBIDEM	AR	1 caractère alphanumérique	F
---------------	----	----------------------------	---

- Précise si l'on utilise le **Delivery Bit** au sein de la connexion X25, c'est à dire si l'acquittement X25 se fait à l'arrivée. Pour un appel sortant, dans ce cas tous les paquets l'utilisent, et donc les performances réseau peuvent être dégradée. Pour un appel entrant, cette utilisation demandée par votre correspondant est alors ici simplement signalée.

PRIRES	AR	1 caractère alphabétique	D
---------------	----	--------------------------	---

- Précise la **priorité** du message gérée par le réseau:
 - U = urgent,
 - N = normal (valeur par défaut),
 - H = heures creuses.

AUTHOR	AR	32 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Précise l'**auteur** du message.

OBJECT	AR	256 caractères alphanumériques	F
---------------	----	--------------------------------	---

- Précise l'**objet** du message, c'est à dire permet d'indiquer brièvement le contenu du message.

COMREC	R	32 caractères alphanumériques	
---------------	---	-------------------------------	--

- Précise la **zone de commutation en réception** en provenance du réseau. Pour Atlas, c'est l'unité d'organisation A.

ATTENT	AR	32 caractères alphanumériques	F
---------------	----	-------------------------------	---

- Précise l'**attention** désignant l'utilisateur visé du message.

IDTERM	AR	24 caractères numériques	F
---------------	----	--------------------------	---

- Précise l'**indicatif de terminal** du destinataire du message.

NUMNOR	A	16 caractères numériques	F
---------------	---	--------------------------	---

- Précise l'**adresse** de type normalisé **X121**.

NUMTLX	A	16 caractères numériques	F
---------------	---	--------------------------	---

- Précise le **numéro télex** sous la forme code pays télégraphique + numéro. Pour la France, ne codifier que le numéro à 6 chiffres (voir Annexe).

NUMFAX	A	16 caractères numériques	F
---------------	---	--------------------------	---

- Précise le **numéro télécopie** sous la forme code pays téléphonique + numéro. Pour la France, ne codifier que le numéro à 8 chiffres, précédé de 1 pour un destinataire situé en Ile de France (voir Annexe).

NUMTPC	A	16 caractères numériques	F
--------	---	--------------------------	---

- Précise le **numéro Transpac**.

NUMRNI	A	16 caractères numériques	F
--------	---	--------------------------	---

- Précise le **numéro Numéris**.
- **ATTENTION:** . Un des 5 derniers champs de type adresse (et un seul) doit être renseigné en émission, sauf si une adresse propre au réseau est renseignée dans les blocs P2 ou P3 (par exemple, pour ATLAS 400, c'est l'adresse X400 composée des champs CODCTR, DOMADM, DOMPRV, NOMPER, ORGANI et UNORGA).
- Si un des 4 derniers champs est renseigné, il sert à encoder l'adresse NUMNOR, qui est alors écrasée même si elle est également renseignée.

SSAADR	A	16 caractères numériques	F
--------	---	--------------------------	---

- Précise la **sous-adresse X25** utilisée si demandée par votre correspondant.

3. Exemples de programmes

3.1. Programme d'émission Telex X25

Ce programme permet d'envoyer un message sous forme de fichier (bannière **IPLS**), sur un terminal Télex dont le numéro est fourni en paramètre, par le réseau Transpac.

NOTA: Ces exemples ne sont fournis qu'à titre purement documentaire.

Programme en langage RPG (IPZPGRPTLX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

*****
* Exemple d'appel de l'API TBT/AS400 *
* *
* Ce programme émet un Télex par le réseau TRANSPAC *
* Le numéro du destinataire lui est passé en argument *
* *
*****
H          1      D-J
*
*
*****
*   définition des divers constantes
I          'IPSSGDEB'          C          WAPIDB
I          'IPSSGFIN'         C          WAPIFN
I          'IPSSGAPI'         C          WAPI
I          'IPSSAMPLES'       C          WFILE
I          'IPZIGBAN'         C          WMEMB
I          'DEMONSTRATION TBT' C          WAUT
I          'EXEMPLE D EMISSION' C        WOBJ
I          'MONSIEUR LE DESTINAT-C' WATTN
I          'AIRE'
*****
*   NBPARAM indique le nombre de paramètres reçus
I          SDS
I          *PARMS  NBPARAM
*
*****
*   décomposition des blocs de communication généraux
I/COPY IPSSAMPLES,IPSIRPAP
*
*****
* Paramètre  PARMNU: Numéro d'appel
*           RTNCOD: Code Retour si différent de Zéro alors
*           appel erroné.
*****
/EJECT
C          *ENTRY  PLIST
C          PARM          PARMNU 16
C          PARM          RTNCOD  1
*
* validation un paramètre obligatoire
*
C          NBPARAM  IFNE 2
C          GOTO FIN
C          END
*
* appel fonction début de l'API

```

```

*
C          CALL WAPIDB
*
* initialisations
*
C          MOVE '0'          RTNCOD
C          EXSR INIBLK
* fonction d'émission
C          MOVE APISND      FNCDEM
* désignation de l'objet à envoyer: fichier + membre
C          MOVE WFILE      OBJFIL
C          MOVE WMEMB      OBJMBR
* alimente le numéro Téléx
C          MOVEPARMNU      NUMTLX      P
* alimente l'auteur du courrier
C          MOVEWAUT        AUTHOR      P
* alimente l'objet du courrier
C          MOVEWOBJ        OBJECT      P
* alimente le A l'attention de ...
C          MOVEWATTN      ATTENT      P
*
* appel des API de TBT
*
C          EXSR TBTAPI
*
* appel de TERMINAISON DES API
*
C          CALL WAPIFN
* fin du module général
C          FIN          TAG
C          RETRN
*
* initialisation des blocs
*
*
C          INIBLK      BEGSR
C          MOVE*LOVAL  WP0
C          MOVE*LOVAL  WP1
C          MOVE '2'    TRADEM
C          ENDSR
* appel la fonction fin de l'API
*
C          TBTAPI      BEGSR
C          CALL 'IPSSGAPI'
C          PARM          WP0
C          PARM          WP1
C          ENDSR
*

```

Programme en langage Cobol (IPZPGCBTLX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

IDENTIFICATION DIVISION.
PROGRAM-ID.          ZPGCBTLX.
AUTHOR.             IPLS.
DATE-WRITTEN.       1993.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY IPSICBAP.

```

```

LINKAGE SECTION.
01  PARM-NUTLX          PIC X(16).
PROCEDURE DIVISION USING PARM-NUTLX.

*validation un paramètre obligatoire
  IF PARM-NUTLX NOT > SPACES
    GO TO FIN.

* appel fonction début de l'API
  CALL 'IPSSGDEB'.

* initialisation des blocs
  PERFORM INI-BLOC THRU FIN-INI-BLOC.

* fonction d'émission de message
  MOVE APISND          TO P0-FNCDEM.

* désignation de l'objet à envoyer fichier + membre
  MOVE "IPSSAMPLES" TO P1-OBJFIL.
  MOVE "IPZIGBAN"   TO P1-OBJMBR.

* alimente numéro Téléx
  MOVE PARM-NUTLX    TO P1-NUMTLX.

* alimente zones utilisateur
  MOVE "DEMONSTRATION TBT"          TO P1-AUTHOR.
  MOVE "EXEMPLE D EMISSION"         TO P1-OBJECT.
  MOVE "MONSIEUR LE DESTINATAIRE" TO P1-ATTENT.

* appel de l'API
  CALL "IPSSGAPI" USING  WP0
                          WP1.

* appel fonction fin de l'API
  CALL "IPSSGFIN".

* fin du programme
FIN.
  STOP RUN.

INI-BLOC.
  MOVE LOW-VALUE TO WP0.
  MOVE LOW-VALUE TO WP1.
FIN-INI-BLOC.
  EXIT.

```

Programme en langage C (IPZPGLCTLX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

/*****
/* Exemple d'appel de l'API TBT/AS400 */
/*
/* Ce programme émet un Téléx par le réseau TRANSPAC */
/* Le numéro du destinataire lui est passé en argument */
/*
/*****

```

```
#include "ipsilcap.ipssamples"      /* Include files TBT      */
                                   /* Le fichier IPSSAMPLES */
                                   /* doit être accessible  */
                                   /* dans la "Liblist"     */
                                   /* du Job de compilation */

/*****
/* Point d'entrée
*****/

int main(int argc,char *argv[])

{
TBTBLOCS()                          /* Macro de définition   des blocs TBT */

if (argc != 2)                       /* Validation un paramètre obligatoire */
    abort();

IPSSGDEB();                          /* Appel fonction début de l'API      */

TBTINIT();                          /* Macro d'initialisation des blocs TBT */

wtbt_p0.fncdem = FNCAPISND;          /* Fonction d'émission de message    */

TBTLIT(wtbt_p1.objfil,"IPSSAMPLES"); /* Désignation de l'objet à          */
TBTLIT(wtbt_p1.objmbr,"IPZIGBAN");   /* envoyer : Fichier + Membre        */
                                   /* --> Bannière IPLS              */

TBTSTR(wtbt_p1.numtlx,argv[1]);      /* Alimente numéro télex            */

TBTLIT(wtbt_p1.author,"Demonstration TBT");
TBTLIT(wtbt_p1.object,"Exemple d'émission");
TBTLIT(wtbt_p1.attent,"Monsieur le Destinataire");

IPSSGAPI(&wtbt_p0,&wtbt_p1,NULL,NULL); /* Appel de l'API */

IPSSGFIN();                          /* Appel fonction fin de l'API      */

return(wtbt_p0.rtncdb);
}
```

3.2. Programmes Atlas 400 ou Graphnet

Les différents exemples de programmes qui suivent permettent de mieux illustrer l'utilisation des API pour des échanges avec Atlas 400 ou un Réseau à Valeur Ajoutée. Ces programmes sont en langage C, sauf le premier qui est repris en RPG et en Cobol.

NOTA: Ces exemples ne sont fournis qu'à titre purement documentaire.

3.2.1. Programme d'émission de Fax

Ce programme permet d'envoyer un message sous forme de fichier (bannière IPLS) sur un terminal Fax dont le numéro est fourni en paramètre, par le réseau Atlas 400 ou Graphnet.

Programme en langage RPG (IPZPGRPFAX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

*****
* Exemple d'appel de l'API TBT/AS400 *
* *
* Le numéro du destinataire lui est passé en argument *
* *
*****
H          1      D-J                                RPFAX
*
*
*****
*   définition des divers constantes
I          'IPSSGDEB'          C          WAPIDB
I          'IPSSGFIN'         C          WAPIFN
I          'IPSSGAPI'         C          WAPI
I          'IPSSAMPLES'       C          WFILE
I          'IPZIGBAN'         C          WMEMB
I          'DEMONSTRATION TBT' C          WAUT
I          'EXEMPLE D EMISSION' C        WOBJ
I          'MONSIEUR LE DESTINAT-C  WATTN
I          'AIRE'
*****
*   NBPARM indique le nombre de param2tres recus
I          SDS
I          *PARMS      NBPARM
*
*****
*   décomposition des blocs de communication généraux
I/COPY IPSSAMPLES,IPSIRPAP
*
*****
* Parametre  NUMFAX: Numéro d'appel
*           RTNCOD: Code Retour si different de Zéro alors
*           appel erronné.
*****
/EJECT
C          *ENTRY      PLIST
C          PARM          PARMNU 16
C          PARM          RTNCOD  1
*
* validation un parametre obligatoire
*
C          NBPARM      IFNE 2

```



```

C          GOTO FIN
C          END
*
* appel fonction début de l'api
*
C          CALL WAPIDB
*
* initialisations
*
C          MOVE '0'          RTNCOD
C          EXSR INIBLK
* fonction d'émission
C          MOVE APISND      FNCDEM
* désignation de l'objet a envoyer: fichier + membre
C          MOVEW WFILE      OBJFIL    P
C          MOVEW WMEMB      OBJMBR    P
* alimente le numero de fax
C          MOVEW PARMNU     NUMFAX    P
* alimente l'auteur du courrier
C          MOVEW WAUT       AUTHOR    P
* alimente l'objet du courrier
C          MOVEW WOBJ       OBJECT    P
* alimente le a l'attention de ...
C          MOVEW WATTN      ATTENT    P
*
* appel des api de tbt
*
C          EXSR TBTAPI
*
* appel de TERMINAISON DES API
*
C          CALL WAPIFN
* fin du module général
C          FIN          TAG
C          RETRN
*
* initialisation des blocs
*
*
C          INIBLK      BEGSR
C          MOVEW*LOVAL  WP0
C          MOVEW*LOVAL  WP1
C          MOVE '2'     TRADEM
C          ENDSR
* appel la fonction fin de l'api
*
C          TBTAPI      BEGSR
C          CALL 'IPSSGAPI'
C          PARM          WP0
C          PARM          WP1
C          ENDSR
*

```

Programme en langage Cobol (IPZPGCBFAX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

IDENTIFICATION DIVISION.
PROGRAM-ID.          IPZPGCBFAX.
AUTHOR.             IPLS.
DATE-WRITTEN.       1996.

```

```

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY IPSICBAP OF IPSSAMPLES.
LINKAGE SECTION.
00459800
01  PARM-NUFAX          PIC X(16).
PROCEDURE DIVISION USING PARM-NUFAX.
DEBUT.
*validation un parametre obligatoire
    IF PARM-NUFAX NOT > SPACES
        GO TO FIN.

* appel fonction debut de l'api
    CALL "IPSSGDEB".

* initialisation des blocs
    MOVE LOW-VALUE TO WP0.
    MOVE LOW-VALUE TO WP1.

* fonction d'émission de message
    MOVE APISND          TO P0-FNCDEM.

* désignation de l'objet a envoyer fichier + membre
    MOVE "IPSSAMPLES" TO P1-OBJFIL.
    MOVE "IPZIGBAN"   TO P1-OBJMBR.

* alimente numéro fax
    MOVE PARM-NUFAX     TO P1-NUMFAX.

* alimente identification destinataire (Facultatif)
    MOVE "DEMONSTRATION TBT"          TO P1-AUTHOR.
    MOVE "EXEMPLE D EMISSION"         TO P1-OBJECT.
    MOVE "MONSIEUR LE DESTINATAIRE" TO P1-ATTENT.

* appel de l'api
    CALL "IPSSGAPI" USING WP0
                                WP1.

* appel fonction fin de l'api
    CALL "IPSSGFIN".

* fin du programme
FIN.
STOP RUN.

```

Programme en langage C (IPZPGLCFAX disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP):

```

/*****/
/* Exemple d'appel de l'API TBT/AS400 */
/* */
/* Ce programme émet un Fax par le réseau ATLAS 400 */
/* ou Graphnet */
/* Le numéro du destinataire lui est passé en argument */
/* */
/*****/

#include "ipsilcap.ipssamples" /* Include files TBT */
/* Le fichier IPSSAMPLES */

```

```

/* doit etre accessible */
/* dans la "Liblist" */
/* du Job de compilation */

/*****
/* Point d'entree */
*****/

int main(int argc,char *argv[])

{
TBTBLOCS() /* Macro de définition des blocs TBT */

if (argc != 2) /* Validation un parametre obligatoire */
    abort();

IPSSGDEB(); /* Appel fonction début de l'API */

TBTINIT(); /* Macro d'initialisation des blocs TBT */

wtbt_p0.fncdem = FNCAPISND; /* Fonction d'émission de message */

TBTLIT(wtbt_p1.objfil,"IPSSAMPLES"); /* Désignation de l'objet ; */
TBTLIT(wtbt_p1.objmbr,"IPZIGBAN"); /* envoyer : Fichier + Membre*/
/* --> Banniere IPLS */

TBTSTR(wtbt_p1.numfax,argv[1]); /* Alimente numéro de fax */

TBTLIT(wtbt_p1.author,"Demonstration TBT");
TBTLIT(wtbt_p1.object,"Exemple d'emission");
TBTLIT(wtbt_p1.attent,"Monsieur le Destinataire");

IPSSGAPI(&wtbt_p0,&wtbt_p1,NULL,NULL); /* Appel de l'API */

IPSSGFIN(); /* Appel fonction fin de l'API */

return(wtbt_p0.rtncdb);
}

```

3.2.2. Programme d'émission de message

Ce programme (IPZPGLCEMI disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP) permet d'envoyer une série de messages sous forme de fichiers (contenant une bannière IPLS), à une application \$\$\$DEM. La lecture des messages et des acquittements demandés se fait par les autres exemples qui suivent.

```

/*****
/* Exemple d'appel de l'API TBT/AS400 */
/*
/* Ce programme émet une série de messages dont le nombre */
/* est passé en paramètre. L'application destinatrice */
/* est $$$DEM. Les acquittements sont demandés; les files */
/* d'attente message et acquittements sont déclarées en */
/* démarrage automatique, et associées aux programmes */
/* IPZPGLCREC et IPZPGLCACK respectivement; */
/*****
#include <stdio.h>
#include "ipsilcap.ipssamples" /* Include files TBT */

/* Le fichier IPSSAMPLES */
/* doit être accessible */
/* dans la "Liblist" */
/* du Job de compilation */

#include <time.h>

typedef struct STAMP_S
{ char datheu[12];
  char numero[4];
} STAMP;

/*****
/* Procédure d'insertion d'un message */
/*****

static int insrtmsg (int numeromes)

{
STAMP stamp;
time_t ltime;
TBTBLOCS() /* Implémentation des blocs TBT/400 */
TBTINIT(); /* Initialisation de ceux-ci */

/* Définition d'une clé utilisateur (basée sur un timestamp) */
/* ----- */

time(&ltime);
strftime((char *) &stamp.datheu, sizeof(stamp.datheu)+1,
"%y%m%d%H%M%S", localtime(&ltime));
sprintf((char *) &stamp.numero, "%4.4i", numeromes);

wtbt_p0.fncdem = FNCAPISND; /* Fonction Send */

TBTLIT(wtbt_p1.appeme, "$$$DEM"); /* Application émettrice */
TBTLIT(wtbt_p1.appdes, "$$$DEM"); /* Application Destinatrice */
/* ----- */
/* En forçant l'application destinatrice, le message se verra */

```

```

/* appliqué tous les controles liés ; ATLAS/400, mais ne partira */
/* ou a Graphnet */
/* pas sur le réseau : Simulation */
/* ----- */

TBTLIT(wtbt_p1.objfil,"IPSSAMPLES"); /* Fichier a envoyer */
TBTLIT(wtbt_p1.objmbr,"IPZIGBAN"); /* Membre */

TBTLIT(wtbt_p1.numfax,"112233445"); /* Numéro fax "Bidon" */

wtbt_p1.ackdem = TYPACKOUI; /* Acquittements souhaités */

TBTLIT(wtbt_p1.author,"Demonstration TBT");
TBTLIT(wtbt_p1.object,"Exemple d'emission");
TBTLIT(wtbt_p1.attent,"Monsieur le Destinataire");

memcpy(&wtbt_p1.keyusr,&stamp,sizeof(wtbt_p1.keyusr));

IPSSGAPI(&wtbt_p0,&wtbt_p1,NULL,NULL); /* Appel de l'API */

return(wtbt_p0.rtncdb);
}

/*****
/* Point d'entree */
*****/

int main(int argc,char *argv[])

{
int nbre_fois = 0;
int nbre_bouc = 0;
int nbre_mess = 0;
int nbre_iter;

if (argc != 2)
nbre_fois = 50;
else
nbre_fois = atoi(argv[1]);

nbre_iter = nbre_fois;

IPSSGDEB(); /* API de début */

while (nbre_iter-- )
{ ++ nbre_bouc;
if (insrtmsg(nbre_bouc))
break;
++ nbre_mess;
}

IPSSGFIN(); /* API de fin */

return(0);
}

```

3.2.3. Programme de réception de message

Ce programme (IPZPGLCREC disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP) permet de recevoir et de traiter les messages envoyés par le programme précédent.

```

/*****
/* Exemple d'appel de l'API TBT/AS400 */
/*
/* Ce programme reçoit les messages émis par le programme
/* IPZPGLCEMI.
*****/

#include "ipsilcap.ipssamples" /* Include files TBT */

/* Le fichier IPSSAMPLES */
/* doit etre accessible */
/* dans la "Liblist" */
/* du Job de compilation */

/*****
/***** Sample de Consommation en File d'Attente */
/*****
int main(int argc,char *argv[])

{
TBTBLOCS() /* Définition des blocs */

TBTINIT(); /* Initialisation de ceux-ci */

IPSSGDEB(); /* Appel de l'API d'initialisation */

while (1) /* Boucle jusqu'a épuisement */
{
/* ----- */
/* Si la file d'attente est en démarrage automatique,
/* les deux lignes suivantes sont inutiles,
/* la sélection étant implicite.
/* ----- */
TBTLLIT(wtbt_p1.appdes,"$$$$DEM"); /* Application a consommer*/
wtbt_p1.typobj = TYPOBJMSG; /* Objets souhaités */

wtbt_p0.fncdem = FNCAPIRCV; /* Fonction Receive */
IPSSGAPI(&wtbt_p0,&wtbt_p1,NULL,NULL);
if ( wtbt_p0.rtncdb ) /* Abandon si erreur */
break;

/* ..... */
/* Traitement message */
/* ..... */

wtbt_p0.fncdem = FNCAPIPUR; /* Fonction Purge */
IPSSGAPI(&wtbt_p0,&wtbt_p1,NULL,NULL);
if ( wtbt_p0.rtncdb ) /* Abandon si erreur */
break;

}

```

```
IPSSGFIN();                               /* Api de fin           */
return(0);
}
```

3.2.4. Programme de réception d'acquiescement

Ce programme (IPZPGLCACK disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP) permet de recevoir et de traiter les messages d'acquiescement envoyés par le programme d'émission de message.

```

/*****
/* Exemple d'appel de l'API TBT/AS400 */
/*
/* Ce programme recoit les acquiescements des messages émis */
/* par le programme IPZPGLCEMI et recus par IPZPGLCREC */
/*****

#include "ipsilcap.ipssamples" /* Include files TBT */

/* Le fichier IPSSAMPLES */
/* doit etre accessible */
/* dans la "Liblist" */
/* du Job de compilation */

/*****
/***** Sample de Consommation en File d'Attente */
/*****
int main(int argc,char *argv[])

{
TBTBLOCS() /* Définition des blocs */

TBTINIT(); /* Initialisation de ceux-ci */

IPSSGDEB(); /* Appel de l'API d'initialisation */

while (1) /* Boucle jusqu'à épuisement */
{
/* ----- */
/* Si la file d'attente est en démarrage automatique, */
/* les deux lignes suivantes sont inutiles */
/* la sélection étant implicite. */
/* ----- */
TBTLIT(wtbt_pl.appdes,"$$$$DEM"); /* Application ; consommer*/
wtbt_pl.typobj = TYPOBJACK; /* Objets souhaités */

wtbt_p0.fncdem = FNCAPIRCV; /* Fonction Receive */
IPSSGAPI(&wtbt_p0,&wtbt_pl,NULL,NULL);
if ( wtbt_p0.rtncdb ) /* Abandon si erreur */
break;

/* ..... */
/* Traitement acquiescement*/
/* ..... */

wtbt_p0.fncdem = FNCAPIPUR; /* Fonction Purge */
IPSSGAPI(&wtbt_p0,&wtbt_pl,NULL,NULL);
if ( wtbt_p0.rtncdb ) /* Abandon si erreur */
break;

}

```



```
IPSSGFIN(); /* Api de fin */  
return(0);  
}
```

3.2.5. Programme de réception par CL

Ce programme (IPZPGCLRCV disponible dans le fichier IPSSAMPLES, bibliothèque IPLSP) permet de recevoir les messages par la commande CL IPSRCVTBT intégrée dans votre exploitation.

```

PGM
DCL      VAR (&DEBDEM) TYPE (*CHAR) LEN(1) VALUE(0)
DCL      VAR (&MSGTXT) TYPE (*CHAR) LEN(256)
DCL      VAR (&RTNCDP) TYPE (*DEC) LEN(11)
DCL      VAR (&TYPOBJ) TYPE (*CHAR) LEN(1)
DCL      VAR (&TYPMSG) TYPE (*CHAR) LEN(1)
DCL      VAR (&APPEME) TYPE (*CHAR) LEN(8)
DCL      VAR (&APPDES) TYPE (*CHAR) LEN(8)
DCL      VAR (&KEYTBT) TYPE (*CHAR) LEN(16)
DCL      VAR (&KEYUSR) TYPE (*CHAR) LEN(16)
DCL      VAR (&TYPRES) TYPE (*CHAR) LEN(10)
DCL      VAR (&IDNRES) TYPE (*CHAR) LEN(32)
DCL      VAR (&TYPTRN) TYPE (*CHAR) LEN(1)
DCL      VAR (&SUPDEM) TYPE (*CHAR) LEN(1)
DCL      VAR (&COMUSR) TYPE (*CHAR) LEN(128)
DCL      VAR (&OBJLIB) TYPE (*CHAR) LEN(10)
DCL      VAR (&OBJFIL) TYPE (*CHAR) LEN(10)
DCL      VAR (&OBJMBR) TYPE (*CHAR) LEN(10)
DCL      VAR (&PRIRES) TYPE (*CHAR) LEN(1)
DCL      VAR (&AUTHOR) TYPE (*CHAR) LEN(32)
DCL      VAR (&OBJECT) TYPE (*CHAR) LEN(256)
DCL      VAR (&ATTENT) TYPE (*CHAR) LEN(32)
DCL      VAR (&CODCTR) TYPE (*CHAR) LEN(04)
DCL      VAR (&DOMADM) TYPE (*CHAR) LEN(16)
DCL      VAR (&DOMPRV) TYPE (*CHAR) LEN(16)
DCL      VAR (&ORGANI) TYPE (*CHAR) LEN(64)
DCL      VAR (&NOMPER) TYPE (*CHAR) LEN(40)
DCL      VAR (&UNORGA) TYPE (*CHAR) LEN(32)
DCL      VAR (&DATFPC) TYPE (*CHAR) LEN(8)
DCL      VAR (&HORFPC) TYPE (*CHAR) LEN(8)
DCL      VAR (&DATFTR) TYPE (*CHAR) LEN(8)
DCL      VAR (&HORFTR) TYPE (*CHAR) LEN(8)
DCL      VAR (&DATRPC) TYPE (*CHAR) LEN(8)
DCL      VAR (&HORRPC) TYPE (*CHAR) LEN(8)
DCL      VAR (&DATRTR) TYPE (*CHAR) LEN(8)
DCL      VAR (&HORRTR) TYPE (*CHAR) LEN(8)
DCL      VAR (&ACKTBT) TYPE (*CHAR) LEN(2)
DCL      VAR (&LIBTBT) TYPE (*CHAR) LEN(128)
DCL      VAR (&NOMLOG) TYPE (*CHAR) LEN(20)
DCL      VAR (&TYPANN) TYPE (*CHAR) LEN(10)
DCL      VAR (&CUDDM) TYPE (*CHAR) LEN(32)
DCL      VAR (&FACDEM) TYPE (*CHAR) LEN(32)
DCL      VAR (&PRENOM) TYPE (*CHAR) LEN(16)
DCL      VAR (&CRDETB) TYPE (*CHAR) LEN(80)
DCL      VAR (&CODETB) TYPE (*CHAR) LEN(04)
DCL      VAR (&LIBETB) TYPE (*CHAR) LEN(70)
DCL      VAR (&RECETP) TYPE (*DEC) LEN(11 0)
DCL      VAR (&RECETPCA) TYPE (*CHAR) LEN(12)
DCL      VAR (&SNSSRV) TYPE (*CHAR) LEN(01)
DCL      VAR (&ABOSRV) TYPE (*CHAR) LEN(20)
DCL      VAR (&PSWSRV) TYPE (*CHAR) LEN(20)
DCL      VAR (&APPSRV) TYPE (*CHAR) LEN(08)

```

```

DCL      VAR(&DATSRV)   TYPE(*CHAR) LEN(06)
DCL      VAR(&ORDSRV)   TYPE(*CHAR) LEN(06)
DCL      VAR(&CRDSRV)   TYPE(*CHAR) LEN(80)
DCL      VAR(&CAREMI)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&CAREMICA) TYPE(*CHAR) LEN(12)
DCL      VAR(&CARREC)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&CARRECCA) TYPE(*CHAR) LEN(12)
DCL      VAR(&DURCOM)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&DURCOMCA) TYPE(*CHAR) LEN(12)
DCL      VAR(&CRTSRP)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&CRTSRPCA) TYPE(*CHAR) LEN(12)
DCL      VAR(&LRCSRP)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&LRCSRPCA) TYPE(*CHAR) LEN(12)
DCL      VAR(&RECSRP)   TYPE(*DEC)  LEN(11 0)
DCL      VAR(&RECSRPCA) TYPE(*CHAR) LEN(12)
DCL      VAR(&SL1SRV)   TYPE(*CHAR) LEN(08)
DCL      VAR(&SL2SRV)   TYPE(*CHAR) LEN(08)
DCL      VAR(&MULSRV)   TYPE(*CHAR) LEN(01)
DCL      VAR(&ODDSN)    TYPE(*CHAR) LEN(26)
DCL      VAR(&ODDATE)   TYPE(*CHAR) LEN(06)
DCL      VAR(&ODTIME)   TYPE(*CHAR) LEN(06)
DCL      VAR(&ODUSES)   TYPE(*CHAR) LEN(08)
DCL      VAR(&ODUSEF)   TYPE(*CHAR) LEN(08)

MONMSG   MSGID(CPF0000) EXEC(GOTO CMDLBL(CPF0000))
MONMSG   MSGID(CPF9999) EXEC(GOTO CMDLBL(CPF9999))
MONMSG   MSGID(IPS9999) EXEC(GOTO CMDLBL(IPS9999))

```

```

/*****/
/* INITIALISE ENVIRONNEMENT API DE TBT/400          */
/*****/

```

BOUCLE:

```

/* TYPE D'OBJET                                     */
/* NOM DE L'APPLICATION DESTINATRICE                */
      CHGVAR      VAR(&TYPOBJ) VALUE('M')

```

```

/*****/
/* APPEL DE LA COMMANDE DE RECEPTION                */
/*****/

```

```

IPSRVCTBT  FNCDEM(R) DEBDEM(&DEBDEM) FINDEM(C) +
            EXCDEM(N) TRADEM(0) RTNCDP(&RTNCDP) +
            TYPOBJ(&TYPOBJ) TYPMSG(&TYPMSG) +
            APPEME(&APPEME) APPDES(&APPDES) +
            KEYTBT(&KEYTBT) KEYUSR(&KEYUSR) +
            ACKTBT(&ACKTBT) LIBTBT(&LIBTBT) +
            OBJLIB(&OBJLIB) OBJFIL(&OBJFIL) +
            OBJMBR(&OBJMBR) MSGTXT(&MSGTXT) +
            TYPRES(&TYPRES) IDNRES(&IDNRES) +
            TYPTRN(&TYPTRN) DATFPC(&DATFPC) +
            HORFPC(&HORFPC) DATFTR(&DATFTR) +
            HORFTR(&HORFTR) DATRPC(&DATRPC) +
            HORRPC(&HORRPC) DATRTR(&DATRTR) +
            HORRTR(&HORRTR) SUPDEM(&SUPDEM) +
            COMUSR(&COMUSR) PRIRES(&PRIRES) +
            AUTHOR(&AUTHOR) OBJECT(&OBJECT) +
            ATTENT(&ATTENT) NOMLOG(&NOMLOG) +
            TYPANN(&TYPANN) CUDDM(&CUDDM) +
            FACDEM(&FACDEM) CAREMI(&CAREMI) +
            CARREC(&CARREC) DURCOM(&DURCOM) +
            CODCTR(&CODCTR) DOMADM(&DOMADM) +

```

```

DOMPRV (&DOMPRV)  ORGANI (&ORGANI)  +
NOMPER (&NOMPER)  PRENOM (&PRENOM)  +
UNORGA (&UNORGA)  CRDETB (&CRDETB)  +
CODETB (&CODETB)  LIBETB (&LIBETB)  +
RECETP (&RECETP)  SNSSRV (&SNSSRV)  +
CRTSRP (&CRTSRP)  LRCSRP (&LRCSRP)  +
ABOSRV (&ABOSRV)  PSWSRV (&PSWSRV)  +
APPSRV (&APPSRV)  DATSRV (&DATSRV)  +
ORDSRV (&ORDSRV)  CRDSRV (&CRDSRV)  +
RECSRP (&RECSRP)  SL1SRV (&SL1SRV)  +
SL2SRV (&SL2SRV)  MULSRV (&MULSRV)  +
ODDSN (&ODDSN)    ODDATE (&ODDATE)  +
ODTIME (&ODTIME)  ODUSES (&ODUSES)  +
ODUSEF (&ODUSEF) /* Appel des API de +
TBT/400 via la Command IPSRCVTBT */

IF          COND (&RTNCDP *NE 0 ) THEN (DO)
SNDPGMMMSG MSGID (CPF9898) MSGF (QSYS/QCPFMSG) +
           MSGDTA ('Plus de Message pour +
           l Application' *CAT &APPDES) +
           MSGTYPE (*COMP)

           GOTO          CMDLBL (ENDPGM)
ENDDO

CHGVAR     VAR (&DEBDEM)  VALUE ('N')
CHGVAR     VAR (&RECETPCA) VALUE (&RECETP)
CHGVAR     VAR (&CAREMICA) VALUE (&CAREMI)
CHGVAR     VAR (&CARRECCA) VALUE (&CARREC)
CHGVAR     VAR (&DURCOMCA) VALUE (&DURCOM)
CHGVAR     VAR (&CRTSRPCA) VALUE (&CRTSRP)
CHGVAR     VAR (&LRCSRPCA) VALUE (&LRCSRP)
CHGVAR     VAR (&RECSRPCA) VALUE (&RECSRP)

SNDPGMMMSG MSG ('MSGTXT=' *CAT &MSGTXT)
SNDPGMMMSG MSG ('TYPOBJ=' *CAT &TYPOBJ)
SNDPGMMMSG MSG ('TYPMSG=' *CAT &TYPMSG)
SNDPGMMMSG MSG ('APPEME=' *CAT &APPEME)
SNDPGMMMSG MSG ('APPDES=' *CAT &APPDES)
SNDPGMMMSG MSG ('KEYTBT=' *CAT &KEYTBT)
SNDPGMMMSG MSG ('KEYUSR=' *CAT &KEYUSR)
SNDPGMMMSG MSG ('TYPRES=' *CAT &TYPRES)
SNDPGMMMSG MSG ('IDNRES=' *CAT &IDNRES)
SNDPGMMMSG MSG ('TYPTRN=' *CAT &TYPTRN)
SNDPGMMMSG MSG ('DATFPC=' *CAT &DATFPC)
SNDPGMMMSG MSG ('HORFPC=' *CAT &HORFPC)
SNDPGMMMSG MSG ('DATFTR=' *CAT &DATFTR)
SNDPGMMMSG MSG ('HORFTR=' *CAT &HORFTR)
SNDPGMMMSG MSG ('DATRPC=' *CAT &DATRPC)
SNDPGMMMSG MSG ('HORRPC=' *CAT &HORRPC)
SNDPGMMMSG MSG ('DATRTR=' *CAT &DATRTR)
SNDPGMMMSG MSG ('HORRTR=' *CAT &HORRTR)
SNDPGMMMSG MSG ('SUPDEM=' *CAT &SUPDEM)
SNDPGMMMSG MSG ('COMUSR=' *CAT &COMUSR)
SNDPGMMMSG MSG ('ACKTBT=' *CAT &ACKTBT)
SNDPGMMMSG MSG ('LIBTBT=' *CAT &LIBTBT)
SNDPGMMMSG MSG ('OBJLIB=' *CAT &OBJLIB)
SNDPGMMMSG MSG ('OBJFIL=' *CAT &OBJFIL)
SNDPGMMMSG MSG ('OBJMBR=' *CAT &OBJMBR)
SNDPGMMMSG MSG ('PRIRES=' *CAT &PRIRES)
SNDPGMMMSG MSG ('AUTHOR=' *CAT &AUTHOR)

```

```

SNDPGMMSG MSG('OBJECT=' *CAT &OBJECT)
SNDPGMMSG MSG('ATTENT=' *CAT &ATTENT)
SNDPGMMSG MSG('NOMLOG=' *CAT &NOMLOG)
SNDPGMMSG MSG('TYPANN=' *CAT &TYPANN)
SNDPGMMSG MSG('CUDEM=' *CAT &CUDEM)
SNDPGMMSG MSG('FACDEM=' *CAT &FACDEM)
SNDPGMMSG MSG('CAREMI=' *CAT &CAREMICA)
SNDPGMMSG MSG('CARREC=' *CAT &CARRECCA)
SNDPGMMSG MSG('DURCOM=' *CAT &DURCOMCA)
SNDPGMMSG MSG('CODCTR=' *CAT &CODCTR)
SNDPGMMSG MSG('DOMADM=' *CAT &DOMADM)
SNDPGMMSG MSG('DOMPRV=' *CAT &DOMPRV)
SNDPGMMSG MSG('ORGANI=' *CAT &ORGANI)
SNDPGMMSG MSG('NOMPER=' *CAT &NOMPER)
SNDPGMMSG MSG('PRENOM=' *CAT &PRENOM)
SNDPGMMSG MSG('UNORGA=' *CAT &UNORGA)
SNDPGMMSG MSG('CRDETB=' *CAT &CRDETB)
SNDPGMMSG MSG('CODETB=' *CAT &CODETB)
SNDPGMMSG MSG('LIBETB=' *CAT &LIBETB)
SNDPGMMSG MSG('RECETP=' *CAT &RECETPCA)
SNDPGMMSG MSG('SNSRV=' *CAT &SNSRV)
SNDPGMMSG MSG('ABOSRV=' *CAT &ABOSRV)
SNDPGMMSG MSG('PSWSRV=' *CAT &PSWSRV)
SNDPGMMSG MSG('APPSRV=' *CAT &APPSRV)
SNDPGMMSG MSG('DATSRV=' *CAT &DATSRV)
SNDPGMMSG MSG('ORDSRV=' *CAT &ORDSRV)
SNDPGMMSG MSG('CRDSRV=' *CAT &CRDSRV)
SNDPGMMSG MSG('CRTSRP=' *CAT &CRTSRPCA)
SNDPGMMSG MSG('LRCSRPA=' *CAT &LRCSRPCA)
SNDPGMMSG MSG('RECSRPA=' *CAT &RECSRPCA)
SNDPGMMSG MSG('SL1SRV=' *CAT &SL1SRV)
SNDPGMMSG MSG('SL2SRV=' *CAT &SL2SRV)
SNDPGMMSG MSG('MULSRV=' *CAT &MULSRV)
SNDPGMMSG MSG('ODDSN=' *CAT &ODDSN)
SNDPGMMSG MSG('ODDATE=' *CAT &ODDATE)
SNDPGMMSG MSG('ODTIME=' *CAT &ODTIME)
SNDPGMMSG MSG('ODUSES=' *CAT &ODUSES)
SNDPGMMSG MSG('ODUSEF=' *CAT &ODUSEF)

```

```

/*****/
/* INSERER L'APPEL DE VOS TRAITEMENTS ICI */
/*****/

```

```

/* . . . . . */
/* . . . . . */
/* . . . . . */
/* . . . . . */
/* . . . . . */

```

```

/*****/
/* APPEL DE LA COMMANDE DE PURGE */
/*****/

```

```

CHGVAR VAR(&LIBTBT) VALUE('Message consommé avec +
succès')
IPSRCVTBT FNCDEM(P) DEBDEM(N) FINDEM(C) EXCDEM(O) +
TRADEM(0) RTNCDP(&RTNCDP) ACKTBT(&ACKTBT) +
LIBTBT(&LIBTBT) MSGTXT(&MSGTXT) +
KEYTBT(&KEYTBT) SUPDEM(&SUPDEM)
GOTO CMDLBL(BOUCLE)

```

```

CPF0000:
CPF9999:
IPS9999:

```

```
        SNDPGMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +  
                MSGDTA('Erreur Grave') +  
                MSGTYPE(*ESCAPE)  
ENDPGM:      ENDPGM
```

4. Annexe : Blocs de communication P0 et P1

Toutes les structures, copy, clause copy qui suivent se trouvent dans le fichier IPSSAMPLES de la bibliothèque progiciel IPLSP, sous le nom de membre précisé pour chaque objet.

4.1. Structure C (IPSILCAP)

Cette liste contient les structures P0 et P1, mais également quelques macros qui peuvent grandement faciliter la tâche de programmation.

```
#ifndef IPSILCAP
#define IPSILCAP
#include "string.h"          /* Utilisé par les Macros proposées */
/*****
/* Include pour l' API de TBT/400 */
*****/

#define FNCAPIIGN 'I'      /* Ignore */
#define FNCAPICTL 'C'     /* Controle */
#define FNCAPISND 'S'     /* Send */
#define FNCAPIRCV 'R'     /* Réception */
#define FNCAPIRMV 'V'     /* Réception et purge */
#define FNCAPITXT 'T'     /* Texte */
#define FNCAPIPUR 'P'     /* Purge */
#define FNCAPIETA 'E'     /* Etat */
#define FNCAPILAS 'L'     /* Last */
#define FNCAPIFOR 'F'     /* Force Purge */

#define TYPOBJMSG 'M'     /* Type Message Entrant */
#define TYPOBJACK 'A'     /* Type Compte Rendu */
#define TYPOBJREJ 'R'     /* Type Message Rejeté */

#define TYPMSG 'M'        /* Type Message */
#define TYPMSGPCA 'P'     /* Type Prise en Compte */
#define TYPMSGAVD 'D'     /* Type Avis de Distribution */
#define TYPMSGTBT 'T'     /* Type Compte Rendu TBT */
#define TYPMSGIMP 'I'     /* Type Compte Rendu Impression */

#define TYPACKOUI 'O'     /* Accusé Demandé */
#define TYPACKNON 'N'     /* Accusé Non Demandé */
#define TYPACKCOND 'C'    /* Accusé Demandé si Problème */

#define STATBTOUI 'O'     /* Message traité */
#define STATBTNON 'N'     /* Message non traité */

/*****
/* Partie Définition du bloc de dialogue dit P0 */
*****/

typedef struct _TBT_DIAL_S
{
    char fncdem;          /* Fonction a réaliser */
    char debdem;         /* Début demandé */
    char findem;         /* Fin demandée */
    char excdem;         /* Exception demandée */
}
```

```

char      tradem;      /* Trace      demandée      */
char      dupdem;     /* Duplication demandée     */
char      locdem;     /* Ann. local demandé       */
char      syndem;     /* Mode synchrone demandé   */
char      fil01[7];   /* --- Réservé IPLS ---    */

char      flgeof;     /* Fin de texte O/N        */
long      recnbr;     /* Nbr de record            */
long      recsiz;     /* Taille du record         */

long      nbrlig;     /* Nbr lignes buffer appli. */
long      nbrcol;     /* Nbr colonnes buffer appli.*/

long      rtncdb;     /* Code retour en Binaire   */
char      rtncdp[6];  /* Code retour en Packé     */
char      fil02[22];  /* --- Réservé IPLS ---    */

long      msgnub;     /* Numéro de message émis   */
char      msgnup[6];  /* --- Idem en Packé       */
char      msgidn[7];  /* Ident. du message émis   */
char      msgtxt[256]; /* Texte du message émis   */
char      fil03[153]; /* --- Réservé IPLS ---    */

char      pgmmsg[10]; /* --- Réservé IPLS ---    */
long      retepm;     /* --- Réservé IPLS ---    */
long      ofserr;     /* --- Réservé IPLS ---    */
long      nubloc;     /* --- Réservé IPLS ---    */
} _TBT_DIAL;

```

```

/*****
/* Partie Definition du bloc de dialogue          dit P1      */
/*****

```

```
typedef struct _TBT_API_S
```

```

{
char      typobj;     /* Type d'Objet M/R/A      */
char      typmsg;     /* Type de Message         */
short     pritbt;    /* Priorité Interne        */
char      statbt;     /* Statut du message       */

char      fil01[03]; /* ---- Réservé IPLS ---- */
long      recemi;     /* Nombre de records émis  */
long      nbremi;     /* Nombre de tentatives émis.*/
char      fatdes[9];  /* Nom de File d'attente   */
char      fatack[9];  /* Nom de File d'attente ack */
char      appeme[8];  /* Application Emettrice   */
char      appdes[8];  /* Application Réceptrice  */
char      keytbt[16]; /* --- Réservé IPLS ---    */
char      keyusr[16]; /* Ident du Message par User */
char      keyext[32]; /* Clé de Corrélation     */

char      ligtbt[10]; /* Ligne d' E/S du réseau  */

char      typres[10]; /* Type réseau de connexion */
char      idnres[32]; /* Signature dans le réseau */

char      lecjob[26]; /* Job de consommation    */
char      echnom[9];  /* ---- Réservé IPLS ---- */
char      fil02[23];  /* ---- Réservé IPLS ---- */
char      refmsg[32]; /* Référence du courrier   */

```



```

char      datper[8]; /* Date péremption   AAAAMMJJ */
char      horper[8]; /* Heure péremption  HHMMSSCC */
char      datdif[8]; /* Date env différé  AAAAMMJJ */
char      hordif[8]; /* Heur env différé  HHMMSSCC */

char      datfpc[8]; /* Date prise en     AAAAMMJJ */
char      horfpc[8]; /* compte dans fatt  HHMMSSCC */
char      datftr[8]; /* Date traitement   AAAAMMJJ */
char      horftr[8]; /*      dans fatt    HHMMSSCC */

char      datrpc[8]; /* Date prise en     AAAAMMJJ */
char      horrpc[8]; /* compte Réseau     HHMMSSCC */
char      datrtr[8]; /* Date traitement   AAAAMMJJ */
char      horrtr[8]; /*      Réseau       HHMMSSCC */

long      caremi;    /* Caractères émis          */
long      carrec;    /* Caractères reçus         */
long      durcom;    /* Durée de la communication */

long      nbrrec;    /* Nombre d'enregistrements */
long      lngrec;    /* Longueur d'un enregistr.  */
long      filsiz;    /* Taille du fichier en K    */

long      haupag;    /* Hauteur de page          */
long      recseg;    /* Enreg. par segment       */
char      typtrn;    /* Texte Transparent Oui ??? */
char      crldem;    /* Ajout                    CRLF  */
char      spadem;    /* Suppression              blancs */
char      ascdem;    /* Traduction                ASCII  */
char      ackdem;    /* Accusé Demandé           */
char      supdem;    /* Suppression Demandée     */
char      puidem;    /* Puit à l'émission        */
char      impdem;    /* Impression demandée      */

char      fil04[3]; /* ---- Réservé IPLS ----  */
char      outlib[10]; /* Bibliothèque pour impress. */
char      outfil[10]; /* Output pour impression    */
char      avidis;    /* Avis de distribution      */
char      comusr[128]; /* Zone utilisateur         */

char      usrprf[10]; /* Userid                   */
char      idnjob[26]; /* Identification du JOB     */

char      acktbt[2]; /* Accusé d'acheminement TBT */
char      libtbt[128]; /* Libellé d'acheminement TBT */

char      objlib[10]; /* Bibliothèque              */
char      objfil[10]; /* Fichier                   */
char      objmbr[10]; /* Membre                    */

char      grprrf[10]; /* Groupe d'appartenance     */
char      acdtaq[10]; /* ---- Réservé IPLS ----  */
char      datrtn[8]; /* Date dernière            AAAAMMJJ */
char      horrtn[8]; /*      tentative           HHMMSSCC */

char      fil05[14]; /* ---- Réservé IPLS ----  */

char      libann[30]; /* Libellé Annuaire         */
char      typann[10]; /* Type d'Annuaire         */

```

```

char      typloc[10]; /* Annuaire Local ou Global */
char      nomlog[20]; /* Nom logique annuaire      */

char      cudlng;    /* Longueur du C U D          */
char      cuddem[16]; /* Zone User d'appel X25      */
char      faclng;    /* Service complémentaire Lng*/
char      facdem[16]; /* Service complémentaire    */
char      tadem;     /* Taxation au demandé        */
char      dbidem;    /* Delivery Bit               */

char      prires;    /* Priorité réseau           */
char      author[32]; /* Auteur du message         */
char      object[256]; /* Objet du message          */

char      comrec[32]; /* Commutation Réception     */

char      attent[32]; /* A l 'Attention de ...     */
char      idterm[24]; /* Indicatif du terminal     */
char      numnor[16]; /* Numéro Format X121        */
char      numtlx[16]; /* Numéro Téléx              */
char      numfax[16]; /* Numéro Fax                */
char      numtpc[16]; /* Numéro Transpac           */
char      numrni[16]; /* Numéro Numéris (RNIS)    */
char      ssaadr[16]; /* Sous adresse Transpac     */
char      fil06[5];  /* ---- Réserve IPLS        */
char      aplint;    /* ---- Réserve IPLS        */
char      msgidx[128]; /* ---- Réserve IPLS        */
char      msghld;    /* ---- Réserve IPLS        */
char      msgind[32]; /* ---- Réserve IPLS        */
} _TBT_API;

```

```

/*****
/* --- Ignorer cette macro */
*****/

#define cpyblap(pre,lre,pem,lem,pad) \
( \
  ((lre) <= (lem)) \
  ?   memcpy((pre),(pem),(lre)) \
  :   (memset((pre),(pad),(lre)), \
      memcpy((pre),(pem),(lem))) \
)

/*****
/* Macro d'alimentation d'un symbole par une chaine de caractères */
/*                               littérale */
/*      Exemple : TBTLIT(symbole,"string"); */
/*                               */
*****/

#define      TBTLIT(re,pem) \
  (cpyblap((char *) &(re),sizeof((re)),(pem),(sizeof((pem))-1),' '))

/*****
/* Macro d'alimentation d'un symbole par une chaine de caractères */
/*                               */
/*      Exemple : TBTSTR(symbole,nomstring); */
/*                               */
*****/

#define      TBTSTR(re,pem) \

```

```

(cpyblap((char *) &(re),sizeof((re)),(pem),(strlen((pem))),' '))

/*****
/* Prototypes des fonctions API de TBT */
/*****

#pragma linkage (IPSSGDEB, OS)
void IPSSGDEB(void);

#pragma linkage (IPSSGFIN, OS)
void IPSSGFIN(void);

#pragma linkage (IPSSGAPI, OS)
void IPSSGAPI(_TBT_DIAL *p0, _TBT_API *p1,
              void *p2, void *p3, ...);

/*****
/* Macro d'implantation des blocs TBT */
/*****

#undef      TBTBLOCS
#define     TBTBLOCS()          \
_TBT_DIAL wtbt_p0;            \
_TBT_API  wtbt_p1;

/*****
/* Macro d'initialisation des blocs TBT */
/*****

#undef      TBTINIT
#define     TBTINIT()          \
(memset(&wtbt_p0,0,sizeof(wtbt_p0)), \
memset(&wtbt_p1,0,sizeof(wtbt_p1)))

#endif
Copy RPG (IPSIRPAP)
*****
* Include pour l' API de TBT/400 *
*****
* Les Constantes *
*****
*Ignore
I          'I'          C          APIIGN
*Controle
I          'C'          C          APICTL
*Send
I          'S'          C          APISND
*Réception
I          'R'          C          APIRCV
*Réception et Purge
I          'V'          C          APIRMV
*Texte
I          'T'          C          APITXT
*Purge
I          'P'          C          APIPUR
*Etat
I          'E'          C          APIETA

```

```

*Last
I          'L'          C          APILAS
*Forcepurge
I          'F'          C          APIFOR
*
*Type Message Entrant
I          'M'          C          OBJMSG
*Type Compte rendu
I          'A'          C          OBJACK
*Type Message rejeté
I          'R'          C          OBJREJ
*
*Type Message
I          'M'          C          MSG
*Type Prise en Compte
I          'P'          C          MSGPCA
*Type Avis de Distribution
I          'D'          C          MSGAVD
*Type Compte rendu TBT
I          'T'          C          MSGTBT
*Type Compte rendu Impression
I          'I'          C          MSGIMP
*
*Accusé demandé
I          'O'          C          ACKOUI
*Accusé non demandé
I          'N'          C          ACKNON
*Accusé demandé si problème
I          'C'          C          ACKCON
*
*Message Traité
I          'O'          C          TBTOUI
*Message non Traité
I          'N'          C          TBTNON
*
*****
* Partie Définition du bloc de dialogue                               Dit P0*      */
*****
*
IWP0      DS          512
*
* Fonction a réaliser
I          1  1  FNCDEM
* Début demandé
I          2  2  DEBDEM
* Fin demandée
I          3  3  FINDEM
* Exception demandée
I          4  4  EXCDEM
* Trace demandée
I          5  5  TRADEM
* Duplication demandée
I          6  6  DUPDEM
* Annuaire local demandé
I          7  7  LOCDEM
* Mode synchrone demandé
I          8  8  SYNDEM
* --- Réserve IPLS ---
I          9 15  FIL001
*
* Flag fin de texte

```

```

I          16 16 FLGEOF
* Nombre d'enregistrements
I          B 17 200RECNR
* Taille d'un enregistrement
I          B 21 240RECSIZ
* Nombre de lignes
I          B 25 280NBRLIG
* Nombre de colonnes
I          B 29 320NBRCOL
* Code retour en Binaire
I          B 33 360RTNCDB
* Code retour en Packé
I          P 37 420RTNCDP
* --- Réservé IPLS ---
I          43 64 FIL002
*
* Numéro de message émis
I          B 65 680MSGNUB
* --- Idem en Packé
I          P 69 740MSGNUP
* Ident. du message émis
I          75 81 MSGIDN
* Texte du message émis
I          82 337 MSGTXT
* --- Réservé IPLS ---
I          338 490 FIL003
*
* --- Réservé IPLS ---
I          491 500 PGMSG
* --- Réservé IPLS ---
I          B 501 5040RETEPM
* --- Réservé IPLS ---
I          B 505 5080OFSERR
* --- Réservé IPLS ---
I          B 509 5120NUBLOC
*
*****
* Partie Definition du bloc de dialogue          Dit P1*
*****
*
IWP1      DS          1536
*
* Type d'Objet M/R/A
I          1 1 TYPOBJ
* Type de Message
I          2 2 TYPMSG
* Priorité Interne
I          B 3 40PRITBT
* Statut du message
I          5 5 STATBT
*
* ---- Réservé IPLS ----
I          6 8 FIL101
* Nombre de records émis
I          B 9 120RECEMI
* Nombre de tentatives d'émission
I          B 13 160NBREMI
*
* Nom de File d'attente
I          17 25 FATDES
* Nom de File d'attente ack

```

I		26	34	FATAACK
	* Application Emettrice			
I		35	42	APPEME
	* Application Réceptrice			
I		43	50	APPDES
	* --- Réserve IPLS ---			
I		51	66	KEYTBT
	* Ident du Message par User			
I		67	82	KEYUSR
	* Clé de Corrélation			
I		83	114	KEYEXT
	* Ligne d' E/S du réseau			
I		115	124	LIGTBT
	* Type réseau de connexion			
I		125	134	TYPRES
	* Signature dans le réseau			
I		135	166	IDNRES
	* Job de consommation			
I		167	192	LECJOB
	* ---- Réserve IPLS ----			
I		193	201	ECHNOM
	* ---- Réserve IPLS ----			
I		202	224	FIL102
	* Référence du courrier			
I		225	256	REFMSG
	* Date péremption AAAAMMJJ			
I		257	264	DATPER
	* Heure péremption HHMMSSCC			
I		265	272	HORPER
	* Date env différé AAAAMMJJ			
I		273	280	DATDIF
	* Heur env différé HHMMSSCC			
I		281	288	HORDIF
	* Date prise en AAAAMMJJ			
I		289	296	DATFPC
	* compte dans fatt HHMMSSCC			
I		297	304	HORFPC
	* Date traitement AAAAMMJJ			
I		305	312	DATFTR
	* dans fatt HHMMSSCC			
I		313	320	HORFTR
	* Date prise en AAAAMMJJ			
I		321	328	DATRPC
	* compte Réseau HHMMSSCC			
I		329	336	HORRPC
	* Date traitement AAAAMMJJ			
I		337	344	DATRTR
	* Réseau HHMMSSCC			
I		345	352	HORRTR
	* Caractères émis			
I		B 353	3560	CAREMI
	* Caractères reçus			
I		B 357	3600	CARREC
	* Durée de la communication			

I		B 361 3640	DURCOM
*			
	* Nombre d'enregistrements		
I		B 365 3680	NBRREC
*	* Longueur d'un enregist.		
I		B 369 3720	LNGREC
*	* Taille du fichier en K		
I		B 373 3760	FILSIZ
*			
	* Hauteur de page		
I		B 377 3800	HAUPAG
*	* Enreg. par segment		
I		B 381 3840	RECSEG
*	* Texte Transparent Oui ???		
I		385 385	TYPTRN
*	* Ajout CRLF		
I		386 386	CRLDEM
*	* Suppression blancs		
I		387 387	SPADEM
*	* Traduction ASCII		
I		388 388	ASCDEM
*	* Accusé Demandé		
I		389 389	ACKDEM
*	* Suppression Demandée		
I		390 390	SUPDEM
*	* Fonction puit lors de l'émission sur le réseau		
I		391 391	PUIDEM
*	* Impression demandée		
I		392 392	IMPDEM
*			
	* ---- Réserve IPLS ----		
I		393 395	FIL104
*			
	* Bibliothèque pour impres.		
I		396 405	OUTLIB
*	* Output pour impression		
I		406 415	OUTFIL
	* Avis de distribution		
I		416 416	AVIDIS
*	* Zone utilisateur		
I		417 544	COMUSR
*	* Userid		
I		545 554	USRPRF
*	* Identification du JOB		
I		555 580	IDNJOB
*			
	* Accusé d'acheminement TBT		
I		581 582	ACKTBT
*	* Libellé d'acheminement TBT		
I		583 710	LIBTBT
*			
	* Bibliothèque		
I		711 720	OBJLIB
*	* Fichier		
I		721 730	OBJFIL
*	* Membre		
I		731 740	OBJMBR
*	* Groupe d'appartenance		
I		741 750	GRPPRF

```

*
* ---- Réserve IPLS ---
I          751 760 ACDTAQ
* ---- Réserve IPLS ---
I          761 774 FIL105
* Date dernière      AAAAMMJJ
I          775 782 DATRTN
*      tentative     HHMMSSCC
I          783 790 HORRTN
*
* Libellé de l'annuaire
I          791 820 LIBANN
* Type de l'annuaire
I          821 830 TYPANN
* Annuaire local ou global (High Value ==> Global)
I          831 840 TYPLOC
* Nom logique de l'annuaire
I          841 860 NOMLOG
* Longueur du CUD
I          861 861 CUDLNG
* Zone user d'appel X25
I          862 877 CUDEM
* Longueur service complémentaire
I          878 878 FACLNG
* Service complémentaire
I          879 894 FACDEM
* Taxation au demandé
I          895 895 TADDEM
* Delivery Bit
I          896 896 DBIDEM
* Priorité réseau
I          897 897 PRIRES
* Auteur du message
I          898 929 AUTHOR
* Objet du message
I          9301185 OBJECT
*
*
* Commutation Réception
I          11861217 COMREC
*
* A l 'Attention de ...
I          12181249 ATTENT
* Indicatif du terminal
I          12501273 IDTERM
* Numéro Format X121
I          12741289 NUMNOR
* Numéro Téléx
I          12901305 NUMTLX
* Numéro Fax
I          13061321 NUMFAX
* Numéro Transpac
I          13221337 NUMTPC
* Numéro Numéris (RNIS)
I          13381353 NUMRNI
* Sous-adresse Transpac
I          13541369 SSAADR
* ---- Réserve IPLS
I          13701374 FIL106
* ---- Réserve IPLS

```



```

I          13751375 APLINT
* ---- Réservé IPLS
I          13761503 MSGIDX
* ---- Réservé IPLS
I          15041504 MSGHLD
* ---- Réservé IPLS
I          15051536 MSGIND
*

```

Clause copy cobol (IPSICBAP)

```

*****
* Include pour l' API de TBT/400 *
*****
* Les Constantes *
*****
*Ignore
 77      APIIGN
          PIC X VALUE "I".
*Contrôle
 77      APICTL
          PIC X VALUE "C".
*Send
 77      APISND
          PIC X VALUE "S".
*Réception
 77      APIRCV
          PIC X VALUE "R".
*Réception et purge
 77      APIETA
          PIC X VALUE "V".
*Texte
 77      APITXT
          PIC X VALUE "T".
*Purge
 77      APIPUR
          PIC X VALUE "P".
*Etat
 77      APIETA
          PIC X VALUE "E".
*Last
 77      APILAS
          PIC X VALUE "L".
*Forcepurge
 77      APIFOR
          PIC X VALUE "F".
*
*Type Message Entrant
 77      OBJMSG
          PIC X VALUE "M".
*Type Compte rendu
 77      OBJACK
          PIC X VALUE "A".
*Type Message rejeté
 77      OBJREJ
          PIC X VALUE "R".
*
*Type Message
 77      MSG
          PIC X VALUE "M".
*Type Prise en Compte

```

```

77      MSGPCA
          PIC X VALUE "P".
*Type Avis de Distribution
77      MSGAVD
          PIC X VALUE "D".
*Type Compte rendu TBT
77      MSGTBT
          PIC X VALUE "T".
*Type Compte rendu Impression
77      MSGIMP
          PIC X VALUE "I".
*
*Accusé demandé
77      ACKOUI
          PIC X VALUE "O".
*Accusé non demandé
77      ACKNON
          PIC X VALUE "N".
*Accusé demandé si problème
77      ACKCON
          PIC X VALUE "C".
*
*Message Traité
77      TBTTOUI
          PIC X VALUE "O".
*Message non Traité
77      TBTNON
          PIC X VALUE "N".
*
*****
* Partie Définition du bloc de dialogue                               dit P0*
*****
01  WP0.
*
* Fonction a réaliser
    02  P0-FNCDEM                                PIC X(0001).
* Début demandé
    02  P0-DEBDEM                                PIC X(0001).
* Fin demandée
    02  P0-FINDEM                                PIC X(0001).
* Exception Demandée
    02  P0-EXCDEM                                PIC X(0001).
* Trace demandée
    02  P0-TRADEM                                PIC X(0001).
* Duplication demandée
    02  P0-DUPDEM                                PIC X(0001).
* Annuaire local demandé
    02  P0-LOCDEM                                PIC X(0001).
* Mode synchrone demandé
    02  P0-SYNDEM                                PIC X(0001).
* --- Réserve IPLS ---
    02  P0-FIL001                                PIC X(0007).

```

```

* Flag fin de texte
  02 P0-FLGEOF          PIC X(0001).

* Nombre d'enregistrements
  02 P0-RECNR          PIC 9(008) BINARY.

* Taille de l'enregistrement
  02 P0-RECSIZ        PIC 9(008) BINARY.

* Nombre de lignes
  02 P0-NBRLIG        PIC 9(008) BINARY.
* Nombre de colonnes
  02 P0-NBRCOL        PIC 9(008) BINARY.
* Code retour en Binaire
  02 P0-RTNCDB        PIC 9(008) BINARY.
* Code retour en Packé
  02 P0-RTNCDP        PIC S9(011) COMP-3.
*--- Réserve IPLS ---
  02 P0-FIL002        PIC X(0022).
*
* Numéro de message émis
  02 P0-MSGNUB        PIC 9(008) BINARY.
* Idem en Packé
  02 P0-MSGNUP        PIC S9(011) COMP-3.
* Ident. du message émis
  02 P0-MSGIDN        PIC X(0007).
* Texte du message émis
  02 P0-MSGTXT        PIC X(0256).

* --- Réserve IPLS ---
  02 P0-FIL003        PIC X(0153).
* --- Réserve IPLS ---
  02 P0-PGMSG         PIC X(0010).
* --- Réserve IPLS ---
  02 P0-RETEPM        PIC 9(008) BINARY.
* --- Réserve IPLS ---
  02 P0-OFSERR        PIC 9(008) BINARY.
* --- Réserve IPLS ---
  02 P0-NUBLOC        PIC 9(008) BINARY.
*
*****
* Partie Définition du bloc de dialogue          dit P1*
*****
*
  01 WP1.
*
* Type d'Objet M/R/A
  02 P1-TYPOBJ        PIC X(0001).
* Type de Message
  02 P1-TYPMSG        PIC X(0001).
* Priorité Interne
  02 P1-PRITBT        PIC 9(004) BINARY.
* Statut du message
  02 P1-STATBT        PIC X(0001).
*
* --- Réserve IPLS ---
  02 P1-FIL101        PIC X(0003).
*
* Nombre de records émis
  02 P1-RECEMI        PIC 9(008) BINARY.
* Nombre de tentatives d'émission

```

```

02 P1-NBREMI PIC 9(008) BINARY.
*
* Nom de File d'attente
02 P1-FATDES PIC X(0009).
* Nom de File d'attente ack
02 P1-FATAACK PIC X(0009).
* Application Emettrice
02 P1-APPEME PIC X(0008).
* Application Réceptrice
02 P1-APPDES PIC X(0008).
* --- Réservé IPLS ---
02 P1-KEYTBT PIC X(0016).
* Ident du Message par User
02 P1-KEYUSR PIC X(0016).
* Clé de Corrélation
02 P1-KEYEXT PIC X(0032).
*
* Ligne d' E/S du réseau
02 P1-LIGTBT PIC X(0010).
*
* Type réseau de connexion
02 P1-TYPRES PIC X(0010).
* Signature dans le réseau
02 P1-IDNRES PIC X(0032).
*
* Job de consommation
02 P1-LECJOB PIC X(0026).
*
* ---- Réservé IPLS ----
02 P1-ECHNOM PIC X(0009).
* ---- Réservé IPLS ----
02 P1-FIL102 PIC X(0023).
* Référence du courrier
02 P1-REFMSG PIC X(0032).
*
* Date péremption AAAAMMJJ
02 P1-DATPER PIC X(0008).
* Heure péremption HHMMSSCC
02 P1-HORPER PIC X(0008).
* Date env différé AAAAMMJJ
02 P1-DATDIF PIC X(0008).
* Heur env différé HHMMSSCC
02 P1-HORDIF PIC X(0008).
*
* Date prise en AAAAMMJJ
* compte dans fatt HHMMSSCC
02 P1-DATFPC PIC X(0008).
02 P1-HORFPC PIC X(0008).
* Date traitement AAAAMMJJ
* dans fatt HHMMSSCC
02 P1-DATFTR PIC X(0008).
02 P1-HORFTR PIC X(0008).
*
* Date prise en HHMMSSCC
* compte Réseau HHMMSSCC
02 P1-DATRPC PIC X(0008).
02 P1-HORRPC PIC X(0008).
* Date traitement AAAAMMJJ
* Réseau HHMMSSCC
02 P1-DATRTR PIC X(0008).
02 P1-HORRTR PIC X(0008).

```

```

*
* Caractères émis
    02 P1-CAREMI PIC 9(0008) BINARY.
* Caractères reçus
    02 P1-CARREC PIC 9(0008) BINARY.
* Durée de la communication (en secondes)
    02 P1-DURCOM PIC 9(0008) BINARY.
*
* Nombre d'enregistrements
    02 P1-NBRREC PIC 9(0008) BINARY.
* Longueur d'un enregist.
    02 P1-LNGREC PIC 9(0008) BINARY.
* Taille du fichier en K
    02 P1-FILSIZ PIC 9(0008) BINARY.
*
* Hauteur de page
    02 P1-HAUPAG PIC 9(0008) BINARY.
* Enreg. par segment
    02 P1-RECSEG PIC 9(0008) BINARY.
* Texte Transparent Oui ???
    02 P1-TYPTRN PIC X(0001).
* Ajout CRLF
    02 P1-CRLDEM PIC X(0001).
* Suppression blancs
    02 P1-SPADEM PIC X(0001).
* Traduction ASCII
    02 P1-ASCDEM PIC X(0001).
* Accusé Demandé
    02 P1-ACKDEM PIC X(0001).
* Suppression Demandée
    02 P1-SUPDEM PIC X(0001).
* Fonction puit à l'émission sur le réseau
    02 P1-PUIDEM PIC X(0001).
* Impression demandée
    02 P1-IMPDEM PIC X(0001).
*
* ---- Réserve IPLS ----
    02 P1-FIL104 PIC X(0003).
*
* Bibliothèque pour impres.
    02 P1-OUTLIB PIC X(0010).
* Output pour impression
    02 P1-OUTFIL PIC X(0010).
*
* Avis de distribution
    02 P1-AVIDIS PIC X(0001).
*
* Zone utilisateur
    02 P1-COMUSR PIC X(0128).
*
* Userid
    02 P1-USRPRF PIC X(0010).
* Identification du JOB
    02 P1-IDNJOB PIC X(0026).
*
* Accusé d'acheminement TBT
    02 P1-ACKTBT PIC X(0002).
* Libellé d'acheminement TBT
    02 P1-LIBTBT PIC X(0128).
*
* Bibliothèque

```

```

02 P1-OBJLIB          PIC X(0010).
* Fichier
02 P1-OBJFIL          PIC X(0010).
* Membre
02 P1-OBJMBR          PIC X(0010).
* Groupe d'appartenance
02 P1-GRPPRF          PIC X(0010).
*
* ---- Réserve IPLS ---
02 P1-ACDTAQ          PIC X(0010).
* ---- Réserve IPLS ---
02 P1-FILL105         PIC X(0014).
*
* Date dernière      AAAAMMJJ
* tentative          HHMMSSCC
02 P1-DATRTN          PIC X(0008).
02 P1-HORRTN          PIC X(0008).
*
* Libellé annuaire
02 P1-LIBANN          PIC X(0030).
* Type d'annuaire
02 P1-TYPANN          PIC X(0010).
* Annuaire local ou global (High Value ==> Global)
02 P1-TYPLOC          PIC X(0010).
* Nom logique dans l'annuaire
02 P1-NOMLOG          PIC X(0020).
* Longueur du CUD
02 P1-LNGCUD          PIC X(0001).
* Zone user d'appel X25
02 P1-CUDEDEM          PIC X(0016).
* Longueur service complémentaire
02 P1-FACLNG          PIC X(0001).
* Service complémentaire
02 P1-FACDEM          PIC X(0016).
* Taxation au demandé
02 P1-TADDEM          PIC X(0001).
* Delivery Bit
02 P1-DBIDEM          PIC X(0001).
*
* Priorité réseau
02 P1-PRIRES          PIC X(0001).
* Auteur du message
02 P1-AUTHOR          PIC X(0032).
* Objet du message
02 P1-OBJECT          PIC X(0256).
* Commutation Réception
02 P1-COMREC          PIC X(0032).
*
* A l 'Attention de ...
02 P1-ATTENT          PIC X(0032).
* Indicatif du terminal
02 P1-IDTERM          PIC X(0024).
* Numéro Format X121
02 P1-NUMNOR          PIC X(0016).
* Numéro Téléx
02 P1-NUMTLX          PIC X(0016).
* Numéro Fax
02 P1-NUMFAX          PIC X(0016).
* Numéro Transpac
02 P1-NUMTPC          PIC X(0016).
* Numéro Numéris (RNIS)

```

02 P1-NUMRNI	PIC X(0016).
* Sous-adresse Transpac	
02 P1-SSAADR	PIC X(0016).
* ---- Réservé IPLS	
02 P1-FIL106	PIC X(0005).
* ---- Réservé IPLS	
02 P1-APLINT	PIC X(0001).
* ---- Réservé IPLS	
02 P1-MSGIDX	PIC X(0128).
* ---- Réservé IPLS	
02 P1-MSGHLD	PIC X(0001).
* ---- Réservé IPLS	
02 P1-MSGIND	PIC X(0032).
*	

5. ANNEXE : INDEXE DES NOMS DE CHAMPS DES BLOCS DE COMMUNICATION

ACKDEM, 15	HORFPC, 13
ACKTBT, 16	HORFTR, 13
APPDES, 12	HORPER, 13
APPEME, 11	HORRPC, 13
ASCDEM, 15	HORRTN, 17
ATTENT, 18	HORRTR, 13
AUTHOR, 18	IDNJOB, 16
AVIDIS, 16	IDNRES, 12
CAREMI, 14	IDTERM, 18
CARREC, 14	IMPDEM, 16
COMREC, 18	KEYEXT, 12
COMUSR, 16	KEYTBT, 12
CRLDEM, 15	KEYUSR, 12
CUDDM, 17	LECJOB, 12
CUDLNG, 17	LIBANN, 17
DATDIF, 13	LIBTBT, 17
DATFPC, 13	LIGTBT, 12
DATFTR, 13	LNGREC, 14
DATPER, 12	LOCDEM, 9
DATRPC, 13	MSGIDN, 10
DATRTR, 17	MSGNUB, 10
DATRTR, 13	MSGNUP, 10
DBIDEM, 18	MSGTXT, 10
DEBDEM, 8	NBRCOL, 9
DUPDEM, 9	NBREMI, 11
DURCOM, 14	NBRLIG, 9
EXCDEM, 8	NBRREC, 14
FACDEM, 18	NOMLOG, 17
FACLNG, 17	NUMFAX, 18
FATAACK, 11	NUMNOR, 18
FATDES, 11	NUMRNI, 19
FILSIZ, 14	NUMTLX, 18
FINDM, 8	NUMTPC, 19
FLGEOF, 9	OBJECT, 18
FNCDEM, 8	OBJFIL, 17
GRPPRF, 17	OBJLIB, 17
HAUPAG, 14	OBJMBR, 17
HORDIF, 13	OUTFIL, 16

OUTLIB, 16	SPLSUP, 10
PRIRES, 18	SSAADR, 19
PRITBT, 11	STATBT, 11
PUIDEM, 16	SUPDEM, 15
RECEMI, 11	SYNDEM, 9
RECNBR, 9	TADDEM, 18
RECSEG, 14	TRADEM, 8
RECSIZ, 9	TYPANN, 17
REFMSG, 12	TYPLOC, 17
RTNCDB, 10	TYPMSG, 11
RTNCDP, 10	TYPOBJ, 11
SPADEM, 15	TYPRES, 12
SPLJOB, 10	TYPTRN, 14
SPLNAM, 10	USRPRF, 16
SPLNUM, 10	

6. Annexe : Statut du message

Quand un message est géré par **TBT/400**, il passe par différentes étapes, qui sont matérialisées par le renseignement de champs particuliers. Cette annexe précise les statuts du message pendant son cheminement à travers **TBT/400**.

6.1. En émission

Dépôt en file d'attente par l'API:

STATBT = N.
DATFPC et HORFPC = date et heure d'écriture dans la file d'attente par l'API.
ACKTBT = Low Value.

Connexion et envoi à Atlas:

DATFTR et HORFTR = date et heure d'envoi par **TBT/400**.
ACKTBT = PC si prise en compte par Atlas,
RF ou autre si refus par Atlas.
LIBTBT = "Pris en compte" si prise en compte par Atlas,
Selon code de ACKTBT (voir table des erreurs réseau) si refus par Atlas.
DATRPC et HORRPC = date et heure de prise en compte par Atlas si non refusé.
STATBT = O.

Historisation du message par **TBT/400**.

Remontée de la prise en compte ou du refus vers l'applicatif si ACKDEM = Oui ou Conditionnel.

Si avis de non distribution par Atlas:

Mise à jour de l'historique avec:
ACKTBT = ED.
LIBTBT = Raison et diagnostic renseignés par Atlas.
ATRTR et HORRTR = date et heure de l'avis de non distribution par Atlas.

Remontée de l'avis de non distribution vers l'applicatif si ACKDEM = Oui ou Conditionnel.

Si avis de distribution positif par Atlas (seulement si AVIDIS = Oui):

Mise à jour de l'historique avec :
ACKTBT = blanc.
LIBTBT = "Distribué par Atlas 400".
DATRTR et HORRTR = date et heure de distribution renseignées par Atlas.

Remontée de l'avis de distribution vers l'applicatif si ACKDEM = Oui.

6.2. En réception

Dépôt par **TBT/400** dans la file d'attente avec:

STATBT = N.
DATFPC et HORFPC, DATRTR et HORRTR = date et heure de dépôt du message par Atlas.

Lecture du message par l'API: pas de changement.

Fonction purge demandée par l'applicatif:

STATBT = O.
DATFTR et HORFTR = date et heure de la purge.

Historisation du message par **TBT/400**.

7. Annexe : Numérotation Téléphonique, Fax et Téléx

7.1. Numérotation Téléphonique ou Fax

Celle-ci doit être saisie sous la forme :

- numéro national pour le réseau français (8 caractères), précédé de 1 pour un destinataire en Ile de France,
- code téléphonique du pays + numéro national pour le réseau international.

Par exemple, le code téléphonique de la France est 33.

7.2. Numérotation Téléx

7.2.1. Numéro Téléx

Celui-ci doit être saisi sous la forme :

- numéro national pour le réseau français (6 caractères),
- code télégraphique du pays + numéro national pour le réseau international.

Par exemple, le code télégraphique de la France est 42.

7.2.2. Indicatif Téléx

Ceci est une notion fondamentale en terminologie Téléx. En effet, lors de la composition d'un numéro Téléx, il n'est pas rare d'obtenir une "erreur de commutation". En conséquence, le Téléx émis peut l'être vers un destinataire erroné.

Le moyen de l'éviter est de se protéger par un contrôle d'indicatif. Sur le réseau Téléx, conventionnellement, sur réception du code "who are you" (croix de malte), le terminal doit envoyer un indicatif qui lui est propre.

En émission, toute communication débute par l'envoi du "Who Are You". L'indicatif reçu en réponse est comparé caractère par caractère avec l'indicatif de contrôle spécifié (le contrôle s'effectue de gauche à droite sur la seule partie renseignée de l'indicatif, le caractère X étant un masque générique **(1)**).

Exemple

Numéro 123456 Indicatif ABCDE

Indicatif reçu ABCDEF 123456 OK

Indicatif reçu ABDCEF 123456 KO

Numéro 123456 Indicatif ABXXE

Indicatif reçu ABCDEF 123456 OK

Indicatif reçu ABDDEF 123456 OK

Sur le réseau français, l'indicatif est en général formé de deux parties, une partie lettres (mnémonique), et une partie chiffres rappelant le numéro d'appel.

Sur le réseau international, l'indicatif est en général formé de deux parties, une partie lettres (mnémonique), et une partie chiffres rappelant le numéro d'appel national. Mais cette règle comporte nombre d'exceptions, la partie chiffres pouvant précéder la partie lettres. Il est impératif de coder de gauche à droite ce qui est réellement reçu.

(1) Ce caractère est utilisé pour masquer les indicatifs des lignes groupées, ou d'une ligne à l'autre un ou plusieurs caractères peuvent différer.

Certains automates en réception ne respectent pas l'obligation de réponse au Who Are You. Dans ce cas, il faut pour pouvoir émettre le message, coder un "." dans le premier caractère de la zone indicatif (pas de Who Are You). Dans ce cas le message ne peut être sécurisé.

Les critères d'identification du correspondant sont également repris pour le traitement des messages entrants. Un minimum de **cinq caractères significatifs** doit avoir été spécifié.

7.2.3. Codes des incidents Téléx

Code	Signification du code incident
FF	Ligne coupée (alimentation défectueuse)
PP	Ligne (re)validée (suite à un incident PTT)
CO	Echange durée de la communication en cours
TT	Transmission terminée
IP	Communication interrompue (Incident central PTT)
AE	Début de réception d'un message
AB	Correspondant absent
DO	Poste téléx en dérangement
GA	Incident central
ID	Indicatif différent
NC	Pas de circuit
NP	Poste résilié
OO	Poste occupé
PM	Position manuel
RE	Communication refusée
NH	Numéro transféré
CI	Pas de réponse de compte rendu de fin d'émission
IF	Indicatif faux lors du compte rendu de fin de message Le message est revalidé (statut = 2)

Dans le cadre des améliorations apportées au progiciel, **IPLS** se réserve la possibilité de modifier certaines des fonctions décrites ci-dessus.

Vos commentaires nous permettent d'améliorer la qualité de nos publications; ils jouent un rôle important lors des mises à jour. Si vous avez des observations sur le document ci-joint, nous vous serions reconnaissants de nous en faire part en utilisant cette feuille et en indiquant, le cas échéant, le numéro des pages et des lignes concernées. **IPLS** pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie de ces informations que, de votre côté, vous pourrez évidemment continuer à exploiter.

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	

Si vous désirez une réponse, n'oubliez pas de mentionner vos nom et adresse. Nous vous remercions de votre collaboration.

Toutes les marques citées sont des marques déposées.

