

1.	Présentation générale	4
1.1.	Objectifs	4
1.2.	Sites d'information	5
1.2.1.	Site de l'éditeur	5
1.2.2.	Site du progiciel TBT/400	5
1.3.	Architecture générale	6
1.3.1.	Le noyau	6
1.3.2.	Les pilotes	6
1.3.3.	Les API	6
1.3.4.	La supervision	6
1.4.	Schéma de fonctionnement	7
1.5.	Description du fonctionnement	8
1.6.	Schéma d'utilisation des API	9
1.7.	Schéma de fonctionnement en émission	10
1.8.	Schéma de fonctionnement en réception	11
1.9.	Concepts	12
1.9.1.	Application	12
1.9.2.	File d'attente	12
1.9.3.	Exemple	13
2.	Fonctionnalités	14
2.1.	Fonctionnalités fichiers	15
2.1.1.	Types de fichiers supportés	15
2.1.2.	Modes d'Accès aux fichiers	15
2.1.3.	Encodage CMS	15
2.2.	Fonctionnalités réseaux/protocoles	17
2.2.1.	ATLAS 400 - ALLEGRO	17
2.2.2.	X400 : Allegro Calvacom Carrefour Diva	17
2.2.3.	INTERNET	18
2.2.4.	GRAPHNET	19
2.2.5.	ODETTE - FTP - GEIS - IBM GN - GALIA	19
2.2.6.	PeSIT	19
2.2.7.	FTP	20
2.2.8.	Télex ou Télécopie	20
2.2.9.	Remote ETEBAC 3	20

2.2.10.	Serveur de fichiers	22
2.2.11.	Protocole interne - Télémaintenance	22
2.3.	Fonctionnalités annuaire	22
2.3.1.	Annuaire multi protocoles	22
2.3.2.	Contrôle d'adresse X25 et TCP/IP	22
2.3.3.	Contrôle de sous adresse X25	22
2.3.4.	Contrôle d'accès aux applications	23
2.3.5.	Création automatique d'entrées dans l'annuaire	23
2.4.	Fonctions de Supervision	23
2.4.1.	Menus de supervision	23
2.4.2.	Messages Queues	24
2.4.3.	Output Queues	24
2.4.4.	Vue OS/400	24
2.4.5.	Remontées d'alertes	24
2.5.	Interfaces applicatives	25
2.5.1.	Emission	25
2.5.2.	Remontées applicatives	25
2.5.3.	Réception	25
2.6.	Aides à la programmation	26
2.6.1.	Générateur de commandes : Scan d'objets	26
2.6.2.	Générateur de commandes : Scan de spools	26
2.7.	Fonctionnalités diverses	26
2.7.1.	Echéancier	26
2.7.2.	Historisation	26
2.7.3.	Purge automatique	26
2.7.4.	Gestion dynamique des menus	27
2.7.5.	Aides en ligne	27
2.7.6.	Editeur intégré	27
2.7.7.	Automate d'installation	27
2.7.8.	Fonctions utilisateur	27
2.7.9.	Bannières	27
2.8.	Passerelles avec traducteur ou messagerie	28
2.9.	Exemple de Serveur ETEBAC 1-2 ou ETEBAC 3	29
2.9.1.	Sécurité des accès	29
2.9.2.	Identification applicative	29
2.9.3.	Mise à disposition d'un fichier	29
2.9.4.	Traitement d'un appel entrant	30

2.9.5.	Double signature ou confirmation d'ordre	30
3.	Installation	31
3.1.	Réquisits	31
3.1.1.	Logiciels	31
3.1.2.	Matériels	31
3.1.3.	Connexions	31
3.2.	Bibliothèques du système	32
3.2.1.	Bibliothèque IPLSP	32
3.2.2.	Bibliothèque IPLSC	32
3.2.3.	Bibliothèque IPLSE	32
3.2.4.	Bibliothèque IPLSM	32
3.3.	Sous-système TBT/400	33
3.4.	Procédure d'installation	34
3.4.1.	Procédure automatique	34
3.4.2.	Procédure semi automatique	34
3.5.	Schéma d'ensemble de l'installation	35
3.6.	Sécurité TBT/400	36
4.	Exemples de programmes	37
4.1.	Emission d'un télex par Transpac	37
4.2.	Emission d'un fax par ATLAS 400	39
4.3.	Réception d'un message (En C)	40
4.4.	Réception d'un message (EN CLP)	41

1. Présentation générale

1.1. Objectifs

TBT/400 est une plate-forme logicielle assurant la gestion, le suivi et le dialogue avec vos différents moyens de communication externes, en assurant leur indépendance vis à vis de vos différents processus informatiques.

TBT/400 est naturellement intégré à l'architecture de l'**IBM AS/400**, et respecte les spécifications de **SAA**.

TBT/400 utilise exclusivement OS/400, et ne nécessite pas de logiciels particuliers (pas même **PDM**).

TBT/400, c'est l'opportunité de posséder au sein d'une plate-forme unique des accès sécurisés à des réseaux et protocoles multiples comme:

ATLAS 400	Odette FTP	Réseau téléphonique,
ALLEGRO	PeSIT	Transpac en X25 ou X32
GEIS	ETEBAC	RNIS
IBM Network	X400	TCP/IP
Calvacom	EBICS	Canal D Numeris
Diva	SMTP	Graphnet
Télex	FTP	GALIA
Télécopie		TEDECO

pour communiquer avec tous vos partenaires:

CLIENTS	TRANSPORTEURS
FOURNISSEURS	AGENCES
BANQUES	DEPOTS
ADMINISTRATION	USINES

TBT/400 est une plate-forme **EDI** qui s'intègre à vos architectures applicatives et à votre traducteur **EDI** (**EDI400™**, **EDITRADE™**, **EDIBASE™**, **GENEDI™**, **EDIMANAGER™**,...) afin de participer à des solutions logicielles en "workflow".

TBT/400 permet la mise en relation d'un applicatif AS/400 avec un applicatif externe (exemple: un télécopieur, une télécopie, une autre application, ...) en fournissant une logique applicative (ou **API**) simple, et indépendante du réseau de transmission et du protocole utilisés.

De par sa structure et via ses **API**, **TBT/400** permet l'intégration de plates-formes clientes du type PC/PS.

TBT/400 est donc un commutateur de messages entre applications, les réseaux étant considérés eux-mêmes comme des applications classiques.

TBT/400 participe à l'évolution de vos communications externes vers de nouveaux partenaires, de nouveaux réseaux et de nouveaux protocoles.

TBT/400 est un progiciel en constante évolution; aussi n'hésitez pas à nous consulter sur les réseaux et/ou interfaces dont vous avez besoin.

1.2. Sites d'information

IPLS met à disposition plusieurs sites Internet d'information et de téléchargement.

1.2.1. Site de l'éditeur

<http://www.ipls.fr> présente la société **IPLS** et ses produits.

1.2.2. Site du progiciel TBT/400

<http://www.tbt400.com> présente le produit **TBT/400** et vous donne toute l'information actualisée.

1.3. Architecture générale

TBT/400 se compose de quatre éléments logiciels:

1.3.1. Le noyau

Commun à toute installation, il surveille les différents éléments du produit. Il gère les files d'attente, les accès à celles-ci, les tâches de réception et les différents "pilotes" de communication.

Il se concrétise par plusieurs fonctions principales :

- ⇒ **La gestion des files d'attente** (spouler) pour les fichiers reçus ou à émettre.
- ⇒ **Le déclenchement** (automatique ou manuel) et la surveillance de tâches (programmes clients) de communication avec la plate-forme.
- ⇒ **La gestion d'un historique** complet de tous les flux gérés par la plate-forme.
- ⇒ **Périodiquement, l'épuration automatique** sur les différents composants **TBT/400** (Messages Queues, Output Queues, fichier historique, fichiers émis ou reçus...).

1.3.2. Les pilotes

Ils permettent la gestion de la communication. Ils sont spécifiques à chacun des réseaux auxquels ils doivent accéder (**ATLAS 400**, **Graphnet**, **Odette**, **PeSIT**, **X400**, **FTP**, **Réseau à Valeur Ajoutée**, ...). Ils sont indépendants les uns des autres, et sont sous la tutelle du noyau. Ceci permet l'utilisation de nouveaux accès réseau, par la mise en place de pilotes supplémentaires, sur des installations déjà existantes.

1.3.3. Les API

La couche **API** permet l'interfaçage de vos applicatifs avec l'architecture de **TBT/400**.

Les **API** offrent différents niveaux d'accès afin de faciliter au maximum leur mise en place. Ces **API** sont accessibles à partir de tout **L3G/L4G** ou **CL** avec un point d'accès unique.

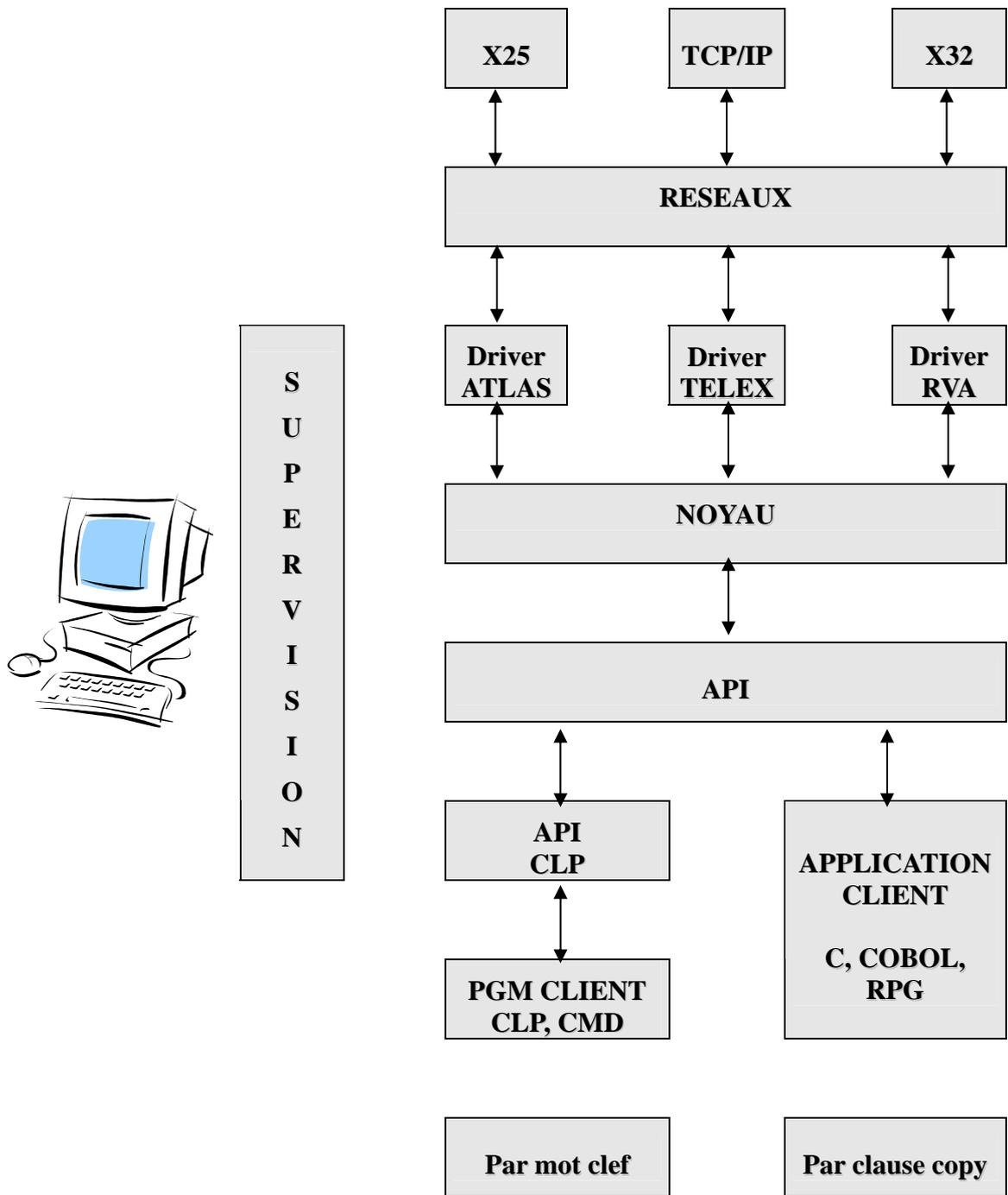
1.3.4. La supervision

C'est la couche de présentation, d'initialisation et de suivi de **TBT/400**.

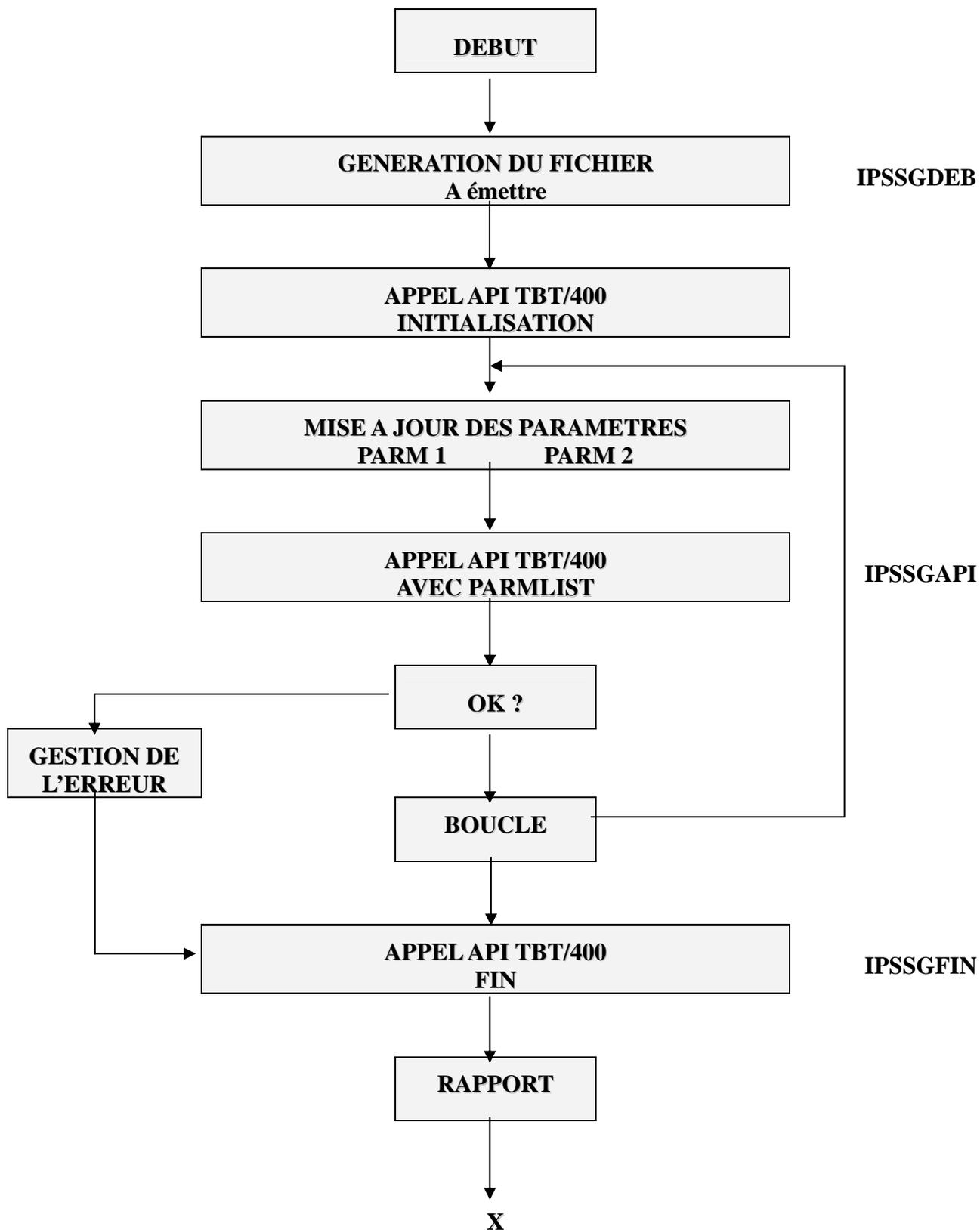
Elle permet la mise en place des différentes options du produit, puis, dans sa phase d'exploitation, le suivi du trafic, la gestion des différents flux (fichiers entrants, fichiers sortants, ...), ainsi que le suivi du progiciel lui-même.

Elle comprend également un éditeur intégré, permettant la saisie et l'émission de messages en direct par les utilisateurs.

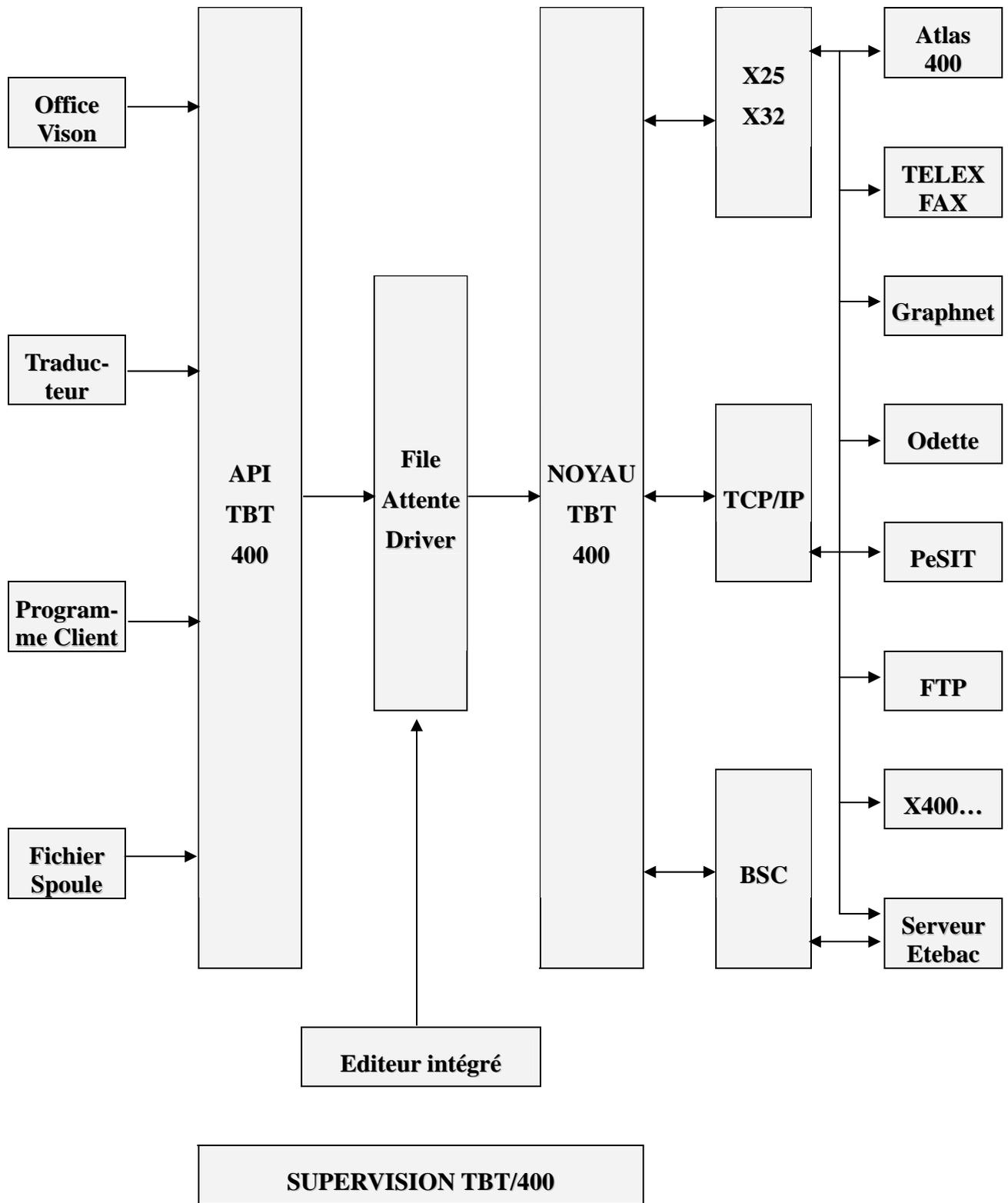
1.4. Schéma de fonctionnement



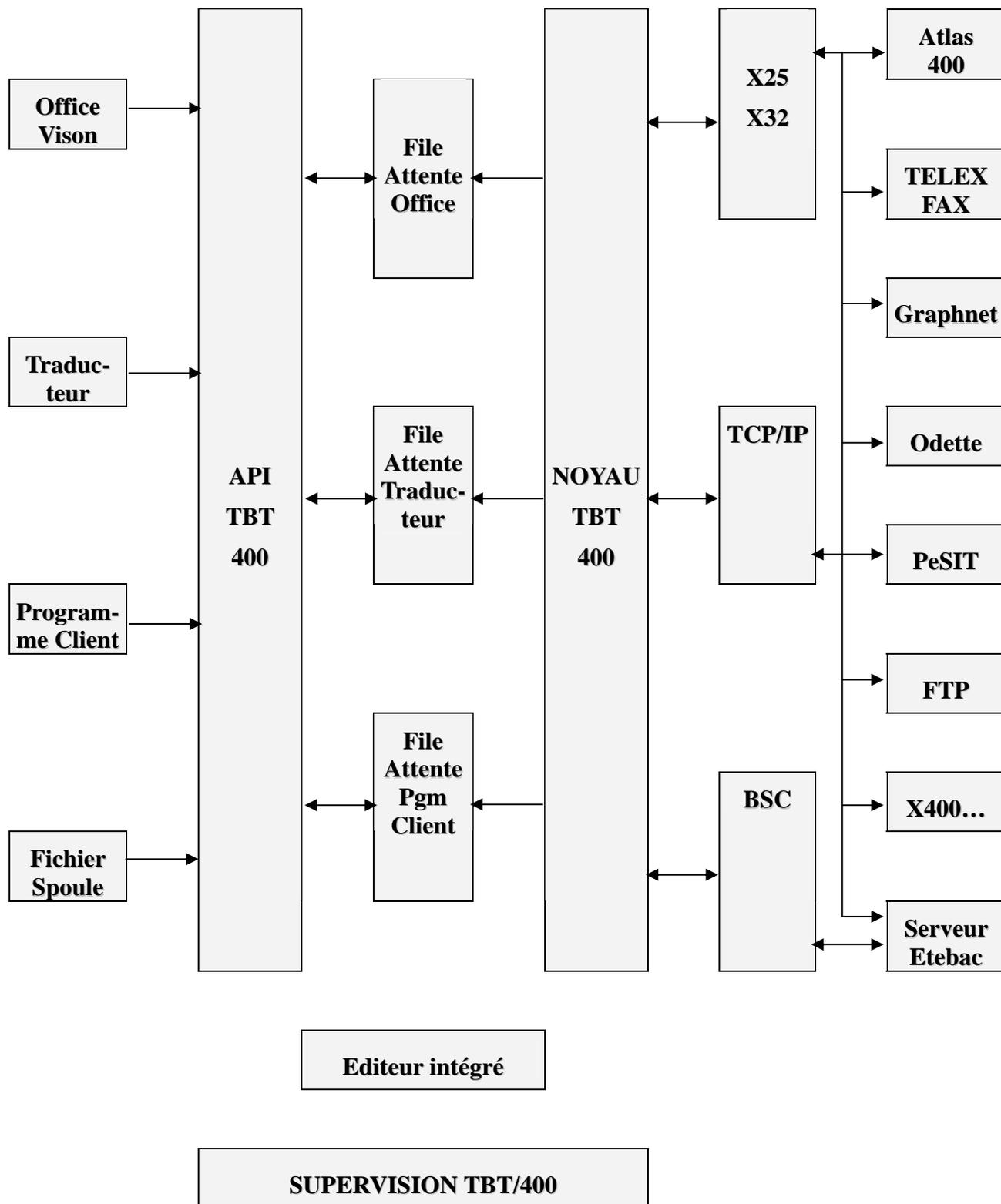
1.6. Schéma d'utilisation des API



1.7. Schéma de fonctionnement en émission



1.8. Schéma de fonctionnement en réception



1.9. Concepts

Quelque soit l'utilisation, ou les différentes options réseaux, **TBT/400** fait appel à un certain nombre de concepts. Ceux-ci permettent d'appréhender efficacement **TBT/400** aussi bien dans sa partie customisation du noyau (gestion de temporisateurs, de péremptions) que dans l'établissement des options ou des objets réseaux (Nom d'O/R X400 - boîte aux lettres **ATLAS 400**).

De plus, ces différents concepts seront usités dans la mise en place des **API**, pour l'intégration de vos applications, et dans les différents menus de **TBT/400** (Editeur, Gestion des adresses, Annuaire, ...)

1.9.1. Application

La principale notion au sein de **TBT/400** est ce qu'on nomme "**APPLICATION**". C'est une entité logique qui représente le point d'accès à la plate-forme. C'est votre identification, votre "handle", votre fenêtre d'accès à **TBT/400**. Par son intermédiaire vous vous identifiez et vous pouvez identifier une cible à votre transmission. Cette cible pourra être une chaîne applicative ou alors l'identification d'un élément externe à votre **AS/400** comme, par exemple, un réseau externe.

Toutes les **APPLICATIONS** sont traitées et gérées par **TBT/400** sur le même plan. Une différence subsiste entre les **APPLICATIONS** utilisateurs, celles que vous pouvez définir, et les **APPLICATIONS** technologiques, fournies par **TBT/400**. Les **APPLICATIONS** technologiques sont livrées avec **TBT/400**; elles peuvent être utilisées dans vos intégrations mais ne peuvent être gérées par vous-même, à la différence des **APPLICATIONS** utilisateurs.

A toute communication, quelque soit le réseau ou le protocole, un schéma unique se dégage: il faut un processus Source et un processus Cible. L'identification de ces processus s'effectue par l'intermédiaire d'un nom d'**APPLICATION**. Donc, pour communiquer via **TBT/400**, il faudra une **APPLICATION** émettrice et une **APPLICATION** destinatrice.

Attention : l'utilisateur est libre du choix du nom d'**APPLICATION**; par contre l'identification du monde extérieur subit les règles de nomenclature de **TBT/400**. En d'autres termes, l'identification d'un destinataire accessible pour une connexion **X25** se fera obligatoirement par le nom d'**APPLICATION** destinatrice **\$EXTERNA** (**APPLICATION** technologique).

1.9.2. File d'attente

A chaque définition d'**APPLICATION**, **TBT/400** associe trois files d'attente logiques : une file d'attente de message, une file d'attente d'accusé de transmission, et une file d'attente de rejet.

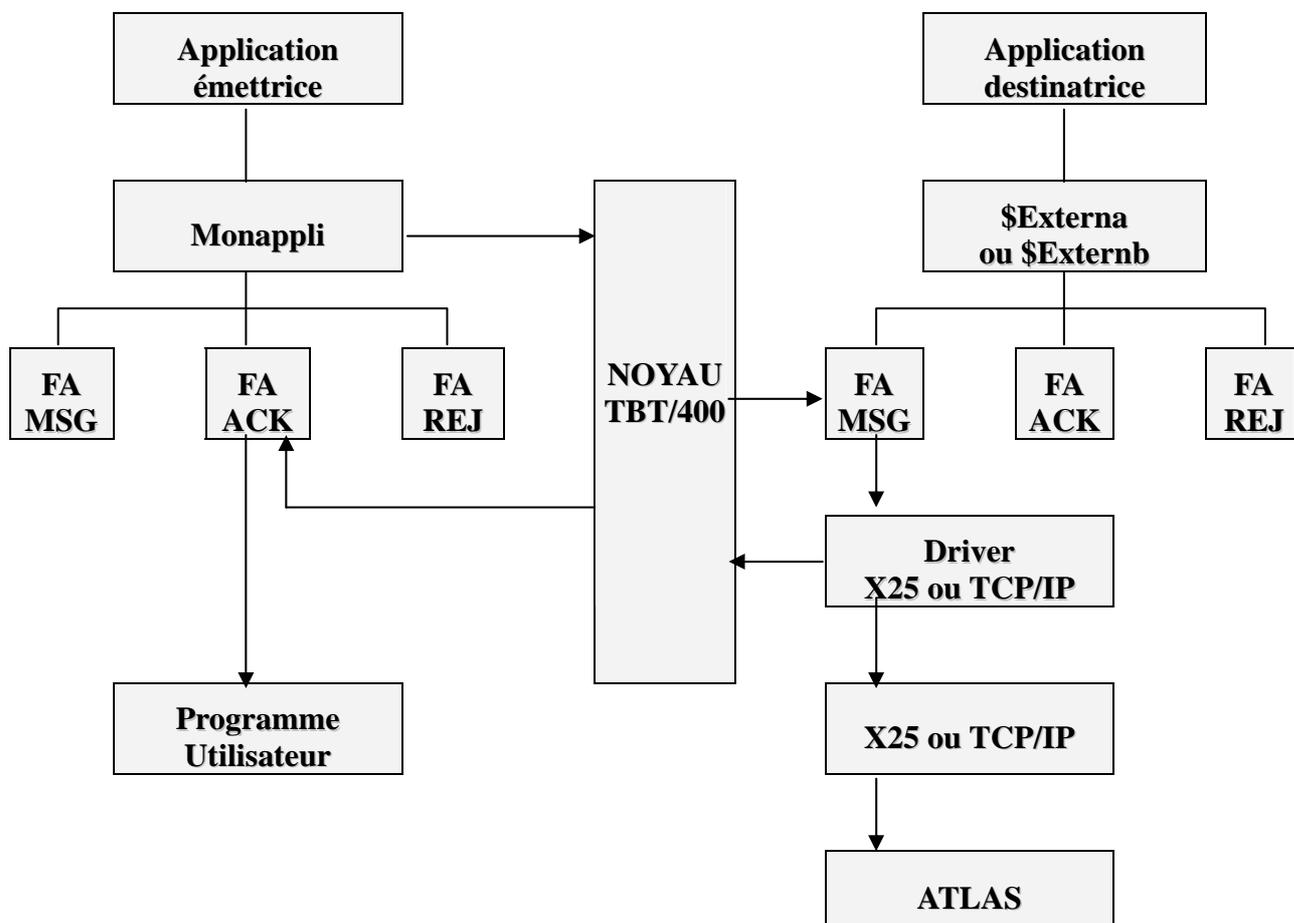
Une file d'attente est une vue logique pour une **APPLICATION** et pour un type d'événement message du spouler. C'est la possibilité pour une **APPLICATION** de canaliser et d'identifier les différents événements qui circulent:

- **File d'attente message :** Est le réceptacle de tous les messages à destination de l'**APPLICATION** susnommée et à consommer par la dite **APPLICATION**.
- **File d'attente accusé :** Permet de stocker et traiter les retours de bonne ou mauvaise transmission d'un message émis.
- **File d'attente rejet :** Stocke les messages entrants dont la couche protocole ou réseau n'a pu aboutir à une transmission correcte (problèmes d'incompréhension protocolaire), ou qui ont été refusés pour des raisons d'autorisation.

A chaque file d'attente une description est demandée, dont les principaux éléments sont :

- L'association d'un **programme de traitement** et de consommation de la file d'attente.
- **Un mode d'exploitation** permettant de définir si le processus rattaché à la file d'attente est de type temps réel et donc démarré par **TBT/400** au coup par coup (traitement en flux tendu par exemple), ou en mode batch (avec ou sans contrôle de **TBT/400**).
- Une description de bibliothèque de stockage des données pour les fichiers/messages provenant du réseau et géré par **TBT/400**.

1.9.3. Exemple

**Vous voulez transmettre un télex via le Réseau à Valeur Ajoutée ATLAS 400:**

- 1) **Appel des API de TBT/400** : fournir un nom d'APPLICATION émettrice (MONAPPLI), un nom d'application destinatrice (\$EXTERNA car réseau externe), le type de réseau, le nom du fichier et au moins le numéro Télex du destinataire.
- 2) **Le noyau prend en compte la demande**, vérifie la cohérence de l'enveloppe du message (adresse du destinataire, existence du fichier, etc...) et acquitte la prise en compte au programme émetteur. Il dépose un événement dans la file d'attente des messages de l'APPLICATION destinatrice \$EXTERNA.
- 3) Suivant le mode d'exploitation du programme rattaché à la file d'attente (dans notre exemple, il s'agit du pilote X25 de TBT/400, donc d'un démarrage à l'initialisation de TBT/400), **consommation de l'événement** par le programme rattaché.
- 4) **Transmission** sur le réseau ATLAS 400 du fichier/message.
- 5) Si l'applicatif l'a souhaité, le noyau dirige l'**acquiescement** vers la file d'attente de traitement des accusés de l'application émettrice.
- 6) TBT/400 démarre le **programme utilisateur** attaché à la file d'attente des accusés de ladite application suivant le mode d'exploitation renseigné (mode temps réel, mode batch, mode vacation ...)

2. Fonctionnalités

2.1. Fonctionnalités fichiers

2.1.1. Types de fichiers supportés

TBT/400 utilise en émission comme en réception plusieurs types de fichiers OS/400 sur l'ensemble des réseaux disponibles :

- Les fichiers physiques de toute longueur d'enregistrement,
- Les fichiers source de toute longueur d'enregistrement (sourcefile),
- Les fichiers de sauvegarde (savefile).
- Les fichiers « IFS »

De plus les fichiers logiques ou joints sont supportés (en émission seulement).

Enfin, en émission seulement, un accès aux fichiers spoule (spoolfiles) est proposé. Ceux-ci peuvent donc être envoyés (sous forme logique : ils perdent leurs attributs OS/400) aux différents réseaux, et en particulier aux destinataires finaux de type télex ou télécopie). Ils peuvent également être envoyés en format PCL, GIF ou TIFF ; un format 'propriétaire' est également prévu permettant à un TBT/400 distant de recréer un fichier spoule.

2.1.2. Modes d'Accès aux fichiers

Transcodification EbcDic/Ascii (par correspondant, par transfert...)

- Il est possible d'annuler l'effet d'une transcodification intempestive :
 - Un fichier PC (code page 1252) est envoyé par ce dernier traduit en EBCDIC (selon la table dite standard) , le code page d'arrivée est alors totalement inventé. On pourra revenir au code page original pour le transcoder correctement selon le code page d'arrivée.

Gestion des **codes page** (en émission comme en réception)

Gestion du '**Multibytes**' (en particulier UTF-8, **UTF-16**)

De multiples '**manipulations**' des fichiers émis ou reçus telles que :

- En réception :
 - Transcodification selon un code page spécifié,
 - Ecriture 'tel que reçu', par accumulation, sur rupture CR/LF LF/CR CR LF;
 - la longueur d'enregistrement annoncée par le protocole peut être honorée ou ignorée.
- En émission :
 - Groupage d'enregistrements,
 - Génération de CR/LF, LF/CR, CR ou LF,
 - Transcodification selon un code page spécifié. La traduction se fait enregistrement par enregistrement, ou, si le fichier à une description, champ par champ

Ce paramétrage est propre à une **application**, ou **particularisé par correspondant**.

2.1.3. Encodage CMS

Pour tous les protocoles réseau supportés peut :

1. Déterminer le Hash du fichier (SHA1, SHA256, SHA384, SHA512)
2. Crypter le fichier (DES, TDES_064, TDES_128, TDES_192, AES_128, AES_192, AES_256, RC2_040, RC2_064, RC2_128)
3. Signer le fichier (RSA)

4. Compresser le fichier (zlib)

Le fichier résultant est alors transmis en format CMS aisément traitable par un site distant. Un **TBT/400** distant pourra d'ailleurs reconstituer le fichier en réalisant les opérations inverses (décryptage, décompression, calcul du hash, vérification de la signature).

Ces opérations se font au fil de l'eau pendant la transmission du fichier.

Une commande est cependant à disposition pour réaliser l'encodage CMS (et le décodage) hors transmission.

TBT/400 a son propre gestionnaire de certificats ;

Les clés privées des certificats utilisés peuvent être sauvegardées dans le gestionnaire TBT/400, dans le gestionnaire IBM, ou dans les Keystores (structure OS/400 quasi inviolable).

2.2. Fonctionnalités réseaux/protocoles

Les options de **TBT/400** disponibles à ce jour sont présentées ci-dessous. N'hésitez pas à nous consulter pour tout besoin de communication, de nouveaux modules étant régulièrement ajoutés à **TBT/400**.

2.2.1. ATLAS 400 - ALLEGRO

ATLAS 400 est le nom d'un Réseau à Valeur Ajoutée à la norme **X400** commercialisé par **TRANSPAC**. Son accès permet l'émission et la réception de fichiers ou messages vers des abonnés individuels **ATLAS 400** mais aussi vers d'autres serveurs privés **X400**, ainsi que vers des télécopieurs Téléx, ou un quelconque Fax. Via **ATLAS 400**, vous disposez ainsi d'une pluralité de destinataires en France et à l'international.

TBT/400 permet l'accès à ce **RVA** par une connexion **X25**, **X32** ou **RNIS**. Il reçoit, en remise directe ou par scrutation explicite, des fichiers ou messages en mode texte ou transparent afin de les distribuer à vos applicatifs, et assure à partir de vos applicatifs l'émission de messages ou fichiers vers des destinataires finaux qui peuvent être un abonné **X400**, un fax ou un Téléx.

TBT/400 utilise le protocole **ATLAS440** ; ce protocole, d'accès natif à **ATLAS 400**, évite les pertes d'information liées à l'usage de protocoles supportés par **ATLAS 400**, mais inadaptés (de type Odette FTP).

TBT/400 gère la richesse des comptes-rendus de transmissions fournis par **ATLAS 400**. Il peut transmettre à des applicatifs des accusés de prise en compte, et des accusés de bonne ou mauvaise distribution comme défini dans la normalisation **X400**.

TBT/400 peut gérer différents flux sur une même boîte aux lettres **ATLAS 400** et peut commuter les flux entrants vers vos applicatifs grâce à l'analyse de l'enveloppe du message. Ceci permet par exemple, via une même boîte aux lettres, d'associer les flux **EDI** de type commande vers la chaîne de traitement des commandes et les flux de type facture vers la chaîne applicative comptable. De plus, le type de fichier créé peut être défini dans l'enveloppe (longueur d'enregistrement, fichier source, alphabet utilisé, etc ...).

Si vous le souhaitez, la gestion d'un nombre illimité de boîtes aux lettres **ATLAS** sur la même connexion est aussi possible; par exemple une boîte aux lettres **ATLAS** peut être égale à une adresse réseau et correspondre à une seule entreprise de votre groupe ou de votre holding. Ceci permet également, pour les forts trafics, d'apporter une solution à la limite actuelle de 255 messages dans une boîte **ATLAS**.

De plus, grâce à la passerelle **ATLAS-ALLEGRO**, vous avez désormais accès au **RVA ALLEGRO**, en émission comme en réception. La remise ouverte d'**ATLAS 400** vous permet d'utiliser **ALLEGRO** en flux tendu, en économisant les appels de scrutation de boîte et les communications des messages entrants (ce qui n'est pas possible avec ALLEGRO en direct).

TBT/400 gère des télécopies jusqu'à 198 colonnes en format horizontal et 132 colonnes en format vertical.

TBT/400 gère aussi bien le mode transparent que le mode non transparent d'**ATLAS**. L'écriture peut se faire en 'découpant' le flux reçu selon une longueur d'enregistrement définie, ou en recherchant des **CR/LF**.

Pour plus d'informations, voir le site Web <http://www.atlas400.org/>

2.2.2. X400 : Allegro Calvacom Carrefour Diva

X400 est un ensemble de recommandations de transfert de messages habituellement utilisé en **EDI**.

TBT/400 implémente un "MTA", réalisant ainsi les liaisons de type **P1** et **P2**.

Ceci permet la connexion directe avec des serveurs de messagerie de type **Allegro Calvacom, Carrefour, Diva**

TBT/400 gère autant d'UA locales (boîtes aux lettres) que nécessaire.

Le remplissage des buffers est optimisé pour rendre optimales la vitesse et le coût des transferts.

Les trames de contrôle du protocole sont décodées et archivées (Logs et MSGQ) pour faciliter les analyses d'incident.

Une trace complète des échanges peut être globalement ou sélectivement (par correspondant) mise en œuvre.

Pour plus d'informations, voir le site Web <http://www.x400.org/>

2.2.3. INTERNET

TBT/400 dispose d'un **FTP** client serveur sécurisé et automatisé.

Via l'opérateur **Graphnet** , il vous est possible de transférer un fichier **AS/400** à l'adresse E-MAIL d'un correspondant.

Via le réseau à valeur ajoutée **ATLAS 400** à la condition exclusive d'avoir souscrit l'option internet en complément de votre abonnement **ATLAS 400**, vous pouvez transférer un fichier **AS/400** à l'adresse E-MAIL d'un correspondant

TBT/400 dispose également d'un client **SMTP** intégré.

2.2.4. GRAPHNET

GRAPHNET est un Réseau à Valeur Ajoutée spécialisé dans l'émission de Télex et de Télécopies. Une grande souplesse de mise en page (préimprimés, caractères accentués, etc ...) est supportée par cet opérateur.

TBT/400 permet l'accès à ce **RVA** par une connexion **X25, X32, RNIS, TCP/IP**. Il assure à partir l'émission de messages ou fichiers vers des destinataires finaux qui peuvent être un fax ou un Télex.

TBT/400 gère la richesse des comptes-rendus de transmissions fournis par **GRAPHNET**. Il peut transmettre à des applicatifs des accusés de prise en compte, et des accusés de bonne ou mauvaise distribution.

TBT/400 permet la gestion d'un nombre illimité de boîtes aux lettres **GRAPHNET** sur la même connexion est aussi possible; ceci permet entre autre une gestion multisociétés.

TBT/400 gère des télécopies jusqu'à 198 colonnes en format horizontal et 132 colonnes en format vertical.

TBT/400 donne accès à l'ensemble des services de cet opérateur.

2.2.5. ODETTE - FTP - GEIS - IBM GN - GALIA

Le protocole **ODETTE** de transfert de fichier est adopté par de nombreux pays regroupés dans l'organisation **ODETTE**, dont **GALIA** est un correspondant en France. Il est notamment utilisé dans le monde de l'automobile, et est un accès au Réseau à Valeur Ajoutée **GEIS**, ainsi qu'à **IBM GN** (ex IN).

TBT/400 assure la communication par une connexion **X25, X32, RNIS, TCP/IP**. Il reçoit, en remise directe ou par scrutation explicite, des fichiers ou messages afin de les distribuer à vos applicatifs, et assure à partir de vos applicatifs l'émission de messages ou fichiers vers des destinataires finaux.

TBT/400 échange des fichiers directement avec un correspondant final, ou à travers un **RVA** (par exemple **GEIS**) disposant d'un accès en protocole **ODETTE**. Il gère tous les messages prévus par ce protocole, et peut transmettre à vos applicatifs l'ensemble des informations du réseau.

TBT/400 peut être aussi bien appelant qu'appelé par vos correspondants.

TBT/400 peut mettre à disposition des fichiers que vos correspondants viendront ultérieurement rechercher (fonctionnalité serveur).

TBT/400 a subi avec succès tous les tests imposés par **GALIA**, et est donc homologué **ODETTE - GALIA**.

TBT/400 supporte tous les types de fichier **ODETTE**, la compression, le restart, et même la "Special logic".....

Les tailles de buffer et taille de fenêtre sont négociées avec le partenaire distant, pour faciliter le paramétrage.

Le remplissage des buffers est optimisé pour rendre optimales la vitesse et le coût des transferts.

Les trames de contrôle du protocole sont décodées et archivées (Logs et MSGQ) pour faciliter les analyses d'incident.

Une trace complète des échanges peut être globalement ou sélectivement (par correspondant) mise en œuvre.

TBT/400 supporte **OFTP V2**

Pour plus d'informations, voir le site Web <http://www.oftp.net/>

2.2.6. PeSIT

Le protocole **PeSIT** a été initialement conçu pour le raccordement des Centres de Traitements Bancaires. Ce protocole, intégrant des mécanismes de compression et de reprise très sophistiqués, a été implémenté dans la plupart des sites ES/9000 et est devenu une référence dans le monde de la communication **X25**.

TBT/400 assure la communication par une connexion **X25, X32, RNIS, TCP/IP**. Il reçoit, en remise directe ou par scrutation explicite selon un échéancier, des fichiers ou messages afin de les distribuer à vos applicatifs, et assure à partir de vos applicatifs l'émission de messages ou fichiers vers des destinataires finaux.

TBT/400 échange des fichiers directement avec un correspondant disposant d'un accès en protocole **PeSIT**. Il gère tous les messages prévus par ce protocole, et peut transmettre à vos applicatifs l'ensemble des informations du réseau.

TBT/400 peut être aussi bien appelant qu'appelé par vos correspondants.

TBT/400 peut mettre à disposition des fichiers que vos correspondants viendront ultérieurement rechercher (fonctionnalité serveur).

TBT/400 supporte les niveaux D et E; les transferts fixe ou variable, le restart, les deux types de compression.

Les tailles de buffer et taille de fenêtre sont négociées avec le partenaire distant, pour faciliter le paramétrage.

Le remplissage des buffers est optimisé pour rendre optimales la vitesse et le coût des transferts.

Les trames de contrôle du protocole sont décodées et archivées (Logs et MSGQ) pour faciliter les analyses d'incident.

Une trace complète des échanges peut être globalement ou sélectivement (par correspondant) mise en œuvre.

Pour plus d'informations, voir le site Web <http://www.pesit.com/>

2.2.7. FTP

TBT/400 dispose d'un **FTP** client serveur sécurisé et automatisé.

Ceci permet de « voir » les correspondants FTP comme tous les autres correspondants **TBT/400** :

Exemple : collecte d'information sur l'AS/400 émise par des PC sur Intranet.

En mode client, un suivi intégral des transferts est ainsi immédiatement disponible.

En mode serveur, la sécurité mise en œuvre n'utilise pas la sécurité objet de l'**OS/400**. Les signatures utilisées sont purement internes **TBT/400**; En réception de fichier, le nom de fichier fourni par le client est un nom logique, **TBT/400** créant des noms physiques dynamiques évitant tout "écrasement". De plus, le fichier est passé automatiquement à l'application de traitement selon un protocole standard **TBT/400**. Ceci dispense d'une tâche scannant régulièrement une bibliothèque d'arrivée à la recherche de nouveautés, traitant au passage des fichiers partiels (suite à une rupture de communication par exemple).

Les trames de contrôle du protocole sont décodées et archivées (Logs et MSGQ) pour faciliter les analyses d'incident.

Une trace complète des échanges peut être globalement ou sélectivement (par correspondant) mise en œuvre.

2.2.8. Téléx ou Télécopie

Plusieurs interfaces réseau de **TBT/400** permettent l'envoi de télex et/ou télécopie/fax à partir de vos applicatifs, ou directement par vos utilisateurs via l'éditeur intégré ou votre messagerie, sans posséder sur votre site informatique la moindre infrastructure matérielle: pas de boîtier, de "boîte noire" ou de frontal dit intelligent, et surtout pas de lignes Téléx ou Fax.

TBT/400 utilise alors une connexion **X25** ou **X32** ou **RNIS** pour transmettre vos messages en direct ou via un RVA selon les besoins et le volume. Par ce type de connexion et via un protocole géré en natif par **TBT/400**, c'est la possibilité de disposer au moment voulu d'un nombre illimité de lignes de sorties Téléx ou Télécopie sans en subir le reste du temps la gestion et le coût.

C'est une infrastructure qui allie la disponibilité du réseau commuté à la souplesse du progiciel, sans subir les contraintes liées aux divers composants matériels. Elle apporte des économies au niveau matériel, mais également par une baisse importante des coûts de fonctionnement, à l'international et dans certains cas en France même, par rapport aux solutions directes. Nous consulter pour toute précision sur ces services et pour un devis de fonctionnement gratuit.

TBT/400 gère des télécopies jusqu'à 198 colonnes en format horizontal et 132 colonnes en format vertical.

2.2.9. Remote ETEBAC 3

Dans les échanges entre les banques et leurs clients, les télétransmissions sont de plus en plus utilisées; des normes interbancaires définissent le protocole, et notamment **ETEBAC 3** pour les transferts sur réseau Transpac en procédure **X25**, **X32**, **RNIS**, **TCP/IP**.

L'interface Remote ETEBAC 3 de **TBT/400** prend en charge les transferts de fichiers selon ce protocole, dans le sens Client vers Banque comme dans le sens Banque vers Client. Il assure l'échange de la carte paramètre, son contrôle, et le transfert proprement dit. Suivant les règles du protocole, **TBT/400** est dans ce cas toujours à l'initiative de l'appel.

Pour plus d'informations, voir le site Web <http://www.etebac.org/>

2.2.10. Serveur de fichiers

La fonction serveur de **TBT/400** se présente sous plusieurs options :

- serveur de type ETEBAC 1-2,
- serveur de type ETEBAC 3,
- serveur de type ODETTE.
- serveur de type PeSIT.
- serveur de type FTP

Ces options permettent les échanges de fichiers avec un ensemble de correspondants distants, sur matériel hétérogène (PC, IBM, DEC, ...). C'est par exemple le cas des serveurs bancaires pour les échanges avec leurs clients, mais également le cas des sociétés géographiquement dispersées, des groupes, ...

2.2.11. Protocole interne - Télémaintenance

TBT/400 dispose d'un protocole interne adapté à la structure des fichiers de l'**AS/400**.

Ce protocole, avec l'accord du client, est au moins ouvert en réception sur le site. Ceci permet une télétransmission d'objets **OS/400**, et donc une télémaintenance (livraison de PTF, nouvelles versions, etc ...). De fait, il n'est nul besoin de la complexité de mise en œuvre d'un réseau de type SNADS (orienté réseau interne au demeurant).

2.3. Fonctionnalités annuaire

2.3.1. Annuaire multi protocoles

TBT/400 dispose d'un annuaire multi protocoles, à mise à jour interactive. Toute mise à jour est effective immédiatement. Tous les réseaux connus de **TBT/400** sont intégrés dans cet annuaire.

TBT/400 centralise par l'annuaire toutes les adresses réseau et les profils de transmission utilisés (EBCDIC ou ASCII par exemple).

En émission, le programme d'application utilisant **TBT/400** peut ne connaître ni le réseau ni l'adresse sur ce réseau du correspondant. **TBT/400** assure ainsi l'indépendance avec le réseau utilisé.

En réception, le même annuaire sert à identifier l'origine des fichiers reçus et éventuellement à particulariser les traitements.

2.3.2. Contrôle d'adresse X25 et TCP/IP

TBT/400 peut, de manière optionnelle, contrôler l'adresse **X25** ou **TCP/IP** d'un appelant. Ceci permet de renforcer de manière drastique la sécurité d'accès

En **TCP/IP**, le contrôle se fait sur le couple adresse, masque de sous réseaux. Un contrôle est également possible sur un nom de host, ce par l'utilisation d'une résolution inverse d'adresse.

Cette fonctionnalité permet également de discriminer des appelants s'identifiant sous le même nom.

2.3.3. Contrôle de sous adresse X25

TBT/400 peut, de manière optionnelle, contrôler la sous adresse **X25** d'un appelant.

Cette fonctionnalité permet de discriminer des appelants s'identifiant sous le même nom.

2.3.4. Contrôle d'accès aux applications

TBT/400 peut, de manière optionnelle, restreindre l'accès aux applications de traitement des flux reçus aux seuls correspondants formellement autorisés (et donc identifiés). Ceci permet de sécuriser les applications qui n'ont pas les moyens d'effectuer un contrôle elles mêmes (un traducteur EDIFACT, par exemple, ne peut connaître l'origine réelle d'un fichier, et en conséquence ne peut sécuriser de manière satisfaisante).

2.3.5. Création automatique d'entrées dans l'annuaire

TBT/400 doit connaître l'adresse réseau des correspondants récepteurs de fichiers. En revanche, pour les correspondants émetteurs, cette connaissance de l'adresse est optionnelle.

Ceci est particulièrement vrai pour les protocoles **ATLAS**, **X400** et **Odette** utilisant des 'indirections'.

Dans chacun de ces cas, le correspondant au sens réseau doit être connu (boîte de réception pour Atlas, MTA distant pour **X400** et Correspondant réseau pour Odette), mais le correspondant à l'origine du message n'a pas à l'être.

Lorsque **TBT/400** reçoit un message transmis par un correspondant réseau connu, mais dont l'émetteur réel n'est pas déclaré dans l'annuaire, une entrée 'dynamique' sera créée dans l'annuaire **TBT/400**. Cette entrée facilite le suivi des transmissions.

Les entrées dynamiques peuvent être aisément repérées dans l'annuaire.

Une commande permet de 'balayer' l'historique des transmissions, pour réaliser le rapprochement avec l'annuaire. Ceci permet en particulier, après avoir rebaptisé dans l'annuaire les entrées dynamiques, en leur donnant des noms mnémoniques, de remettre l'historique en phase avec celui-ci.

De plus en **X400**, **TBT/400** supportant plusieurs UA locales, celles-ci peuvent également être créées dynamiquement. Il en est de même pour les correspondants 'locaux' Odette.

Ces entrées 'locales' sont également la cible du 'scan de l'historique'.

2.4. Fonctions de Supervision

Plusieurs services de supervision et de suivi des échanges sont fournis par **TBT/400**:

2.4.1. Menus de supervision

Les besoins essentiels d'exploitation d'une plate-forme de communication peuvent se résumer dans les éléments suivants:

- Suivre l'activité en cours
- Connaître l'activité à venir
- Vérifier les transferts effectués
- Rechercher les erreurs et incidents
- Les expliquer et y remédier

Toute la supervision de **TBT/400** s'inscrit dans cette optique.

- Au sein de sa couche de supervision, **TBT/400** permet de suivre à tout moment l'état des communications en cours, à venir, ou passé.
- C'est une visualisation en direct de tous les éléments de communication déclarés dans **TBT/400**.
- Ces diverses fonctionnalités permettent des vues synthétiques des différentes files d'attente logiques, mais aussi les détails exhaustifs de tous les éléments transmis ou à transmettre, ainsi que la partie donnée (Visualisation du fichier).
- Les enveloppes réseau, avis de prise en compte, avis de distribution, sont intégralement archivés et rapprochés des messages émis. Ces divers éléments sont tous consultables dans les menus de supervision.

- De plus, afin d'optimiser le travail de l'exploitant, des recherches multicritères sur des zones sensibles décrivant les transferts sont possibles pour les messages en cours ou historisés.
- **TBT/400** donne l'accès immédiat à la liste des transferts en erreur. Ceux-ci sont signalés en rouge (ou en double brillance pour les écrans monochromes) dans les menus de supervision.
- **TBT/400** permet à l'applicatif de traitement d'un fichier reçu de lui affecter un code retour. Ceci autorise, avec un peu de complicité de l'applicatif, une supervision du traitement des fichiers reçus.
- Une touche de fonction donne accès aux spools du job ayant traité un fichier reçu. Il est donc quasi immédiat de retrouver la '**JOBLOG**' d'un job de traitement.
- Un message peut être réémis par écran: ceci permet par exemple la correction du nom erroné d'une application pour un message entrant, pour recyclage interne et traitement automatique par l'applicatif.
- Les échanges réseau essentiels (trames de signature par exemple) sont archivés en clair y compris la traduction des codes diagnostics connus pour accélérer les recherches d'incidents.
- Il est même possible de visualiser, voire modifier si on y est autorisé, un fichier de **TBT/400** émis ou reçu (jusqu'à une vue hexadécimale éventuellement), simplement par une touche de fonction.
- Impression éventuelle des fichiers reçus.

2.4.2. Messages Queues

Les erreurs graves liées au réseau (par exemple ligne **X25** ou **BSC** hors service) sont signalées dans la Message Queue Opérateur. Les erreurs fonctionnelles (par exemple refus de carte paramètre) le sont dans une Message Queue dédiée accessible directement depuis les menus **TBT/400**.

De manière optionnelle, des messages peuvent être envoyés dans la message queue '**QSYSOPR**', ou la message queue de l'émetteur (au sens **OS/400**) selon le bon ou mauvais traitement d'un fichier émis ou reçu.

2.4.3. Output Queues

Des options de trace des échanges peuvent être activées permettant, dans les cas complexes, ou pour la mise au point finale applicative, d'avoir des éléments de résolution.

2.4.4. Vue OS/400

Toutes les commandes **OS/400** utiles à la surveillance du fonctionnement de **TBT/400** sont habillées, de manière à permettre une assistance aisée, l'utilisateur n'ayant pas à connaître la syntaxe 'ésotérique' des commandes natives **OS/400**.

2.4.5. Remontées d'alertes

Tous les incidents réputés graves (Incidents en gestion de ligne, Appel sortant refusé, Appel entrant refusé, Incident dans applicatif de traitement de fichier reçu,...) Rencontrés par **TBT/400** peuvent appeler un exit de traitement d'alertes. Celui-ci dispose de tous les éléments contextuels, pour signaler l'incident au système de traitement d'alertes en place sur le site.

2.5. Interfaces applicatives

2.5.1. Emission

TBT/400 dispose d'un ensemble de commandes **OS/400** permettant d'envoyer un fichier sur un réseau donné.

Certaines assurent l'indépendance par rapport au réseau : par exemple, **IPSNEDI OBJFIL**(COMMANDE) **NOMLOG**(MONFOURNISSEUR) demande à **TBT/400** d'envoyer le fichier COMMANDE au correspondant MONFOURNISSEUR recensé dans l'annuaire.

Ces commandes constituent un API de 1^{er} niveau qui sera le plus souvent utilisé. Elles appellent un API de 2^{ème} niveau aux fonctionnalités plus riches et lui même accessible aux programmes **RPG**, **COBOL**, et **C**.

Lors de la demande de transmission, une possibilité de duplication du fichier à émettre existe ; **TBT/400** en prend alors une copie, et travaille à partir de celle-ci. Ceci permet :

- de désolidariser l'applicatif d'émission de **TBT/400**; sitôt la demande d'émission acceptée, l'applicatif peut à nouveau travailler sur le fichier initial (sinon, la transmission étant asynchrone, l'applicatif doit s'interdire d'utiliser le fichier jusqu'à la fin du traitement de celui-ci par **TBT/400**, ou assurer son unicité)
- d'assurer une historisation des fichiers émis identique à l'historisation des fichiers reçus.

2.5.2. Remontées applicatives

Pour informer directement les applicatifs, des acquittements à différents niveaux peuvent leur être transmis. Ceux-ci peuvent donc suivre totalement la vie d'un message (en plus de l'historisation systématique effectuée par **TBT/400**).

2.5.3. Réception

TBT/400, après réception d'un fichier, peut gérer un processus applicatif de plusieurs manières :

- démarrage immédiat du processus : couplé à la remise directe, ceci autorise le traitement en flux tendu.
- accumulation et démarrage par une commande **TBT/400** du processus.
- accumulation et démarrage hors contrôle de **TBT/400** du processus, les traitements étant supervisés par **TBT/400**.
- accumulation simple, **TBT/400** ignorant les traitements applicatifs ultérieurs.

Le choix s'effectue par un Paramétrage externe; dans les trois premiers cas la programmation des applicatifs est identique : il est donc permis de changer de mode par Paramétrage direct.

Un ensemble de commandes (de programmation uniquement) permet d'accéder aux informations disponibles dans **TBT/400**. Ces commandes constituent un API de 1^{er} niveau qui sera le plus souvent utilisé. Comme pour l'émission, elles appellent un API de 2^{ème} niveau aux fonctionnalités plus riches et lui même accessible aux programmes **RPG**, **COBOL**, et **C**.

IPSRCVTBT OBJLIB(&LIB) OBJFIL(&FIL) OBJMBR(&MBR) NOMLOG(&NOM) valorise les variables &LIB, &FIL, &MBR avec les caractéristiques du fichier reçu, et &NOM avec le nom du correspondant émetteur du fichier. De manière optionnelle, la quasi totalité des champs de l'enveloppe réseau sont ainsi accessibles.

Un modèle de programme de réception (en **CLP**) est disponible. Ce modèle est opérationnel, seule la partie 'traitement utilisateur' restant à compléter. Si un programme applicatif disposant d'un fichier en entrée est disponible, il est donc possible de l'interfacer avec **TBT/400** en quelques minutes.

2.6. Aides à la programmation

2.6.1. Générateur de commandes : Scan d'objets

TBT/400 met à disposition une commande permettant de rechercher dans un ensemble de bibliothèques un ensemble d'objets, en en précisant le type, le nom (générique), l'attribut (générique), de sélectionner éventuellement un ensemble de membres (pour les objets de type **FILE*), et pour tout élément sélectionné de générer une commande **OS/400**, ce après substitution de valeurs (nom d'objet, de bibliothèque). Ceci permet, par exemple, d'envoyer à **TBT/400** le contenu de toute une bibliothèque en une seule commande....

2.6.2. Générateur de commandes : Scan de spools

TBT/400 met à disposition une commande permettant de rechercher dans une Output queue un ensemble de spool files, en en précisant le nom, le statut, le code utilisateur, la référence utilisateur, et pour tout élément sélectionné de générer une commande **OS/400**, ce après substitution de valeurs (nom de spool, nom de Job, numéro de spool,). Ceci permet, par exemple, d'envoyer à **TBT/400** le contenu de toute une Output queue en une seule commande....

2.7. Fonctionnalités diverses

2.7.1. Echéancier

TBT/400 dispose d'un échéancier intégré permettant :

- d'envoyer des fichiers,
- d'effectuer des scrutations (vidages de boîte aux lettres),
- éventuellement de soumettre des jobs.

Cet échéancier dispose de la notion de répétition, de fréquence, de plage, de jours d'activité, et de traitement des jours fériés ; il est de fait beaucoup plus efficace (dans son domaine d'utilisation) que le 'job scheduler' de l'**OS/400**. Il peut, par exemple, en un seul ordre traiter le cas suivant : aller vider ma boîte **ATLAS** toutes les heures entre sept heures et dix huit heures les jours ouvrés. Il n'a pas la prétention de se substituer à un robot d'exploitation, mais à pallier à son absence.

Il est disponible pour tous les protocoles supportés par **TBT/400**.

2.7.2. Historisation

TBT/400 archive tous les fichiers reçus, ainsi que, sur option, tous les fichiers émis.

Les événements (émissions ou réceptions) sont également historisés. En supervision, il est possible d'en visualiser la liste, et d'accéder directement aux fichiers objets par touche de fonction.

2.7.3. Purge automatique

TBT/400 dispose d'un automate de purge permettant :

- de 'nettoyer' le fichier historique,
- de supprimer les fichiers archivés,
- de faire le ménage dans les divers composants **OS/400** (messages queues, output queues).

Cette purge, entièrement paramétrée, se fait au fil de l'eau, par un job en background de priorité basse pour ne pas pénaliser le système. Il n'y a donc pas besoin d'arrêter le sous-système, ce qui en augmente la disponibilité

Cette purge se fait de manière sélective :

- par correspondant
- par type d'évènement (Echéancier, avis de distribution X400, scrutations)

2.7.4. Gestion dynamique des menus

Les menus sont gérés dynamiquement selon l'utilisateur et les différentes options utilisées par votre société, et donc certains choix peuvent être occultés.

Des menus, en cohérence avec les différentes fonctionnalités Réseaux/Protocoles souscrites par votre société et disponibles sur votre plate-forme **TBT/400**, permettent la gestion de vos flux et de vos correspondants via l'annuaire ou sans le support de l'annuaire.

Ces menus présentent des écrans pré formatés aux spécificités du réseau de votre choix afin de fournir une transmission aisée de vos fichiers ou messages.

2.7.5. Aides en ligne

Bien entendu, quelque soit la fonctionnalité que vous utilisez, une aide en ligne contextuelle et conceptuelle vous est fournie afin de répondre à vos différentes questions.

Ces aides sont disponibles dans les menus, ou dans les commandes. Des liens de type hypertexte sont également définis, permettant une recherche ou un apprentissage aisés.

2.7.6. Editeur intégré

TBT/400 dispose d'un éditeur intégré, de type PDM, permettant de saisir ou modifier des messages, avec des notions de messages préenregistrés.

2.7.7. Automate d'installation

TBT/400 dispose d'une procédure en permettant l'installation en un minimum de temps (moins d'une heure).

2.7.8. Fonctions utilisateur

TBT/400 dispose d'un sous-ensemble de fonctions directement accessible à l'utilisateur 'final' :

- annuaire 'local',
- saisie et émission de messages,
- supervision 'locale'.

2.7.9. Bannières

Dans les divers protocoles utilisés, certains éléments nécessaires au traitement d'un fichier sont absents (par exemple longueur d'enregistrement pour un transfert **ATLAS**, alphabet pour un transfert Odette, etc...). En standard, **TBT/400** recherche ces divers éléments dans son Paramétrage. Une possibilité existe cependant, dans tous les protocoles, de définir une bannière contenue dans un champ de l'enveloppe et définissant ces éléments inconnus.

Ceci permet, si l'émetteur collabore un peu, d'introduire des notions de longueur d'enregistrement quand le réseau ne connaît pas cette notion (cas d'**ATLAS** par exemple), la notion d'alphabet (cas d'**Odette**), la notion de mode d'écriture du fichier reçu (recherche de **CR/LF**), la notion de type de fichier (fichier 'source', fichier 'physique'....), ou la notion de code page.

TBT/400 peut également générer cette bannière automatiquement (ce qui suppose l'existence d'un **TBT/400** sur le site cible).

Exemples d'application :

- Recevoir des fichiers de longueur d'enregistrement quelconque, longueur non annoncée par le protocole, sans avoir à recenser tous les cas à l'avance.
- Recevoir des 'Savefiles' au travers d'un réseau de type flux (**ATLAS** par exemple).

2.8. Passerelles avec traducteur ou messagerie

TBT/400 fournit, dans le but d'être le plus possible intégré à votre système d'information, un ensemble de passerelles avec des progiciels phares dans le monde **AS/400** qui peuvent avoir des besoins de communication.

C'est notamment le cas des traducteurs pour l'**EDI**, ou de votre messagerie d'entreprise.

Ces passerelles ont pour objectif de vous décharger d'un éventuel interfaçage entre **TBT/400** qui gère les communications et votre progiciel **AS/400**, ceci vous laissant l'entretien maîtrisé de vos applicatifs vers le produit **AS/400** de votre choix.

Les passerelles disponibles sont :

- Passerelle **TBT/400 EDI400** Passerelle entre **TBT/400** et le traducteur **EDI400** d'**IBM**.
- Passerelle **TBT/400 Synchronlink** Passerelle entre **TBT/400** et le traducteur **Synchronlink** d'**INFLUE**.
- Passerelle **TBT/400 EDIBASE** Passerelle entre **TBT/400** et le traducteur **EDIBASE** de **CGI**.
- Passerelle **TBT/400 GENEDI** Passerelle entre **TBT/400** et le traducteur **GENEDI** d'**AGENA3000**.
- Passerelle **TBT/400 GALION** Passerelle entre **TBT/400** et le traducteur **EDIMANAGER** d'**ORACLE**.
- Passerelle **TBT/400 OFFICE:** Passerelle entre **TBT/400** et la messagerie **OFFICE/400** d'**IBM**.
- Passerelle **TBT/400 OPEN/400:** Passerelle entre **TBT/400** et le système fax de **DPII**

Les passerelles avec les traducteurs sont de type intégral : suivi du comportement du traducteur par **TBT/400**, et quand le traducteur le permet, suivi par celui-ci des transferts.

2.9. Exemple de Serveur ETEBAC 1-2 ou ETEBAC 3

Le serveur **ETEBAC** permet à toute station ou ordinateur de type client, de se connecter à **TBT/400** en protocole **ETEBAC** pour échanger des fichiers. Les principales fonctionnalités sont:

2.9.1. Sécurité des accès

TBT/400 utilise l'annuaire interne. Tous les correspondants **ETEBAC** doivent y être recensés sous un nom logique, nom sous lequel il devra s'identifier par la carte paramètre. Les mises à jour sont à effet immédiat. A un correspondant, plusieurs attributs sont associés, dont un mot de passe, et une liste optionnelle de numéros d'appel **X25**. Le correspondant ne pourra alors appeler qu'en **ETEBAC 3**, et exclusivement à partir d'un de ces numéros (option de sécurité maximale). De plus, les correspondants peuvent être autorisés ou non par application.

2.9.2. Identification applicative

Une fois l'appel identifié, et donc le protocole **ETEBAC** retenu, une identification applicative entre en jeu. Elle se traduit par la réception et le décodage d'une carte paramètre.

La structure de cette carte est définie à l'installation (position et longueur de chaque champ), avec:

1. un code utilisateur (20 caractères maximum),
2. un ou deux mots de passe (20 caractères maximum),
3. une application demandée (8 caractères maximum),
4. une date en format AAMMJJ ou en format N signifiant date du jour - N (facultatif),
5. un numéro d'ordre (6 caractères maximum, facultatif) ,
6. deux critères de sélection libres (8 caractères maximum chacun, facultatifs),
7. le nombre d'enregistrements à recevoir, au sens serveur (8 caractères maximum, facultatif).

2.9.3. Mise à disposition d'un fichier

Tout fichier mis à disposition pour une réception par le client (sens R de la carte paramètre) aura au préalable été défini à **TBT/400** par les **API** d'émission. Un fichier mis à disposition est un fichier émis au sens **TBT/400** vers un correspondant **ETEBAC**, pour une application, une date et un numéro d'ordre définis. L'API d'émission valide ces notions d'après les éléments renseignés (Annuaire, table des applications). Les diverses mises à jour sont synchrones, c'est à dire à effet immédiat. Lors du dépôt d'un message, le statut du sous-système **TBT/400** est indifférent: le dépôt peut se faire noyau à l'arrêt ou actif. Le dépôt ne présume pas du type de sortie **BSC** ou **X25**.

De plus, **TBT/400** offre de la souplesse au niveau des fichiers à émettre. Par exemple, lors de la demande d'émission, une option de duplication de fichier peut être utilisée, auquel cas **TBT/400** libère le fichier initial dès le dépôt de la requête, et prendra en charge l'épuration des fichiers selon divers critères optionnels:

- purge après émission sur le réseau: dans ce cas, l'utilisateur ne peut "consommer" le fichier qu'une seule fois.
- purge liée à celle de l'historique (sur critère global de délai): le fichier bascule dans l'historique après émission.
- purge non liée à l'historique (sur critère global de délai): dans le cadre de l'épuration automatique des composants.

2.9.4. Traitement d'un appel entrant

Lorsque **TBT/400** reçoit un appel entrant de type **ETEBAC**, il lit et décode la carte paramètre, et après certains contrôles, l'accepte ou la refuse. En **ETEBAC 3** le refus se manifeste par une réponse **NOKxxxx** avec attente de la carte paramètre suivante, en **ETEBAC 1-2** par une coupure violente de la communication.

Les contrôles sont:

- validation du sens de transmission A ou R,
- validation du code utilisateur d'après l'annuaire,
- validation du numéro de l'appelant (si demandée, auquel cas **X25** seul autorisé),
- validation de l'application,
- validation de la date (par défaut date du jour + incrément numérique paramétrable),
- validation du numéro d'ordre (par défaut 1),
- si sens **R**, validation de présence du fichier qui doit être déposé au préalable. Une option d'installation permet ou non de consommer le fichier plusieurs fois.

Lorsque l'accès est autorisé et valide, le transfert peut s'effectuer:

- dans le cas d'une réception (sens **A**), un fichier est alloué dynamiquement, et un événement de type message est inséré en fin de réception dans la file d'attente des messages. Un applicatif de traitement pourra être initialisé immédiatement par **TBT/400**, ou ultérieurement par vacation.
- dans le cas d'une émission (sens **R**), un événement de type acquittement est déposé dans la file d'attente des messages, pour informer l'applicatif. Cet événement n'est positionné que lors de la première consommation, si l'option consommation multiple est autorisée.

2.9.5. Double signature ou confirmation d'ordre

TBT/400 vous permet en standard d'utiliser une option de double signature: dans ce cas, le correspondant doit confirmer son ordre par l'envoi d'un deuxième mot de passe, soit lors de l'envoi du fichier, soit ultérieurement lors d'une autre transmission par l'envoi d'une deuxième carte paramètre avec ce deuxième mot de passe. Le fichier n'est considéré comme exploitable que lorsque les deux mots de passe valables ont été reçus.

3. Installation

3.1. Réquisits

3.1.1. Logiciels

TBT/400 utilise seulement l'**OS/400** en version **320** en architecture CISC, **370** et au delà, en architecture RISC et ne nécessite pas d'autres logiciels particuliers. **TBT/400** est un ensemble de jobs batch et de menus abrité dans son propre sous-système, et disposant de tous les objets systèmes de suivi et de contrôle (**JOBQUEUE**, ...).

Pour l'installation, il est nécessaire de disposer d'un user avec les habilitations ***SECADM** et ***ALLOBJ**, par exemple **QSECOFR**.

3.1.2. Matériels

Le progiciel est livré en standard sur cartouche pour CISC, CD-R pour RISC. Il peut être livré sur un autre support magnétique sur demande expresse.

Pour les options en **X25**, aucun matériel spécifique n'est à prévoir.

Pour les options en **X32**, un modem **X32**, avec éventuellement un identifiant ID32 (selon les partenaires), est nécessaire.

Pour l'option **ETEBAC 1-2**, un modem **BSC** par ligne est nécessaire.

3.1.3. Connexions

Pour les options en **X25**, vous devez bien sûr disposer au niveau de votre **IBM AS/400** d'une connexion et d'un abonnement **TRANSPAC**, avec au minimum **1 CVC** disponible pour **TBT/400**.

Pour l'option **ETEBAC 1-2**, vous devez disposer d'une ou plusieurs connexions téléphoniques, avec les modems **BSC** appropriés. Pour les options en **X32**, vous devez disposer d'une connexion téléphonique avec un modem **X32**, et un identifiant **ID32** fourni par **Transpac**.

Pour les options Réseau à Valeur Ajoutée en **X25** (par exemple **ATLAS 400**), vous devez bien sûr disposer au niveau de votre **IBM AS/400** d'une connexion et d'un abonnement **TRANSPAC**, avec en **X25** au minimum **2 CVC** disponibles pour **TBT/400** pour pouvoir émettre et recevoir simultanément.

De plus, pour les options Réseau à Valeur Ajoutée (par exemple **ATLAS 400**), vous devez disposer d'au moins une boîte aux lettres sur le serveur approprié. Si vous disposez déjà d'une boîte aux lettres que vous souhaitez maintenant dédicacer à **TBT/400**, assurez vous que les options demandées sont bien en place, notamment la remise directe conseillée pour **ATLAS 400**.

IPLS peut vous fournir les exemples de demande d'abonnement pour vous faciliter la tâche et éviter les aller-retour inutiles.

3.2. Bibliothèques du système

TBT/400 utilise quatre bibliothèques, chacune ayant un rôle très précis. Au global, il est nécessaire de disposer d'une place d'environ **50 millions** d'octets.

3.2.1. Bibliothèque IPLSP

Cette bibliothèque progiciel contient tous les objets, de type programme, commande, ..., livrés avec le progiciel. Elle doit impérativement être en phase avec les objets livrés: lors de l'installation d'une nouvelle version du progiciel, cette bibliothèque est entièrement recrée. En conséquence, elle n'est jamais modifiée sur le site, et ne nécessite donc pas de traitement de sauvegarde périodique. Place nécessaire: **50 millions** d'octets.

3.2.2. Bibliothèque IPLSC

Cette bibliothèque de configuration, créée automatiquement par la procédure d'installation, contient tous les objets customisés propres au site (tables, définitions réseau, ...), et l'historique des flux. Elle est conservée lors des installations de nouvelles versions du progiciel. Elle doit donc faire l'objet de sauvegardes régulières. Place nécessaire: **1 million** d'octets environ, variable en fonction de votre trafic et surtout des délais de péremption choisis.

3.2.3. Bibliothèque IPLSE

Cette bibliothèque d'exploitation, créée automatiquement par la procédure d'installation, contient tous les objets éphémères créés, non vitaux pour le fonctionnement du système, mais utiles pour votre propre exploitation. Une sauvegarde régulière, conforme à votre exploitation, doit donc être organisée. Place nécessaire: **1 million** d'octets environ, très variable en fonction de votre trafic et des délais de péremption choisis.

3.2.4. Bibliothèque IPLSM

Cette bibliothèque de messages, créée automatiquement par la procédure d'installation, contient tous les messages créés par l'éditeur intégré, non vitaux pour le fonctionnement du système, mais utiles pour votre propre utilisation. Une sauvegarde régulière, conforme à votre exploitation, doit donc être organisée. Place nécessaire: dépend exclusivement de l'utilisation (ou non) de l'éditeur intégré.

3.3. Sous-système TBT/400

TBT/400 évolue dans un sous-système propre, de nom **IP\$\$\$\$SUBS** (préfixe **IPS** par défaut), totalement et automatiquement généré avec l'ensemble des objets utiles, lors de la procédure d'installation. Ce paragraphe est donc purement informatif.

Ce sous-système dispose:

- d'une file d'attente de jobs (JOB QUEUE) de nom **IP\$\$\$\$JOBQ**,
- d'une file d'attente de sorties (OUTPUT QUEUE) de nom **IP\$\$\$\$OUTQ**,
- d'une file d'attente des messages (MESSAGE QUEUE) de nom **IP\$\$\$\$MSGQ**.

Pour "habiller" ses jobs, **TBT/400** utilise les JOB DESCRIPTION suivantes:

- **IP\$\$\$\$DISP** pour le noyau,
- **IP\$\$\$\$DRIV** pour les drivers de communication **Transpac**,
- **IP\$\$\$\$APPL** pour les applicatifs utilisateurs,
- **IP\$\$\$\$AUTO** pour le démarrage automatique de **TBT/400**,
- **IP\$\$\$\$JOB** pour les drivers de communication **BSC**,
- **IP\$\$\$\$JOB** pour la gestion de l'échéancier,
- **IP\$\$\$\$JOBM** pour la gestion des épurations (ménage automatique).

Dans chaque JOB DESCRIPTION, une donnée de routage, de même nom que la JOB, permet de référencer la classe d'exécution utilisée par le job grâce à la table adéquat du sous-système.

Les classes d'exécution utilisées sont les suivantes:

- **IP\$\$\$\$DISP** pour le noyau,
- **IP\$\$\$\$DRIV** pour les drivers de communication, **Transpac**,
- **IP\$\$\$\$APPL** pour les applicatifs utilisateurs,
- **IP\$\$\$\$AUTO** pour le démarrage automatique de **TBT/400**,
- **IP\$\$\$\$JOB** pour les drivers de communication **BSC**,
- **IP\$\$\$\$JOB** pour la gestion de l'échéancier,
- **IP\$\$\$\$JOBM** pour la gestion des épurations (ménage automatique).

A chaque classe est affectée une priorité d'exécution correspondant à la hiérarchie interne des jobs générés par **TBT/400**.

Ces paramètres sont tous initialisés lors de l'installation, et n'ont pas à être altérés; en effet, toute modification peut entraîner d'importantes baisses de performance du système.

3.4. Procédure d'installation

TBT/400 s'installe soit automatiquement, soit semi automatiquement ; votre environnement prévaudra dans ce choix. Quelque soit la procédure utilisée, **TBT/400** s'installe avec un USER disposant des habilitations ***SECADM** et ***ALLOBJ**, par exemple **QSECOFR**.

L'installation s'effectue en **30 minutes** environ selon votre **AS/400**.

Deux sites de téléchargement des dernières versions de **TBT/400** sont disponibles. Ces deux sites sont miroirs l'un de l'autre.

- <http://www.ipls400.com>
- <http://www.ipls400.net>

3.4.1. Procédure automatique

L'installation automatique est sujette à certaines conditions :

- Version **RISC**
- Terminologie **IPLSC IPLSP IPLSE IPLSM** respectée
- Code utilisateur **IP\$\$\$\$USER**
- Préfixe **IPS**
- Média de livraison disponible (pas de téléchargement)

3.4.2. Procédure semi automatique

En installant **TBT/400** semi automatiquement, cela permettra de définir le préfixe des bibliothèques de manière à avoir plusieurs versions du progiciel sur une même machine.

La procédure vous propose le menu de configuration pour modifier éventuellement les paramètres suivants: • 

- ***IP\$\$\$\$USER*** Code utilisateur de **TBT/400** (user utilisé par **TBT/400**).
- ***IPS*** Préfixe des noms du sous-système et des drivers (**IPS** par défaut).

Entrez vos valeurs, ou appuyez seulement sur Entrée pour les options par défaut.

La commande se poursuit et vous propose en final les menus **TBT/400**. Vous pouvez ainsi naviguer dans la supervision du système, afin d'effectuer les différentes mises à jour des configurations réseaux et lignes, les définitions des applications pouvant dialoguer avec le progiciel, gérer le suivi du trafic, ... La customisation minimale d'installation consiste à renseigner la clé de sécurité, pour **X32** à mettre dans la table des lignes les paramètres Type **X32** et Surveillance à Oui et indiquer le Nom de Ressource, et pour les options Réseau à Valeur Ajoutée à renseigner les paramètres de votre boîte.

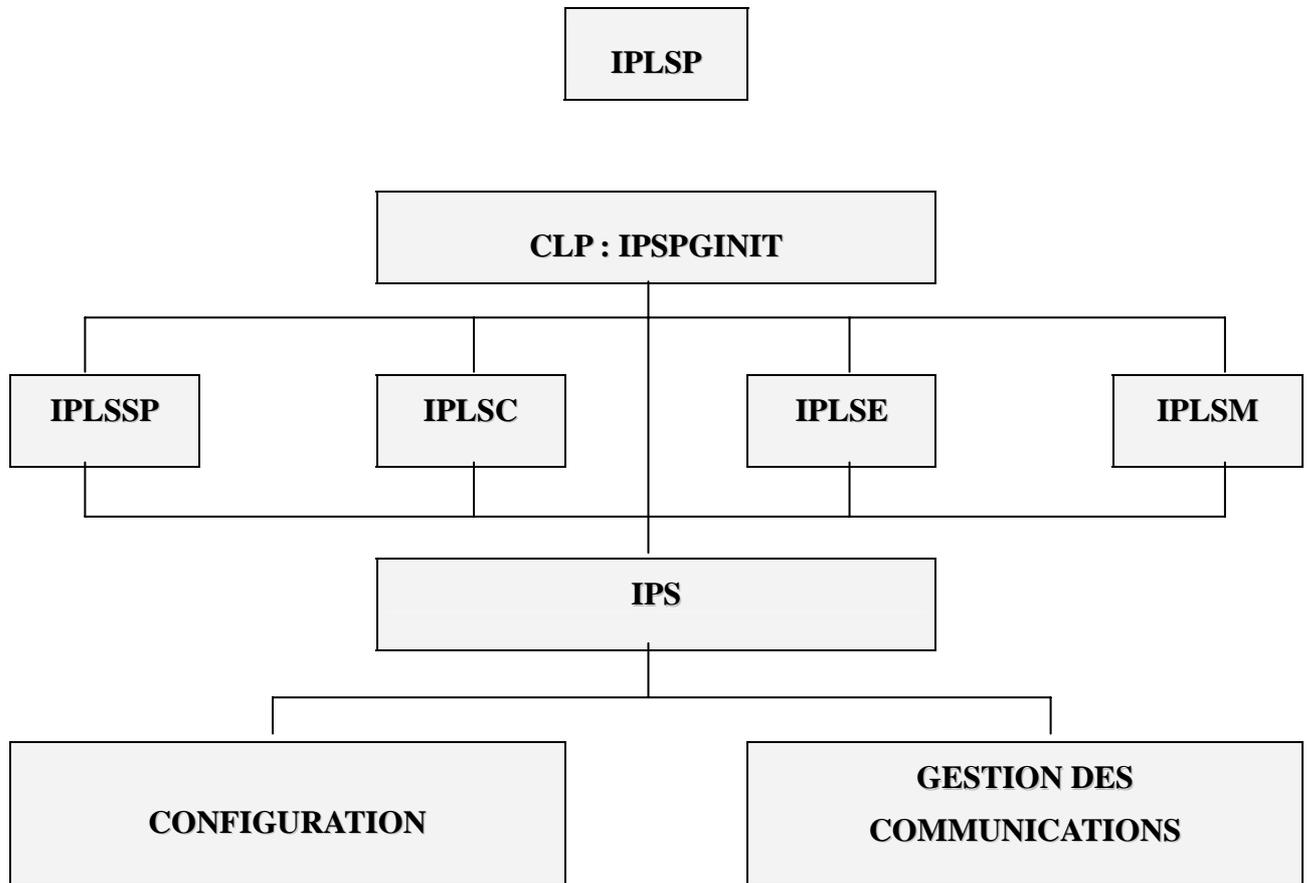
TBT/400 est alors opérationnel sur votre site; grâce aux différents menus et aux différentes commandes (**IPSNATLAS**, ...), vous pouvez transférer dès lors vos premiers fichiers vers vos correspondants. Le test d'installation consiste en l'envoi d'une télécopie pour tester l'émission, puis l'envoi d'un message en boucle pour tester la réception. Pour démarrer **TBT/400** à l'IPL, il faut intégrer une commande de lancement du sous-système **TBT/400** au démarrage du sous-système de base:

- **STRSBS SBS(*IPLSC* /*IP\$\$\$\$SUBS*)**

Ou utiliser la commande **IPLSP /IP\$STARTBT** livrée avec le progiciel.

• les noms en italiques peuvent être modifiés pour respecter vos contraintes méthodologiques; vous disposez alors de 1 à 9 caractères, suffixés par le caractère correspondant à la bibliothèque (par exemple P pour Progiciel).

3.5. Schéma d'ensemble de l'installation



3.6. Sécurité TBT/400

TBT/400 intègre des sécurités que vous pouvez adapter à votre environnement conformément à votre besoin. Elles sont basées sur le système de sécurité standard de l'**AS/400**.

Ainsi, **TBT/400** fonctionne sous un USER d'autorité ***ALLOBJ**, et de nom **IP\$\$\$\$USER**, qui lui est affecté lors de l'installation. Cela lui permet d'être le OWNER de tous les objets qui lui sont propres: bibliothèques du système, files d'attente, et tous les objets qu'il est amené à créer.

Les paramètres de sécurité générés lors de l'installation sont (pour le user **IP\$\$\$\$USER**, GRPPRF est à ***NONE**):

	IPLSP	IPLSC	IPLSE	IPLSM
Accès public bibli.	*USE	*EXCLUDE	*USE	*ADD *USE
CRTAUT bibliothèque	*EXCLUDE	*EXCLUDE	IPLSE¹	IPLSM²
TYPE bibliothèque	*PROD	*PROD	*PROD	*PROD
OWNER bibliothèque	IP\$\$\$\$USER	IP\$\$\$\$USER	IP\$\$\$\$USER	IP\$\$\$\$USER
Accès public objets	*EXCLUDE³	*EXCLUDE	IPLSE	IPLSM
OWNER objets	IP\$\$\$\$USER	IP\$\$\$\$USER	IP\$\$\$\$USER	IP\$\$\$\$USER

Ce qui donne lors de l'utilisation du progiciel:

	IPLSP	IPLSC	IPLSE	IPLSM
AUT objets	*EXCLUDE	*EXCLUDE	IPLSE	IPLSM
OWNER objets	IP\$\$\$\$USER	IP\$\$\$\$USER	USRPRF ou GRPPRF	USRPRF ou GRPPRF

Lors de réinstallation ultérieure (livraison d'une nouvelle version), les paramètres de sécurité seront régénérés de la même manière pour les quatre bibliothèques **IPLSP**, **IPLSC**, **IPLSE** et **IPLSM**.

Par contre, les deux listes d'autorisation **IPLSE** et **IPLSM**., créées lors de la première installation, ne seront jamais modifiées. Vous pouvez donc modifier leurs paramètres pour les adapter aux spécificités de votre site. (ces deux listes ont des droits publics définis à l'installation en ***EXCLUDE**).

Les USER utilisateurs ont accès aux API, et bien sûr aux fonctions d'Aide.

Pour l'accès aux **menus 1,2,4 et 5 de configuration ou supervision du système**, orienté homme système, le USER doit disposer de l'habilitation ***JOBCTL**.

Pour l'accès au **menu 2 de supervision du trafic**, orienté homme système et production, le USER doit disposer de l'habilitation ***JOBCTL**. Mais pour visualiser les textes des messages, il doit en plus disposer de ***SPLCTL**.

Pour l'accès au **menu 3 utilisateur**, orienté utilisateurs finaux, le USER utilisateur n'a accès qu'aux messages de son groupe d'appartenance, c'est à dire ceux qui sont sous la responsabilité de son USER ou des USERS qui ont le même groupe d'appartenance définit par GRPPRF.

-
1. Liste d'autorisation créée à l'installation
 2. Liste d'autorisation créée à l'installation
 3. sauf sur les objets nécessaires à l'utilisation de **TBT/400** qui ont alors ***USE** ou ***READ**

4. Exemples de programmes

NOTA: Ces exemples ne sont fournis ici qu'à titre purement documentaire. De nombreux exemples sont fournis en source avec TBT/400, dans les différents langages utilisés.

4.1. Emission d'un télex par Transpac

Ce programme en langage RPG 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.

```

*****
* Exemple d'appel de l'API TBT/AS400 *
* *
* Ce programme émet un Telex 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'
*****
* NBPARM indique le nombre de paramètres reçus
I           SDS
I                                     *PARMS  NBPARM
*
*****
*   décomposition des blocs de communication généraux
I/COPY IPSSAMPLES,IPSIRPAP
*
*****
* Parametre  PARMNU: Numéro d'appel
*           RTNCOD: Code Retour si different de Zéro alors
*           appel erroné.
*****

```

```

/EJECT
C      *ENTRY      PLIST
C              PARM          PARMNU 16
C              PARM          RTNCOD  1

*
* validation un paramètre 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 à envoyer: fichier + membre
C              MOVE WFILE   OBJFIL
C              MOVE WMEMB   OBJMBR
* alimente le numéro Telex
C              MOVELPARAMNU NUMTLX   P
* alimente l'auteur du courrier
C              MOVELWAUT    AUTHOR   P
* alimente l'objet du courrier
C              MOVELWOBJ    OBJECT   P
* alimente le A l'attention de ...
C              MOVELWATTN   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

```

4.2. Emission d'un fax par ATLAS 400

Ce programme en langage Cobol permet d'envoyer un message sous forme de fichier par le réseau ATLAS 400 sur un terminal Fax dont le numéro est fourni en paramètre.

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                ZPGCBFAX.
AUTHOR.                    IPLS.
DATE-WRITTEN.              1993.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY IPSICBAP.
LINKAGE SECTION.
01  PARM-NUFAX              PIC X(16).
PROCEDURE DIVISION USING PARM-NUFAX.

*validation un paramètre obligatoire
    IF PARM-NUFAX 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 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.

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

4.3. Réception d'un message (En C)

Ce programme en langage C permet de recevoir et de traiter les messages envoyés par un correspondant.

```

/*****
/* 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 */

/*****
/** Exemple 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. */
/*****
TBTLIT(wtbt_p1.appdes,"$$$$DEM"); /* Application à 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);
}

```

4.4. Réception d'un message (EN CLP)

Ce programme en langage **CLP** permet de recevoir et de traiter les messages envoyés par un correspondant.

```

PGM
DCL      VAR(&RTNCDP) TYPE(*DEC)  LEN(11) /* Code retour des API
*/
DCL      VAR(&KEYTBT) TYPE(*CHAR) LEN(16) /* Clé TBT du message lu
*/
DCL      VAR(&KEYUSR) TYPE(*CHAR) LEN(16) /* Clé applicative pour suivi
*/
DCL      VAR(&OBJLIB) TYPE(*CHAR) LEN(10) /* Bibliothèque du fichier
reçu*/
DCL      VAR(&OBJFIL) TYPE(*CHAR) LEN(10) /* Fichier reçu
*/
DCL      VAR(&OBJMBR) TYPE(*CHAR) LEN(10) /* Membre reçu
*/
DCL      VAR(&ACKTBT) TYPE(*CHAR) LEN(2)  /* Acquittement applicatif
*/
DCL      VAR(&LIBTBT) TYPE(*CHAR) LEN(128)/* Libellé applicatif
*/
DCL      VAR(&NOMLOG) TYPE(*CHAR) LEN(20) /* Correspondant émetteur
*/

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

/*****
/* BOUCLE DE TRAITEMENT */
*****/
BOUCLE:
      IPSRCVTBT  FNCDEM(R) EXCDEM(N) RTNCDP(&RTNCDP) +
                KEYTBT(&KEYTBT) OBJLIB(&OBJLIB) +
                OBJFIL(&OBJFIL) OBJMBR(&OBJMBR) +
                NOMLOG(&NOMLOG)          /* Appel de la commande de réception
*/

      IF          COND(&RTNCDP *NE 0 ) THEN(DO)
        SNDPGMSG  MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                  MSGDTA('Plus de Message pour l''Application') MSGTYPE(*COMP)
        GOTO      CMDLBL(ENDPGM)
      ENDDO

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

/*****
/* APPEL DE LA COMMANDE DE PURGE */
*****/
      CHGVAR     VAR(&KEYUSR) VALUE('Clé user')
      CHGVAR     VAR(&ACKTBT) VALUE('OK')
      CHGVAR     VAR(&LIBTBT) VALUE('Message consommé avec succès')
      IPSRCVTBT  FNCDEM(P) KEYTBT(&KEYTBT) KEYUSR(&KEYUSR) +
                ACKTBT(&ACKTBT) LIBTBT(&LIBTBT)
      GOTO      CMDLBL(BOUCLE)

CPF0000:
CPF9999:
IPS9999:
      SNDPGMSG   MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
                MSGDTA('Erreur Grave') +
                MSGTYPE(*ESCAPE)

MONMSG   MSGID(CPF0000)
ENDPGM:  ENDPGM

```

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

Service Documentation
Le Pascal - 23 bis avenue de l'Europe
78402 Chatou Cedex - France

Téléphone 33 (0) 1.30.15.70.80

Télécopie 33 (0) 1.30.15.70.91

33 (0) 1.30.15.70.96

Hot Line 33 (0) 1.30.15.70.99

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.

