



# TBT400 - GUIDE AS2

IPLS [technic@ipls.fr](mailto:technic@ipls.fr)

Version 0.2, 03/03/2023

# Notice

La reproduction, le transfert, la distribution ou le stockage de tout ou partie du contenu de ce document, sous quelque forme que ce soit, sans l'autorisation préalable de SysperTec Communication est interdite.

Tous les efforts possibles ont été mis en oeuvre par SysperTec Communication pour rendre ce document complet, pertinent et non inutilement redondant. En aucun cas SysperTec Communication ne peut être tenu responsable pour tout dommage, direct ou indirect, dû à des inexactitudes ou omissions dans cette documentation.

SysperTec Communication appliquant une méthode de développement continue, les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis, et ne sauraient constituer, de quelque manière que ce soit, un droit d'utilisation de tout ou partie des produits et marques citées.

SysperTec Communication et IPLS sont des marques déposées. Les autres noms de produits et de sociétés mentionnés dans ce document peuvent être des marques commerciales ou des noms de marques de leurs détenteurs respectifs.

# Table des matières

1. Gestion du document	4
2. Description option AS2	5
2.1. Concepts	5
2.1.1. Sécurité	5
2.1.2. Identifiants	5
3. Installation spécifique AS2	6
3.1. Pré-requis	6
3.2. Digital Certificate Manager (DCM)	6
3.3. DCM TBT400	6
3.4. Gestion des certificats électronique	8
3.4.1. Création des certificats par TBT400	8
3.4.2. Importation d'un certificat SSL dans le DCM IBM	8
En V5R4 et au delà	8
En V5R3M0	8
3.4.3. Importation d'un certificat dans le DCM TBT	9
Certificat public seul	9
Certificat public + clé privée	9
3.4.4. Assignment du certificat dans le DCM IBM	9
3.5. Sauvegardes des Certificats	10
3.6. Renouvellement des certificats	10
4. Initialisation d'une connexion avec un partenaire AS2	12
4.1. Considération TCP/IP pour le protocole AS2	12
4.2. Table des Hosts	12
4.3. Correspondant modèle : IPLSPROFIL	13
4.4. Création d'un correspondant AS2	13
4.4.1. Détail d'un correspondant	13
4.4.2. Détail d'un correspondant AS2	14
4.4.3. Détail d'un correspondant HTTP	15
4.4.4. Détail des paramètres TCP/IP	15
4.4.5. Détail des certificats	16
5. Utilisation de plusieurs jeux de certificats TBT400	17
5.1. Définition du besoin	17
5.2. Création d'un jeu de certificats supplémentaire	17
5.2.1. En V5R4 et au delà	17
5.2.2. En V5R3M0	18
5.2.3. Gestion des applications supplémentaires	18
5.2.4. Création des entrées dans l'annuaire et affectation des certificats	18
6. 5. Envoi d'un fichier	20
6.1. Utilisation	20
6.2. MDN reçu du partenaire distant	20
6.2.1. Définition	20
6.2.2. MDNASY='O'	21
6.2.3. MDNASY='N'	21
6.3. Champs principaux disponibles dans le programme de traitement d'acquittement	21
7. Réception d'un fichier	22
7.1. Utilisation	22
7.2. MDN émis par TBT400	23
7.2.1. MDNASY='O'	23
7.2.2. MDNASY='N'	24
8. Commandes spécifiques	25
8.1. Commande d'envoi	25

8.2. Commande de Statut .....	26
9. Intégration dans MultiTBT .....	27
10. Exemple d'implémentation de la commande IPSNDAS2 .....	31
10.1. Mise en situation .....	31
10.2. Terminologie .....	31
10.3. Émission .....	31
10.4. Réception .....	31
10.5. Paramétrage des applications et files d'attente .....	32
10.6. Sources des programmes CL .....	32
10.6.1. IPZAS2SEND .....	33
10.6.2. IPZAS2PCAK .....	34
10.6.3. IPZAS2TRAK .....	37
10.6.4. IPZAS2PCMG .....	39
10.6.5. IPZAS2TRMG .....	41

# 1. Gestion du document

Version	Commentaires	Statut	Auteurs	Date
0.1	Initialisation	Initié	Philippe CASSARD	4/01/2023
0.2	Modification	Finalisé	Philippe CASSARD	3/03/2023

## 2. Description option AS2

### 2.1. Concepts

#### 2.1.1. Sécurité

La nature des connexions **TCP/IP** via internet rend obligatoire l'utilisation du cryptage et de la signature électronique dès qu'il est question d'échange de données confidentielles.

Le protocole **AS2** permet de transférer des fichiers **EDI** de façon sécurisée par l'emploi (optionnel) de la fonctionnalité **SSL/TLS** et/ou des techniques de cryptographie les plus modernes.

Pour ce faire **AS2** nécessite l'utilisation de certificats électroniques qu'ils soient auto-signés ou délivrés par une autorité de certification reconnue.

Pour un même transfert, il est possible de n'utiliser qu'un seul et unique certificat pour toutes ces opérations mais, dans des circonstances particulières, il est possible d'utiliser :

- un certificat de signature électronique (xxx\_A)
- un certificat de cryptage (xxx\_E)
- un certificat pour la communication SSL (xxx)

TBT/400 permet la création de certificats dits « auto signés » et l'importation de certificats existants (voir le chapitre [Création des certificats par TBT400](#)).

#### 2.1.2. Identifiants

En **AS2** l'identification d'un partenaire se fait par les champs:

- AS2 local : partenaire local, il s'agit dans ce cas du site utilisateur de TBT400.
- AS2 distant : partenaire distant.

Le mode client et/ou serveur est toujours possible en AS2 ; ce choix est directement dépendant de la volonté des utilisateurs.

## 3. Installation spécifique AS2

### 3.1. Pré-requis

- L'**OS/400** doit être au niveau **V5R3M0** ou supérieur
- Les produits suivants doivent être installés
  - IBM HTTP Server for i5/OS DG1 base
  - Digital Certificate Manager SS1 option 34
  - CCA Cryptographic Service Provider SS1 option 35
  - Crypto Access Provider 128-bit AC3 (pour version OS/400 V5R3M0)

### 3.2. Digital Certificate Manager (DCM)

L'installation du Digital Certificate Manager (DCM) est **impérative si l'on choisit d'utiliser la fonctionnalité SSL** puisqu'il permet de gérer tout ce qui concerne les couches SSL..

Après avoir vérifié les pré-requis précédents, lancez la commande **STRTCPSVR SERVER(\*HTTP) HTTPSVR(\*ADMIN)** pour démarrer le serveur d'administration web utilisé par le DCM.

La commande **WRKACTJOB SBS(QHTTPSVR)** permet de visualiser les jobs actifs spécifiques aux différents serveurs web.

Ceux du serveur d'administration se nomment ADMIN, ADMIN1, ADMIN2, etc.

Si aucun job nommé ainsi n'existe dans le sous-système QHTTPSVR, il est très probable que le serveur d'administration web ne soit pas actif.

**Dans ce cas, il est possible de poursuivre le paramétrage AS2 mais la fonctionnalité SSL ne pourra pas être paramétrée.**

La commande **WRKJOB JOB(ADMIN)** peut vous aider à diagnostiquer un éventuel problème de configuration (il est aussi probable que le problème soit lié à un défaut de PTFs).

Si au contraire les jobs sont actifs, essayez de vous connecter au serveur admin depuis un navigateur web à l'adresse :

[http://adresse\\_iseries:2001](http://adresse_iseries:2001)

Après avoir saisi un nom d'utilisateur et un mot de passe cliquez sur « Pages de tâche IBM I » puis sur « Gestionnaire de certificat numérique » ou « Digital Certificate Manager ».

L'URL [http://adresse\\_iseries:2001/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0](http://adresse_iseries:2001/QIBM/ICSS/Cert/Admin/qycucm1.ndm/main0) donne un accès direct au « Gestionnaire de certificat numérique ».

Si la page s'affiche correctement, c'est que votre **DCM** est accessible et devrait vous permettre de paramétrer le **SSL** sur votre **I5/OS**.

Dans le cas contraire, vérifiez les messages d'erreurs et corrigez les problèmes avant de continuer le paramétrage **AS2+SSL** (à défaut, vous ne pourrez pas utiliser SSL).

### 3.3. DCM TBT400

Le **DCM** de **TBT400** est un répertoire IFS dans lequel sont stockés tous les certificats qu'il a à gérer, à savoir :

- Les certificats privés auto-signés créés par TBT400 ainsi que leur partie publique,
- Les certificats privés délivrés par une autorité reconnue et importés par la commande IPSCERTIFS ainsi que leur partie publique,
- Les certificats publics importés par la commande IPSCERTIFS ou par la fonctionnalité « apprentissage » du mode SSL.

Plusieurs suffixes sont utilisés pour différencier chaque type de certificats :

- .P12 : certificat protégé par mot de passes et contenant un couple clé privée/clé publique,
- .crt : certificat public associé à une clé privée stockée dans un « keystore »,
- .cer : certificat ne contenant qu'une clé publique.

Les noms utilisés pour stocker les certificats sont précisés dans le menu « Certificats » du correspondant dans les champs « K ».



Pour les certificats locaux ceci n'est vrai que si le fichier IFS en question existe, autrement il peut s'agir du nom de l'application du DCM IBM ou de la clé d'accès au keystore TBT400. Ceci est lié à l'algorithme de recherche des certificats de TBT400 (DCM TBT puis DCM IBM enfin keystore TBT).

Par exemple (avec les valeurs par défaut affichées) :

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/16 15.36.54
DCRT  9973 Rect          Détail des certificats          IPLSPC  IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS

Certificat local Ssl . IPSTBTSUBS          K IPSTBTSUBS_CLI
                                          IPSTBTSUBS_SRV
                                          IPSTBTSUBS_AUT

Certificat remote Ssl . IPLS          K IPLS
Certificat local Aut .          K
Certificat remote Aut .          K
Certificat local Sig . IPSTBTSUBS          K IPSTBTSUBS_APP
Certificat remote Sig . IPLS          K IPLS
Certificat local Cry . IPSTBTSUBS          K IPSTBTSUBS_APP
Certificat remote Cry . IPLS          K IPLS
Certificat local Avd .          K
Certificat remote Avd .          K

F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cer F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas          Copyright Informatique Pour Les Sociétés
15365480 IPS QPADEV0007 TBT/400 Affichage des valeurs par défaut
```

Ici le champ « Certificat local sig » correspond au certificat utilisé pour signer un fichier et contient donc la clé privée du site.

Sa clé « K » est « IPSTBTSUBS\_APP » ce qui correspond au fichier IFS suivant :

```
'/IFSTBTIPSC/IPSTBTSUBS_APP.p12'
```

Le champ « Certificat remote Cry » correspond au certificat utilisé pour crypter un fichier et contient donc la clé publique du partenaire distant.

Sa clé « K » est « IPLS » ce qui correspond au fichier IFS suivant :

```
'/IFSTBTIPSC/IPLS.cer'
```



Pour les certificats locaux SSL, la clé correspond toujours au nom de l'application du DCM IBM.

## 3.4. Gestion des certificats électronique

### 3.4.1. Création des certificats par TBT400

A cet effet utilisez la commande suivante :

```
IPSCRTCERT COMNAME('Client')
            LOCALITY('Paris')
            STATE('Ile de France')
            COUNTRY('FR')
```

PS : Les autres champs de cette commande ne sont pas à modifier pour une installation standard (et pour un premier certificat) ; dans ce cas le nom de stockage du certificat est le nom crée à partir du nom du sous-système, ici « IPSTBTSUBS\_APP ».

Une fois la commande exécutée correctement TBT/400 vous propose une « vue » de l'IFS où est stocké le certificats (répertoire /IFSTBTIPSC).

A partir de ce moment, le certificat est disponible dans TBT400 et peut être consulté, par exemple, dans le menu « Certificat » du correspondant.



Le champ « REPLACE » est positionné à 'NO' par défaut pour éviter l'écrasement d'un éventuel certificat nommé « xxx.p12 ».

### 3.4.2. Importation d'un certificat SSL dans le DCM IBM

L'utilisation du DCM IBM est obligatoire pour l'utilisation de la fonctionnalité SSL. Il convient de se reporter à la documentation IBM pour réaliser correctement les manipulations suivantes.

#### En V5R4 et au delà

Pour importer un certificat SSL crée par TBT400 et situé dans le répertoire '/IFSTBTIPSC » :

- Dans le DCM importer le certificat dans le magasin « \*SYSTEM » sous le nom IPSTBTSUBS\_APP (le mot de passe est « PASSWORD »),
- Si le certificat existe déjà dans le DCM, ne pas le remplacer,
- Supprimer ce fichier de l'IFS pour ne pas l'avoir en deux exemplaires (pour SSL, le DCM IBM est la référence).

La manipulation est similaire pour un autre type de certificat, seuls changent le chemin de stockage et le mot de passe.

#### En V5R3M0

Dans cette version du système, il est impossible d'importer un certificat auto-signé comme ceux créés par TBT400. un certificat SSL crée par TBT400, la solution est donc de le créer directement dans le DCM :

- Renommer le certificat IPSTBTSUBS\_APP.p12 en IPS530SUBS\_APP.p12,
- Dans le DCM, générer un certificat CA,
- Dans le DCM, générer un certificat de type client/serveur nommé IPSTBTSUBS\_APP,
- Ne pas supprimer le fichier de l'IFS.

### 3.4.3. Importation d'un certificat dans le DCM TBT

#### Certificat public seul

A cet effet utilisez la commande suivante :

```
IPSCERTIFS CRTFNC(*DEL)
            TYPRES($$$$AS2)
            NOMLOG(IPLS)
            CRTCTX(*REMCRY)
            IFSOBJ('/home/certs/certificatCRY.cer')
```

Avec:

- CRTFNC(\*DEL): « Delivery mode », ajoute un certificat public (si déjà présent l'ancien est conservé, le nouveau est suffixé « \*NEW »)
- CRTCTX(\*REMCRY): Certificat de cryptage distant (remote)
- IFSOBJ('xxxxxxx'): Chemin IFS du certificat à importer

Le certificat public contenu dans le fichier '/home/certs/certificatCRY.cer' est maintenant stocké dans le DCM TBT sous le nom prévu dans l'annuaire AS2 pour le correspondant IPLS.

#### Certificat public + clé privée

A cet effet utilisez la commande suivante :

```
IPSCERTIFS CRTFNC(*INT)
            TYPRES($$$$AS2)
            NOMLOG(IPLS)
            CRTCTX(*LOCCRY)
            IFSOBJ('/home/certs/certificatCRY.p12')
            PASSWO(PASSWORD)
```

Avec:

- CRTFNC(\*INT): « Internal mode », ajoute un certificat privé (si déjà présent l'ancien est conservé, le nouveau est suffixé « \*NEW »)
- CRTCTX(\*LOCCRY): Certificat de cryptage local
- IFSOBJ('xxxxxxx'): Chemin IFS du certificat à importer
- PASSWO : Mot de passe d'accès au fichier PKCS12

Le certificat privé contenu dans le fichier '/home/certs/certificatCRY.p12' est maintenant stocké dans le DCM TBT sous le nom prévu dans l'annuaire AS2 pour le correspondant IPLS.

### 3.4.4. Assignment du certificat dans le DCM IBM

La procédure d'installation a créé quatre applications (au sens DCM) ; il s'agit d'associer l'application au certificat SSL précédemment importé (ou généré) :

- Associer IPSTBTSUBS\_APP à IPSTBTSUBS\_CLI(TBT/400)SslCli
- Associer IPSTBTSUBS\_APP à IPSTBTSUBS\_SRV(TBT/400)SslSrv
- Associer IPSTBTSUBS\_APP à IPSTBTSUBS\_AUT(TBT/400)SslAut
- Associer IPSTBTSUBS\_APP à IPSTBTSUBS\_APP(TBT/400)

## 3.5. Sauvegardes des Certificats

Deux répertoires doivent être pris en compte lors des opérations de sauvegardes de votre système :

- /QIBM/USERDATA/ICSS (contient tout le paramétrage SSL système)
- /ifstbtIPSC (contient les certificats privées TBT400 ainsi que les certificats publics des partenaires).

## 3.6. Renouvellement des certificats

Le renouvellement des certificats client s'effectue via la même commande que pour la création de certificats vue au chapitre [Création des certificats par TBT400](#):

```
IPSCRTCERT COMNAME('Client')
            SUFFIX(*NEW)
            LOCALITY('Paris')
            STATE('Ile de France')
            COUNTRY('FR')
            REPLACE(*YES)
```

Le champ « Suffixe = \*NEW » demande à créer une nouvelle version du certificat.

A noter l'utilisation du champ « REPLACE » positionné à « \*YES » pour autoriser l'écrasement d'un éventuel certificat nommé « xxx\_NEW.p12 ».

Dans ce cas, le nouveau certificat vient cohabité avec l'ancien ; pour imposer l'utilisation du plus récent, le champ correspondant dans l'annuaire doit être suffixé par « \*NEW ».

Par exemple :

```
TBT400 V710 M005          Recette      sur IPLSG          2023/01/16 16.21.25
DCRT  9973 Rect          Détail des certificats      IPLSPC    IPLSG
Type d'annuaire . . . . $$$SAS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS              Type réseau . $$$SAS2
Libellé correspondant . TBT400 IPLS

Certificat local  Ssl .                  K

Certificat remote Ssl .                  K
Certificat local  Aut .                  K
Certificat remote Aut .                  K
Certificat local  Sig . *NEW             K
Certificat remote Sig .                  K
Certificat local  Cry .                  K
Certificat remote Cry .                  K
Certificat local  Avd .                  K
Certificat remote Avd .                  K

F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cer F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas                      Copyright Informatique Pour Les Sociétés
```

Avec les valeur par défaut :

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/16 16.22.19
DCRT  9973 Rect          Détail des certificats          IPLSPC          IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS

Certificat local Ssl . IPSTBTSUBS          K IPSTBTSUBS_CLI
                                          IPSTBTSUBS_SRV
                                          IPSTBTSUBS_AUT

Certificat remote Ssl . IPLS          K IPLS
Certificat local Aut .          K
Certificat remote Aut .          K
Certificat local Sig . IPSTBTSUBS_NEW          K IPSTBTSUBS_APP_NEW
Certificat remote Sig . IPLS          K IPLS
Certificat local Cry . IPSTBTSUBS_NEW          K IPSTBTSUBS_APP_NEW
Certificat remote Cry . IPLS          K IPLS
Certificat local Avd .          K
Certificat remote Avd .          K

F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cer F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas          Copyright Informatique Pour Les Sociétés
16221982 IPS QPADEV0007 TBT/400 Affichage des valeurs par défaut
```

Le certificat local de signature pointe donc sur le fichier '/ifstbtips/ IPSTBTSUBS\_A\_APP\_NEW.p12'.

Il est à noter que si rien n'est précisé, la configuration cryptage est déduite de la configuration signature (locale ou distant).

Une autre solution consiste tout simplement à renouveler un certificat en écrasant son ancienne version par la commande suivante :

```
IPSCRTCERT COMNAME('Client')
SUFFIX(*CUR)
LOCALITY('Paris')
STATE('Ile de France')
COUNTRY('FR')
REPLACE(*YES)
```



Dans ce cas, il n'y a rien d'autre à faire mais **attention** cependant :  
L'exécution de cette commande est irréversible puisque l'ancien certificat est écrasé.  
Il ne sera, par exemple, plus possible de décrypter un fichier crypté avec cette clé.

## 4. Initialisation d'une connexion avec un partenaire AS2

### 4.1. Considération TCP/IP pour le protocole AS2

Le protocole AS2 utilise la pile de protocoles TCP/IP et, de ce fait, le paramétrage TCT/IP de votre i5/OS doit être correct et en particulier en ce qui concerne le client DNS.

En effet, les serveurs distants seront, dans 99% des cas, connus par ce que l'on appelle leur « Nom d'hôte » ou « Hostname ».

Il est donc fortement recommandé de paramétrer le client DNS de votre i5/OS de façon à ce qu'il soit capable de résoudre chacun de ces noms.

Pour vérifier ce paramétrage, depuis une ligne de commande saisissez :

- GO TCPADM (appel du menu « TCP/IP Administration »),
- « 1. Configure TCP/IP »,
- « 12. Change TCP/IP domain information »,
- Vérifiez le champ INTNETADR qui devrait être renseigné en fonction des adresses correspondantes à vos serveurs DNS ou, à défaut, à celles de votre fournisseur d'accès à Internet.

Avant même tout paramétrage de TBT/400 vous devriez pouvoir réaliser la commande suivante :

**ping 'NOM DU PARTENAIRE'**

La commande PING doit renvoyer :

*Verifying connection to xxx at address 111.222.333.444.*

xxx étant le nom d'hôte du serveur distant et 111.222.333.444 son adresse résolue par l'un des serveurs DNS.

Il est à noter que la commande PING peut ne pas être concluante et s'achever avec le message :

*Connection verification statistics: 0 of 5 successful (0 %)*

Ce n'est pas nécessairement une erreur et peut simplement vouloir dire que le serveur ne « répond pas » à cette commande.

**Pour notre test, il semble que le seul message d'erreur problématique soit :**

*Unknown host*

ou « *Hôte inconnu* » indiquant l'incapacité du client DNS à résoudre le nom d'hôte.

### 4.2. Table des Hosts

Il est nécessaire d'avoir accès à un serveur DNS, les serveurs diffusant des noms de host.

Cependant, peu de serveurs disposent d'une résolution inverse correcte. Pour assurer un suivi correct, et parfois améliorer les performances, il est souhaitable de définir les Serveurs en table des Hosts depuis le menu système :

**GO TCPADM**

## 4.3. Correspondant modèle : IPLSPROFIL

Ce correspondant n'est pas un correspondant réel ; il fournit des valeurs par défaut à l'ensemble des correspondants AS2.

En particulier :

- Le nom des certificats utilisés
- Le profil par défaut d'émission (avec ou sans CR/LF, ....)
- Le profil de réception

## 4.4. Création d'un correspondant AS2

Depuis une ligne de commande :

- Entrez dans TBT/400 : **IPLSP/IPS**
- 4. *Gestion de l'annuaire*
- 1. *Définition des correspondants*
- F10 sur le correspondant « IPLS ».

### 4.4.1. Détail d'un correspondant

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/16 18.32.36
DANG  9941 Rect          Détail d'un correspondant          IPLSPC          IPLSG
Type d'annuaire . . . . $$$AS2          F4          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$AS2          F4
Libellé correspondant . TBT400 IPLS          Inactif          N
Commentaire utilisateur

Auteur . . . . .
Objet . . . . .

          Suffixe N O,N          Trace          O,N
A l'attention de . . .          Impre.          O,N,C,B
Référence du message .          Scrut.          O,N
Emission mode puits . .          O,N          Messages demandés . . .          O,N,C,B
Accusé demandé . . . .          O,N,C          Avis =====> Distri          Lectur          Applic
Mode transparent . . . . O          O,N          Ajout CR/LF . . . . N N O,N Lrec
Suppression des blancs N          O,N,L          Transfert ASCII . . . .          Ccsi          65533
Priorité réseau . . . .          N,U,H          Enreg. par segment . .          1          0 - 255
R. txt Lr          Tr B Ty          C          297 Ec I R. bin Lr          Tr B Ty          C          297 Ec I
Identifiant réseau . .          Ha          1 C          1 S          1 Cm          6
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F13=Hau F19=Gau F20=Dro F21=Def
F24=Bas          Copyright Informatique Pour Les Sociétés
```

Les champs importants dans cet écran sont :

- Accusé demandé : Précise si l'application émettrice (au sens TBT400) désire recevoir un accusé d'acheminement de ce dernier,
- Avis =====> Distri : Voir « Mdn Asynchrone » de l'écran suivant,
- Transfert ASCII (émission ASCII/EBCDIC) = ASCDEM (associé au CCSID),
- Les champs de la ligne 20 permettent de préciser un mode de réception spécifique à un transfert (longueur d'enregistrements, gestion CR/LF, etc.) comme cela à toujours été le cas dans tous les protocoles de TBT/400,
- Ha, C, S, Cm : permettent de préciser les algorithmes utilisés respectivement pour le calcul de **H**ash, le **C**ryptage, la **S**ignature électronique et la **C**ompression.



Les algorithmes des fonctions cryptographiques pour un partenaire donné sont à configurer en collaboration avec ce dernier.

Après avoir validé les données appuyer sur F20 pour éditer l'écran suivant

## 4.4.2. Détail d'un correspondant AS2

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/17 09.23.53
DAS2  9934 Rect          Détail d'un correspondant AS2          IPLSPC          IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS

AS2 local . .
AS2 distant . IPLS
Mdn Asynchrone  O,N
Mdn Signé . .  O,N,A

Payload . . .

Loc MDN Host
Loc MDN Adr .

Sélection d'application A A,O,D          Application par défaut $TRANSMI F4
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F13=Hau F19=Gau F20=Dro F21=Def
F24=Bas          Copyright Informatique Pour Les Sociétés
```

Les champs importants dans cet écran sont :

<b>AS2 local</b>	Identifiant AS2 du site local (TBT400),
<b>AS2 distant</b>	Identifiant AS2 du site distant,
<b>Mdn Asynchrone</b>	MDN en mode synchrone/asynchrone,
<b>Loc MDN Host et Adr</b>	Nom d'hôte et adresse IP à utiliser par le partenaire distant pour envoyer le MDN (mode asynchrone uniquement).

Après avoir validé les données appuyer sur F20 pour éditer l'écran suivant

### 4.4.3. Détail d'un correspondant HTTP.

Dans cet écran vous pouvez spécifier l'url http du serveur distant.

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/17 09.41.03
DHTP  9994 Rect          Détail d'un correspondant Http      IPLSPC      IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS              Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS

HTTP User . .
HTTP Password
HTTP Post URL

HTTP Ans URL

Filename . .

Codepage Cli
Codepage Srv

Sélection d'application A A,O,D          Application par défaut $TRANSMI F4
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cnf F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas                          Copyright Informatique Pour Les Sociétés
```

Si rien n'est spécifié TBT/400 sur connectera à la racine du serveur :

- HTTP Post URL : URL à joindre,
- Filename : nom de fichier tel que présenté par le protocole AS2 (n'est pas nécessairement lié au nom réel dans le système de fichier).

Après avoir validé les données appuyer sur F20 pour éditer l'écran

### 4.4.4. Détail des paramètres TCP/IP

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/17 10.02.58
DTCP  9947 Rect          Détail des paramètres TCP/IP      IPLSPC      IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS              Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS

Hostname IP distant . . iplsads1.ipls.net

Adresse IP distant . . 83.167.42.242

Port IP distant . .

Hostname IP local . . .

Adresse IP locale . . .

Usage adresse . . . . .

Utilisation Ssl . . . . Option Protocole Cipher Exit Lng

Buffer Emission . . . .
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F13=Hau F19=Gau F20=Dro F21=Def
F24=Bas                          Copyright Informatique Pour Les Sociétés
```

Après avoir validé les données appuyer sur F20 pour éditer l'écran suivant

## 4.4.5. Détail des certificats

Dans cet écran vous pouvez paramétrer les certificats à utiliser pour cette connexion.

Appuyer sur F21 pour afficher les valeurs par défaut.

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/17 10.04.29
DCRT  9973 Rect          Détail des certificats          IPLSPC          IPLSG
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$AS2
Libellé correspondant . TBT400 IPLS
```

```
Certificat local Ssl . IPSTBTSUBS          K IPSTBTSUBS_CLI
                                          IPSTBTSUBS_SRV
                                          IPSTBTSUBS_AUT
Certificat remote Ssl . IPLS          K IPLS
Certificat local Aut .          K
Certificat remote Aut .          K
Certificat local Sig . IPSTBTSUBS          K IPSTBTSUBS_APP
Certificat remote Sig . IPLS          K IPLS
Certificat local Cry . IPSTBTSUBS          K IPSTBTSUBS_APP
Certificat remote Cry . IPLS          K IPLS
Certificat local Avd .          K
Certificat remote Avd .          K
```

```
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cer F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas          Copyright Informatique Pour Les Sociétés
10042964 IPS QPADEV0005 TBT/400 Affichage des valeurs par défaut
```

F10 sur un des champs « K » permet d'obtenir le détail du certificat tel qu'il est utilisé par TBT400.

# 5. Utilisation de plusieurs jeux de certificats TBT400

## 5.1. Définition du besoin

Un certificat unique peut être utilisé pour communiquer en SSL et pour réaliser des opérations de cryptographie (chiffrement, signature électronique, etc.).

Ce même certificat peut être utilisé face à un ou plusieurs partenaires, cependant il peut vous être demandé d'utiliser un certificat spécifique pour une opération particulière.

Certains partenaires peuvent aussi exiger l'utilisation de certificats différents :

- un pour le cryptage,
- un pour la signature électronique,
- un pour le SSL.

## 5.2. Création d'un jeu de certificats supplémentaire

Utilisez la commande suivante :

```
IPSCRTCERT COMNAME('COMPTE2')
LOCALITY('Paris')
STATE('Ile de France')
COUNTRY('FR')
CERTNAME(*CLI)
BROWSE(*YES)
```

où 'COMPTE2» représente le nom du deuxième certificat.

Une fois la commande exécutée correctement TBT/400 vous propose une « vue » de l'IFS où sont stockés les certificats dans le répertoire '/IFSTBTIPSC' :

- COMPTE2\_APP.p12.

Dans cet exemple, le certificat créé est de type « SSL », pour créer un certificat dédié à une utilisation particulière, il faut positionner le champ « USAGE » de façon adéquate :

- USAGE = SSL : le certificat s'appelle « COMPTE2\_APP.p12 », son utilisation est limitée à la fonctionnalité SSL,
- USAGE = SIG : le certificat s'appelle « COMPTE2\_A\_APP.p12 », son utilisation est limitée à la signature électronique et le suffixe « \*S » doit être précisé dans le correspondant,
- USAGE = CRY : le certificat s'appelle « COMPTE2\_E\_APP.p12 », son utilisation est limitée chiffrement de fichier et le suffixe « \*S » doit être précisé dans le correspondant,
- USAGE = ALL : le certificat s'appelle « COMPTE2\_APP.p12 », son utilisation n'est pas limitée,
- USAGE = AUT : inutilisé en AS2,
- USAGE = AVD : inutilisé en AS2.

### 5.2.1. En V5R4 et au delà

Le certificat SSL peut, ensuite, être importé dans le DCM IBM en procédant ainsi :

- Dans le DCM importer le certificat système sous le nom COMPTE2\_APP (le mot de passe est 'PASSWORD')
- Si le certificat existe déjà dans le DCM, ne pas le remplacer

- Supprimer ce fichier de l'IFS (/IFSTBTIPSC) pour supprimer tous risques de confusion
- Associer le certificat à l'application SSL désirée.

## 5.2.2. En V5R3M0

**Le certificat SSL ne peut pas être importé dans cette version du DCM IBM**, il faut en procéder ainsi :

- Renommer le certificat COMPTE2\_APP.p12 en COMPTE2530\_APP.p12
- Dans le DCM IBM, générer une autorité de certification locale
- Dans le DCM IBM, générer un certification système signé par cette autorité et appelé COMPTE2\_APP
- Ne pas supprimer le fichier de l'IFS (il pourra être utilisé pour la cryptographie)
- Associer le certificat à l'application SSL désirée.

## 5.2.3. Gestion des applications supplémentaires

Il est possible de paramétrer TBT/400 pour qu'il utilise un certificat SSL plutôt qu'un autre, grâce à la notion d'application du DCM.

La procédure d'installation de TBT400 ne crée que les quatre applications par défaut, **celles qui seront utilisées pour les nouveaux jeux de certificat devront être créés manuellement** (seule l'application de type « Client » est à créer pour les jeux de certificats supplémentaires) :

- Dans le DCM, sélectionnez le « Certificate store \*SYSTEM » puis,
- « Manage Applications »,
- « Add Application »,
- « Client - Add a client application »,

Dans le menu « Add Application » renseignez les champs suivants (les autres peuvent conserver leur valeur par défaut) :

- « Application ID » : COMPTE2\_CLI,
- « Define the CA trust list » : NO,
- « Certificate revocation processing » : YES,
- « Application description » : COMPTE2\_CLI (TBT/400)SsCli.

Il ne reste qu'à associer le certificat COMPTE2\_APP à l'application COMPTE2\_CLI(TBT/400)SsCli.

## 5.2.4. Création des entrées dans l'annuaire et affectation des certificats

Une fois l'application créée et le certificat assigné vous pouvez éditer le correspondant et renseigner les champs suivants :

- Certificat local Ssl : COMPTE2
- Certificat local Sig : \*S
- Certificat local Cry : \*S

afin d'obtenir l'écran suivant (F21 pour avoir les valeurs par défaut):

```
TBT400 V710 M005          Recette          sur IPLSG          2023/01/17 14.36.19
DCRT  9973 Rect          Détail des certificats          IPLSPC          IPLSG
Type d'annuaire . . . . $$$SAS2          Portée . . . . *GLOBAL
Nom du correspondant . IPLS          Type réseau . $$$SAS2
Libellé correspondant . TBT400 IPLS

Certificat local Ssl . COMPTE2          K

Certificat remote Ssl .          K
Certificat local Aut . COMPTE2_X          K
Certificat remote Aut .          K
Certificat local Sig . COMPTE2_A          K
Certificat remote Sig .          K
Certificat local Cry . COMPTE2_E          K
Certificat remote Cry .          K
Certificat local Avd .          K
Certificat remote Avd .          K

F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cer F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas          Copyright Informatique Pour Les Sociétés
```

En appuyant sur F10 sur chacun des certificats locaux, vous pouvez vérifier la sélection des certificats prévus pour ce correspondant.

## 6. 5. Envoi d'un fichier

### 6.1. Utilisation

Il est possible d'envoyer un fichier :

- Par menu :
  - IPLSP/IPS,
  - choisir le menu « Émission d'un fichier »,
- Par la commande [IPLSP/IPSNDAS2](#).

Le mode classique d'utilisation de TBT/400 est d'employer la commande d'émission IPSNDAS2.

Plusieurs champs sont à renseigner obligatoirement avant de pouvoir émettre un fichier :

- Les qualifiants du fichier à envoyer :
  - OBJLIB,OBJFIL,OBJMBR pour un fichier natif IBMi,
  - OBJFIL(\*IFS), IFSDIR, IFSOBJ pour un fichier IFS,
- Le partenaire cible de l'opération (NOMLOG).

D'autres champs sont optionnels mais permettent d'obtenir une configuration particulière pour une émission AS2 :

- MDNASY : Choix du mode synchrone ou asynchrone pour le MDN (avis de distribution AS2),
- MDNSIG : Signature du MDN demandée,
- IPSMFN : Nom du fichier tel qu'annoncé par le protocole (choix purement applicatif et n'étant pas forcément lié au nom réel du fichier),
- etc

Exemple d'une émission de fichier AS2 :

```
IPLSPNDAS2 NOMLOG(EDIPAS2USRATOB)
KEYUSR('Test AS2')
OBJFIL(IPSSAMPLES)
OBJLIB(IPLS510P)
OBJMBR(IPZIGBAN)
IPSMFN(MONFICHIER.TXT)
```

Cette commande permet d'envoyer le fichier IPLSP/IPSSAMPLES(IPZIGBAN) au partenaire EDIPAS2USRATOB en lui annonçant le nom 'monfichier.txt' et en utilisant un MDN asynchrone.

### 6.2. MDN reçu du partenaire distant

#### 6.2.1. Définition

Le MDN (Message Disposition Notification) permet de s'assurer du bon déroulement d'une émission AS2.

Il peut être traité en mode synchrone ou asynchrone et signé selon le choix des deux partenaires.

**Bien qu'il ne soit pas obligatoire en AS2, il est fortement conseillé de l'utiliser dans le cadre du suivi de vos transferts de fichiers.**

## 6.2.2. MDNASY='O'

Le MDN est traité en mode asynchrone et le champs AVIDIS (Avis de distribution) est forcé à 'O'.

TBT/400, suite à un transfert infructueux, remontera à l'application émettrice un code acquittement 'KO', 'CR', ..., (tout code différent de 'OK', ' ' (Blanc) ou 'PC').

TBT/400, suite à un transfert réussi, remontera à l'application émettrice, le code acquittement 'PC', signifiant que le transfert s'est bien passé, et que le partenaire remontera un statut (MDN) ultérieurement

Lors de la réception du MDN, TBT/400 remontera à l'application émettrice :

- Soit un code acquittement ' ' (Blanc), signifiant que le partenaire a validé (au sens réseau) le transfert,
- Soit un code acquittement 'ED' (Erreur de distribution) signifiant un problème quant au fichier reçu ; le libellé d'acheminement LIBTBT précise le problème (Erreur de décryptage, Erreur de Signature, par exemple).



Il est à noter que dans ce cas, le programme de traitement d'acquittements peut être appelé deux fois.

## 6.2.3. MDNASY='N'

Le MDN est traité en mode synchrone et le champs AVIDIS (Avis de distribution) est forcé à 'N'.

TBT/400, suite à un transfert infructueux, remontera à l'application émettrice le code acquittement en erreur (idem cas précédent MDNASY= 'O')

TBT/400, suite à un transfert réussi, remontera directement à l'application émettrice un code acquittement final ' ' (Blanc).

Si un MDN est reçu ultérieurement, il sera ignoré.

## 6.3. Champs principaux disponibles dans le programme de traitement d'acquittement

Cette liste est non exhaustive :

- KEYUSR : Clé utilisateur (valorisé par le programme d'émission),
- COMUSR : Commentaire (valorisé par le programme d'émission),
- ACKTBT : Code acquittement,
- LIBTBT : Libellé d'acheminement,
- IPSMFN : Nom de fichier AS2.

# 7. Réception d'un fichier

## 7.1. Utilisation

La réception d'un fichier est entièrement gérée par TBT400 pour ce qui est de la phase de communication AS2.

Le choix du type de fichier crée à l'arrivée (natif, IFS, etc.), de la traduction ASCII/EBCDIC effectuée, du profil utilisateur associé au nouvel objet, etc. dépendent de **la file d'attente de réception utilisée (type « M »)**, elle-même déduite de **l'application paramétrée pour l'événement entrant**.

Si aucune application n'est précisée, TBT400 utilise sa valeur par défaut précisée dans le paramétrage global (nommée AS2 lors d'une installation standard du progiciel).

Une fois le fichier reçu, deux possibilités :

- Aucun programme de consommation n'est associé à la file d'attente de type « M » : l'événement entrant est **directement placé dans le fichier historique** (menu « Supervision de l'historique »),
- Un programme de consommation est associé à la file d'attente de type « M » : le traitement applicatif est alors exécuté. L'événement entrant ne sera placé dans le fichier historique que **lorsque le traitement applicatif sera achevé** (menu « Supervision de l'historique »).



Il existe un **risque de blocage dans le traitement de l'événement entrant** si le programme de consommation part en boucle pour une raison ou une autre. Dans ce cas l'événement restera dans le menu « Supervision des messages » et ne pourra être considéré comme étant achevé. Les programmes de consommation sont de **la responsabilité du site utilisateur**, il convient donc de veiller à la qualité de son développement, en particulier dans sa gestion des erreurs/exceptions.

Il est donc **fortement conseillé d'utiliser le modèle** de programme de consommation standard dans TBT400 :

- IPLSP/IPSSAMPLES(IPSPADUMMC) : version longue
- IPLSP/IPSSAMPLES(IPSPADUMMY) : version courte

Après avoir copié ce modèle, libre à vous de le modifier pour appeler un programme utilisateur dédié dans lequel se trouve toute la logique de votre traitement applicatif.

L'exemple suivant propose un programme minimaliste pouvant être appelé par un programme de consommation basé sur IPSPADUMMY :

```

/* Programmes N°2 : Traitement du fichier reçu */

PGM PARM(&OBJLIB &OBJFIL &OBJMBR) /* PGM appelé par TBT/400 */

DCL VAR(&OBJLIB) TYPE(*CHAR) LEN(10) /* Fichier TBT/400 */
DCL VAR(&OBJFIL) TYPE(*CHAR) LEN(10)
DCL VAR(&OBJMBR) TYPE(*CHAR) LEN(10)
DCL VAR(&DSTLIB) TYPE(*CHAR) LEN(10) /* Fichier utilisateur */
DCL VAR(&DSTFIL) TYPE(*CHAR) LEN(10)
DCL VAR(&DSTMBR) TYPE(*CHAR) LEN(10)

CHGVAR VAR(&DSTLIB) VALUE('BIBUTIL')
CHGVAR VAR(&DSTFIL) VALUE('FICUTIL')
CHGVAR VAR(&DSTMBR) VALUE('MBRUTIL')

CPYF FROMFILE(&OBJLIB/&OBJFIL)
      TOFILE(&DSTLIB/&DSTFIL)
      FROMMBR(&OBJMBR)
      TOMBR(&DSTMBR)

/*
/*
/* Gestion des erreurs puis
/* traitement du fichier représenté par & DSTLIB, &DSTFIL et &DSTMBR
/*

```

Le programme reçoit en paramètre l'identifiant du fichier reçu dans TBT400 (&OBJLIB, &OBJFIL et &OBJMBR) et se contente de le copier dans un fichier utilisateur.

## 7.2. MDN émis par TBT400

### 7.2.1. MDNASY='O'

Le MDN est traité en mode asynchrone.

TBT/400, suite à une réception réussie, émettra le MDN dans **une nouvelle connexion AS2** à destination du partenaire afin de lui transmettre le résultat de son traitement.

**Avantage** : Le transfert de gros fichier ne pose pas de problème puisque TBT400 envoie le MDN une fois le traitement AS2 terminé.

**Inconvénient** : Les deux partenaires ne sont plus synchronisés.

Dans ce cas de figure, le MDN n'est envoyé dans une nouvelle connexion AS2 remontera à l'application émettrice, le code acquittement 'PC', signifiant que le transfert s'est bien passé, et que le partenaire remontera un statut (MDN) ultérieurement.

Lors de la réception du MDN, TBT/400 remontera à l'application émettrice :

- Soit un code acquittement ' ' (Blanc), signifiant que le partenaire a validé (au sens réseau) le transfert,
- Soit un code acquittement 'ED' (Erreur de distribution) signifiant un problème quant au fichier reçu ; le libellé d'acheminement LIBTBT précise le problème (Erreur de décryptage, Erreur de Signature, par exemple).

Le mode classique d'utilisation de TBT/400 est d'employer la commande d'émission IPSNDAS2.

Plusieurs champs sont à renseigner obligatoirement avant de pouvoir émettre un fichier :

- Les qualifiants du fichier à envoyer :
  - OBJLIB,OBJFIL,OBJMBR pour un fichier natif IBMi,
  - OBJFIL(\*IFS), IFSDIR, IFSOBJ pour un fichier IFS,
- Le partenaire cible de l'opération (NOMLOG).

D'autres champs sont optionnels mais permettent d'obtenir une configuration particulière pour une émission AS2 :

- MDNASY : Choix du mode synchrone ou asynchrone pour le MDN (avis de distribution AS2),
- MDNSIG : Signature du MDN demandée,
- IPSMFN : Nom du fichier tel qu'annoncé par le protocole (choix purement applicatif et n'étant pas forcément lié au nom réel du fichier),
- etc.

Exemple d'une émission de fichier AS2 :

```
IPSNAS2 NOMLOG(EDIPAS2USRATOB)
KEYUSR('Test AS2')
OBJFIL(IPSSAMPLES)
OBJLIB(IPLS510P)
OBJMBR(IPZIGBAN)
IPSMFN(MONFICHIER.TXT)
```

Cette commande permet d'envoyer le fichier IPLSP/IPSSAMPLES(IPZIGBAN) au partenaire EDIPAS2USRATOB en lui annonçant le nom « monfichier.txt » et en utilisant un MDN asynchrone.

### 7.2.2. MDNASY='N'

Le MDN est traité en mode synchrone.

TBT/400, suite à une réception réussie, émettra le MDN dans **la même connexion AS2**, c'est à dire avant de couper la communication ou de recevoir un nouveau fichier.

Avantage : Le partenaire peut être averti au plus tôt d'une éventuelle erreur de traitement au sens AS2 (erreur de décryptage, de validation de signature, etc.).

Inconvénient : Si le partenaire envoie un fichier demandant un temps de traitement trop long, le « Timeout » réseau (local ou distant) peut être dépassé provoquant une rupture de communication.

## 8. Commandes spécifiques

### 8.1. Commande d'envoi

La commande IPSNDAS2 permet d'envoyer un fichier en protocole AS2.

#### Emission AS2 (IPSNDAS2)

```
Nom logique du correspondant . . . . . NOMLOG
Clé utilisateur . . . . . KEYUSR
Objet à traiter: Fichier . . . . . OBJFIL
Objet à traiter: Bibliothèque . . . . . OBJLIB
Objet à traiter: Membre . . . . . OBJMBR
Fonction demandée . . . . . FNCDEM
Fonction début demandée . . . . . DEBDEM
Fonction fin demandée . . . . . FINDEM
Fonction exception demandée . . . . . EXCDEM
Fonction trace demandée . . . . . TRADEM
Fichier dupliqué demandé . . . . . DUPDEM
Application émettrice . . . . . APPEME
Application destinatrice . . . . . APPDES
Date d'envoi différé . . . . . DATDIF
Heure d'envoi différé . . . . . HORDIF
Date limite d'envoi . . . . . DATPER
Heure limite d'envoi . . . . . HORPER
Accusé demandé . . . . . ACKDEM
Suppression fichier demandée . . . . . SUPDEM
Emission mode puit . . . . . PUIDEM
Impression demandée . . . . . IMPDEM
Break message demandé . . . . . BRKDEM
Scrutation implicite . . . . . SCRDEM
Commentaire utilisateur . . . . . COMUSR
Hauteur de page pour télécopie . . . . . HAUPAG
Environnement demandé . . . . . SETENV
Code utilisateur souhaité . . . . . USRDEM
Mode synchrone . . . . . SYNDEM
Filename . . . . . IPSMFN
Content-Type . . . . . CTNTYP
MDN asynchrone . . . . . MDNASY
MDN signé . . . . . MDNSIG
Format fichier . . . . . FMTFIC
Taille fichier . . . . . SIZFIC
Ajout de CR/LF . . . . . CRLDEM
Ajout de CR/LF fin . . . . . CRLFIN
Suppression blancs . . . . . SPADEM
Traduction ASCII . . . . . ASCDEM
Ccsid demandé . . . . . CCSID
```

## 8.2. Commande de Statut

La commande IPSSTAS2 permet de compléter les informations fournies par la commande IPSRCVTBT pour le protocole AS2.

### Statut AS2 (IPSSTAS2)

```
Client Date . . . . . IPHTDC
Server Date . . . . . IPHTDS
User agent . . . . . IPHTUA
Server . . . . . IPHTSR
Local AS2 . . . . . AS2LOC
Remote AS2 . . . . . AS2REM
Filename . . . . . IPSMFN
Content-Type . . . . . CTNTYP
MDN asynchrone . . . . . MDNASY
MDN signé . . . . . MDNSIG
```

### Autres paramètres

```
Fonction fin demandée . . . . . FINDEM
Fonction exception demandée . . . . . EXCDEM
Fonction trace demandée . . . . . TRADEM
Code retour (Num. étendu) . . . . . RTNCDP
Libellé du compte rendu . . . . . MSGTXT
```

## 9. Intégration dans MultiTBT

Comme tous les protocoles utilisés par TBT400, AS2 s'intègre naturellement dans le concept MultiTBT.

La marche à suivre pour créer une nouvelle liaison vers un partenaire distant est la suivante :

- **L'environnement MultiTBT doit être opérationnel** (configuration MultiTBT serveur et clients, communication établie entre les différents correspondants directs, applications et files d'attente spécifiques correctement paramétrées, etc.),
- Créer un nouveau correspondant AS2 sur le serveur MultiTBT dont le nom logique **commence obligatoirement par « EDI »** (par exemple : « EDIPLS ») ; l'intérêt est de pouvoir valider la communication et le transfert d'un simple fichier de test (code page, longueur d'enregistrements, etc.),
- Effectuer un test d'émission et de réception de fichier avec le partenaire distant ; **ces tests doivent être concluant avant de poursuivre**,
- Sur un des clients MultiTBT, utiliser **la commande IPSNDEDI** pour émettre un fichier de test vers le correspondant « EDIxxx » qui n'existe pour l'instant que sur le serveur : l'intérêt est de provoquer la création automatique sur le client des correspondants intermédiaires et nécessaires au fonctionnement de MultiTBT ; le fichier émis doit avoir été envoyé au partenaire distant via le serveur, **ce test valide la configuration dans le sens « Client MultiTBT vers partenaire distant » et doit être concluant avant de poursuivre**,
- Sur le serveur MultiTBT, éditer le correspondant « EDIxxx »,
- Dans l'écran « Détail d'un correspondant AS2 », saisir le nom de l'application de routage entrant dans le champ « Application par défaut » et saisir « A » dans le champ « Sélection d'application » ; l'intérêt est double car cela permet :
  - d'activer le routage MultiTBT pour chaque fichier reçu de la part de ce partenaire,
  - indiquer au client MultiTBT la file d'attente de traitement à utiliser,
- Effectuer un test de réception de fichier avec le partenaire distant. Le fichier doit être reçu sur le serveur (comme précédemment) puis, être acheminé vers le client MultiTBT via un correspondant indirect de protocole TBT. Sur le client, le fichier doit être traité par l'application de même nom que celle utilisée par le serveur et grâce à laquelle n'importe quel programme utilisateur peut être appelé (exp : un traducteur EDI) ; **ce test valide la configuration dans le sens « Partenaire distant vers client MultiTBT ».**

Les écrans suivants montrent le détail d'un transfert AS2 dans MultiTBT.

Vue de l'historique sur le serveur MultiTBT :

```
TBT400 V710 M005          Recette          sur IPLSD          2023/03/02 16.26.07
SUH0  0023 Rect          Supervision de l'historique  IPLSPC          IPLSD
Fatt          Bib          Rés          Cor          Types M M
Emet          Fic          Grp          Loc          Ack .
Dest          Mbr          Usr          Prot
Datfpc          Horfpc          Refmsg
Keyusr          Keyext
F Date et Heure          Adresse réduite          Clé utilisateur Ak
O d'insertion          du destinataire
  230302 153450 EI AS2 EDIR7AS2MEC JNKS-as2.txt          JNKS-as2.txt
  230302 153512 IE TBT ORVR7AS2MEC EDIR7AS2MEC,ROUTV01          JNKS-as2.txt
```

```
F1=Hlp F2=Trk F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Zoo F11=Cur F13=Hau
F14=Acr F15=Spr F16=Ace F17=Spe F18=Tra F19=Gau F20=Dro F21=Dsp F22=Pdm F23=Obj
```

Avec :

- Un événement de type « EI » en protocole TBT et représentant la liaison client/serveur MultiTBT,
- Un événement de type « IE » en protocole AS2 et représentant la liaison vers le partenaire distant,
- « EDIR7AS2MEC » : Nom du correspondant **AS2** utilisé pour joindre le partenaire distant,
- « ORVR7AS2MEC » : Nom du correspondant TBT utilisé pour joindre le client MultiTBT.

Vue de l'historique sur le client MultiTBT :

```
TBT400 V710 M004          Recette          sur IPLSV          2023/03/02 16.25.39
SUH0  0023 Rect          Supervision de l'historique  IPLSPC          IPLSV
Fatt          Bib          Rés          Cor          Types M M
Emet          Fic          Grp          Loc          Ack .
Dest          Mbr          Usr          Prot
Datfpc          Horfpc          Refmsg
Keyusr          Keyext
F Date et Heure          Adresse réduite          Clé utilisateur Ak
O d'insertion          du destinataire
230302 153513 EI TBT EDIR7AS2MEC ORVR7AS2MEC,ROUTV01 JNKS-as2.txt
```

```
F1=Hlp F2=Trk F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Zoo F11=Cur F13=Hau
F14=Acr F15=Spr F16=Ace F17=Spe F18=Tra F19=Gau F20=Dro F21=Dsp F22=Pdm F23=Obj
```

Avec :

- Un événement de type « IE » en protocole TBT et représentant la liaison client/serveur MultiTBT,
- « EDIR7AS2MEC » : Nom du correspondant **TBT** utilisé par l'appliquatif,
- « ORVR7AS2MEC » : Nom du correspondant TBT utilisé pour joindre le serveur MultiTBT.

Sur le serveur, détail du transfert entre le client et le serveur :

```
TBT400 V710 M005          Recette          sur IPLSD          2023/03/02 16.30.15
ZTBT  9993 Rect          Détail d'un message mode TBT 1/2  IPLSPC          IPLSD
Fa M$EXTERNB Em ROUTV01 De $EXTERNB Bi /IfsTBTIPS Fi In20230302 Mb D3WRGBKH6A
Annu $$$TBT *GLOBAL ORVR7AS2MEC Rés $$$TBT Util IPSTBTUSER
Typ M M Cl TBT 000B451500559ED4F2F7F6F3F6F40001 Cl Uti JNKS-as2.txt Ack
-----
Clé TBT Emission . . 000B451500559ED4F2F7F6F3F6F40001 Fax
Clé TBT Réception . . 000B451500559F2CF4F2F2F3F1F40001 -----Indirect-----
Application distante . ROUTV01 Profil . *TBT Cor IPLSVTCP
Code utilisat. distant EDIR7AS2MEC Usr IPLSDTCP
Mot de passe distant . FROMD Psw FROMD
Code retour . Lng. enreg . . 131072 Nbr. enreg . . 1
Environ. . . . Rect Libellé Recette sur IPLSV Système . IPLSV
Type . . . . 8202 Modèle . . . . E4B Série . . 06623EP Groupe . . . . P05
Version TBT . 710 Modification . 4 OS/400 . . . V7R2M0 Avis Distr. . . . 0
Ccsid . . . 1147 Code Pays . . . FR Chrid . . . . . 0000000695000001147
Nom distant . . . EDIR7AS2MEC App dist . ROUTV01 Compression . . . 2
Rmt Lib /IfsTBTIPS Rmt Fil In20230302 Rmt Mbr D35MM9KGL6 Type de fichier . I
Org Lib /IfsTBTIPS Org Fil In20230302 Org Mbr D3WRGBKH6A App. Init.
Rmt Lib *NONE Rmt Pgm *NONE Rmt Sna *NONE *NONE - *NONE
Rmt Job Z_TCP00001IPSTBTUSER422314 Rmt Lec SMROUTV01 IPSTBTUSER422314
F1=Hlp F2=Trk F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Txt F11=Edt F13=Hau
F14=Acr F15=Spr F16=Ace F17=Spe F18=Tra F19=Gau F20=Dro F21=Dsp F22=Pdm F23=Obj
```

Sur le client, détail du transfert entre le client et le serveur :

```
TBT400 V710 M004          Recette          sur IPLSV          2023/03/02 16.31.39
ZTBT  9993 Rect          Détail d'un message mode TBT 1/2          IPLSPC          IPLSV
Fa MROUTV01 Em $EXTERNB De ROUTV01 Bi /IfsTBTIPS Fi In20230302 Mb D35MM9KGL6
Annu $$$TBT *GLOBAL      EDIR7AS2MEC          Rés $$$TBT      Util IPSTBTUSER
Typ M M Cl TBT 000B451500559F2CF4F2F2F3F1F40001 Cl Uti JNKS-as2.txt      Ack
-----
Clé TBT Emission . . 000B451500559ED4F2F7F6F3F6F40001 Fax
Clé TBT Réception . . 000B451500559F2CF4F2F2F3F1F40001 -----Indirect-----
Application distante . ROUTV01 Profil . *TBT          Cor IPLSDTCP
Code utilisat. distant ORVR7AS2MEC          Usr IPLSVTCP
Mot de passe distant . FROMD          Psw FROMV
Code retour .          Lng. enreg . . 1048576 Nbr. enreg . . 1
Environ. . . . Rect Libellé Recette          sur IPLSD          Système . IPLSD
Type . . . . 8202 Modèle . . . . E4D Série . . 8459A6V Groupe . . . . P05
Version TBT . 710 Modification . 5 OS/400 . . . V7R2M0 Avis Distr. . . . 0
Ccsid . . . 1147 Code Pays . . . FR Chrid . . . . . 0000000695000001147
Nom distant . . . ORVR7AS2MEC          App dist . $EXTERNB Compression . . . 2
Rmt Lib /IfsTBTIPS Rmt Fil In20230302 Rmt Mbr D3WRGBKH6A Type de fichier . I
Org Lib /IfsTBTIPS Org Fil In20230302 Org Mbr D3WRGBKH6A App. Init. ROUTV01
Rmt Lib *NONE          Rmt Pgm *NONE          Rmt Sna *NONE          *NONE          - *NONE
Rmt Job SMROUTV01 IPSTBTUSER276364          Rmt Lec Z_TCP00021IPSTBTUSER276277
F1=Hlp F2=Trk F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Txt F11=Edt F13=Hau
F14=Acr F15=Spr F16=Ace F17=Spe F18=Tra F19=Gau F20=Dro F21=Dsp F22=Pdm F23=Obj
```

Sur le serveur, détail du transfert entre le serveur et le partenaire distant :

```
TBT400 V710 M005          Recette          sur IPLSD          2023/03/02 16.33.43
ZHTP  9996 Rect          Détail d'un message Http          IPLSPC          IPLSD
Fa MROUTV01 Em $EXTERNB De ROUTV01 Bi /IfsTBTIPS Fi In20230302 Mb D3WRGBKH6A
Annu $$$AS2 *GLOBAL      EDIR7AS2MEC          Rés $$$AS2      Util IPSTBTUSER
Typ M M Cl TBT 000B451500559638F2F7F6F3F5F70001 Cl Uti JNKS-as2.txt      Ack
-----
User . . . .

Post URL . .
Ans URL . . http://www.company.com:8080/as2/HttpReceiver
User Agent . mendelson_opensource AS2 1.1 build 49 - www.mendelson-e-c.com
Server . . . Rect-TBT400-V710M005
Clé . . . . <mendelson_opensource_AS2-1677767686637-7@EDIR7AS2IPSREMD_EDIR7A
Date Cli/Srv Thu, 02 Mar 2023 15:34:46 CET      Thu, 02 Mar 2023 14:34:46 GMT
CONNECT . . .
Date . . . 20230302143449873586
Via . . . .
Filename . . MONFICHIER.TXT

Retour HTTP .
Codepage Cli 65534          Codepage Srv 1252
F1=Hlp F2=Trk F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Txt F11=Edt F13=Hau
F14=Acr F15=Spr F16=Ace F17=Spe F18=Tra F19=Gau F20=Dro F21=Dsp F22=Pdm F23=Obj
```

Dans cet exemple le nom de fichier AS2 est « MONFICHIER.TXT ». Il est précisé depuis le client MultiTBT lors de l'utilisation de la commande IPSNDEDI.

Pour cela, le principe de substitution de champs TBT400 est utilisé.

Il est en effet possible, depuis le client, de substituer certains champs du protocole TBT à certains champs du protocole AS2.

Par exemple, en paramétrant ainsi le correspondant « EDIR7AS2MEC » sur le serveur :

```
TBT400 V710 M005          Recette          sur IPLSD          2023/03/02 16.38.54
DHTP  9994 Rect          Détail d'un correspondant Http          IPLSPC          IPLSD
Type d'annuaire . . . . $$$AS2          Portée . . . . *GLOBAL
Nom du correspondant . EDIR7AS2MEC          Type réseau . $$$AS2
Libellé correspondant . Recette AS2 MEC

HTTP User . .
HTTP Password
HTTP Post URL as2/HttpReceiver

HTTP Ans URL

Filename . . &OBJECT

Codepage Cli
Codepage Srv

Sélection d'application A A,O,D          Application par défaut ROUTV01 F4
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F10=Cnf F13=Hau F19=Gau F20=Dro
F21=Def F24=Bas          Copyright Informatique Pour Les Sociétés
```

et en exécutant la commande suivante sur le client :

```
IPSNDEDI NOMLOG(EDIR7AS2MEC)
OBJFIL(IPSSAMPLES)
OBJMBR(IPZIGBAN)
OBJECT(MONFICHIER.TXT)
```

La valeur du champ « objet » (« OBJECT ») du protocole TBT sera automatiquement substituée au champ « Nom de fichier » (« IPSMFN » ou « Mail Filename ») du protocole AS2).

# 10. Exemple d'implémentation de la commande IPSNDAS2

## 10.1. Mise en situation

Il s'agit d'implémenter la commande IPSNDAS2 pour réaliser deux types d'opérations :

- Envoi d'un fichier vers un correspondant AS2,
- Réception d'un fichier depuis un correspondant AS2.

Nous faisons ici le choix de séparer les programmes d'émission et de réception mais ce n'est que pour en faciliter les explications.

## 10.2. Terminologie

Comme à son habitude TBT/400 utilise les notions d'application, de files d'attente et de programmes de consommation :

- AS2APP : Application (au sens TBT/400),
- AAS2APP : File d'attente des acquittements,
- IPZAS2PCAK : Programme de consommation des acquittements reçus (lie TBT/400 à IPZAS2TRAK),
- IPZAS2TRAK : Nom du CL de traitements des acquittements reçus (programme utilisateur),
- MAS2APP : File d'attente des messages,
- IPZAS2PCMG : Programme de consommation des messages reçus (lie TBT/400 à IPZAS2TRMG),
- IPZAS2TRMG : Nom du CL de traitements des messages reçus.

## 10.3. Émission

Déroulement du processus d'émission :

- Phase n°1 : Le programme utilisateur IPZAS2SEND appelle TBT/400,
  - Soumission de la commande « Émission de fichier » utilisant l'application émettrice « AS2APP »,
  - Traitement des codes retour (de la soumission),
  - Fin du programme IPZAS2SEND avec messages d'erreurs si besoin,
- Phase n°2 : TBT/400 appelle le programme IPZAS2TRAK (traitement acquittements TBT/400) pour chaque soumission effectuée,
  - Test des codes retours,
  - Traitement des erreurs éventuelles,
  - Fin du programme IPZAS2TRAK avec messages d'erreurs si besoin.

## 10.4. Réception

Déroulement du processus de réception :

- TBT/400 appelle le programme IPZAS2TRMG pour chaque fichier entrant,
- Traitement du fichier reçu dans l'applicatif client,
- Alimentation des codes retour TBT/400 permettant de mettre à jour l'historique,
- Fin du programme IPZAS2TRMG avec messages d'erreurs si besoin.

## 10.5. Paramétrage des applications et files d'attente

Pour créer l'application AS2APP :

- Entrez dans TBT/400 : IPLSP/IPS,
- «1. Configuration du système »,
- «3. Définition des applications »,
- Positionnez le curseur sur l'application AS2 (créée par TBT/400),
- Saisissez « AS2APP » puis ENTER, ce qui aura pour effet de créer une nouvelle application sur le modèle de l'application AS2 (cette dernière n'étant pas modifiée).

Les valeurs par défaut sont suffisantes pour notre exemple.

Pour créer les files d'attentes AAS2APP et MAS2APP :

- Entrez dans TBT/400 : IPLSP/IPS,
- «1. Configuration du système »,
- «4. Définition des files d'attente »,
- Positionnez le curseur sur la file d'attente MAS2 (créée par TBT/400),
- Saisissez « MAS2APP » puis ENTER, ce qui aura pour effet de créer une nouvelle file d'attente sur le modèle de MAS2 (cette dernière n'étant pas modifiée),
- Positionnez maintenant le curseur sur la file d'attente MAS2APP et modifiez les champs suivants puis « ENTER » :
  - Nom du programme : IPZAS2PCMG (programme de consommation des messages),
  - Nom de la biblio pgm : Votre bibliothèque utilisateur.

Idem pour la file d'attente AAS2APP en utilisant cette fois le modèle MAS2APP et le programme IPZAS2PCAK.

Là aussi, les valeurs par défaut sont suffisantes.

Le paramétrage application + files d'attente est terminé et, dès ce moment, chaque fois que TBT/400 recevra un message utilisant l'application MAS2APP le programme de consommation IPZAS2PCMG sera appelé avec toutes les variables nécessaire au traitement applicatif.

Il en va de même pour les acquittements.

## 10.6. Sources des programmes CL

Tous ces sources sont disponibles dans le fichier « IPLSP/IPSSAMPLES ». Il vous suffit de les copier dans une bibliothèque utilisateur avant de commencer à les exploiter.

Avant de pouvoir les compiler, il faut impérativement avoir la bibliothèque programme de TBT400 en ligne car les commandes ne sont volontairement pas qualifiées.

Pour une installation standard, la commande suivante peut être utilisée :

**CHGCURLIB CURLIB(IPLSP)**



Il ne faut jamais modifier directement ces sources car il seront systématiquement écrasés lors de la prochaine montée de version de TBT400.

Avant de pouvoir exploiter ces programmes, l'environnement AS2 de TBT400 doit être opérationnel et les applications/files d'attentes du chapitre [Paramétrage des applications et files d'attente](#) doivent être configurées correctement.



```

/* Ici : IPSRCVTBT interroge le statut de la dernière commande afin */
/* d'en récupérer le code retour &RTNCDP */
/*
/* FNCDEM(L) : L pour Last (dernière commande émise) */
/* DEBDEM(N) : Ne pas démarrer l'environnement */
/* : (déjà fait par IPSNDAS2) */
/* FINDEM(O) : Clôturer l'environnement */
/* : (si dernière commande TBT du programme) */
/* EXCDEM(N) : Pas d'exception en cas d'erreurs */
/*
/*
/* IPSRCVTBT FNCDEM(L) DEBDEM(N) FINDEM(O) EXCDEM(N) +
/* RTNCDP(&RTNCDP) KEYTBT(&KEYTBT) +
/* KEYUSR(&KEYUSR) MSGTXT(&MSGTXT)
/*
/* MONMSG MSGID(CPF0000)
/*
/* IF COND(&RTNCDP *NE 0) THEN(GOTO CMDLBL(ERREUR))

/* Autres traitements... */

/* Pas d'erreur... */
/* SNDPGMMSG MSG('Appel TBT/400 pour émission OK')
/* GOTO CMDLBL(FIN)

ERREUR: SNDPGMMSG MSG('Erreur de traitement...')
/* SNDPGMMSG MSG('MSGTXT: ' *CAT &MSGTXT)
/* GOTO CMDLBL(FIN)

/* Autres traitements d'erreurs... */

FIN: ENDPGM

```

## 10.6.2. IPZAS2PCAK

Programme de consommation lié à la file d'attente de type A et appelé par TBT400 pour chaque acquittement reçu.

```

/*****
/* Ceci est le source du programme "dummy" de consommation */
/* d'une file d'attente. Il est destiné à servir de modèle. */
/* Ne remplacez pas le programme IPSPADUMMY dans la bibliothèque */
/* du progiciel (IPLSP). Une version plus complète est fournie */
/* sous le nom IPSPADUMMC. */
/*****
/* This is the source of the "IPSPADUMMY" program. It must be */
/* used as a skeleton program and duplicated in customer library */
/* for modifications. */
/*****
PGM
DCL VAR(&DEBDEM) TYPE(*CHAR) LEN(1) VALUE(O)
DCL VAR(&RTNCDP) TYPE(*DEC) LEN(11)
DCL VAR(&KEYTBT) TYPE(*CHAR) LEN(16)
DCL VAR(&KEYUSR) TYPE(*CHAR) LEN(16)
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(&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(&KEYEXT)      TYPE(*CHAR) LEN(32)
DCL      VAR(&USRPRF)      TYPE(*CHAR) LEN(16)
DCL      VAR(&IFSOBJ)     TYPE(*CHAR) LEN(5000)

DCL      VAR(&JOB)         TYPE(*CHAR) LEN(10)
DCL      VAR(&USER)        TYPE(*CHAR) LEN(10)
DCL      VAR(&NBR)         TYPE(*CHAR) LEN(6)
DCL      VAR(&MSGCMD)      TYPE(*CHAR) LEN(64)
DCL      VAR(&MSGACK)     TYPE(*CHAR) LEN(256)

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

RTVJOBA  JOB(&JOB) USER(&USER) NBR(&NBR)
CHGVAR   VAR(&MSGCMD) VALUE('WRKJOB JOB(' *TCAT &NBR +
      *TCAT '/' *TCAT &USER *TCAT '/' *TCAT +
      &JOB *TCAT '))

ITER:
/*****
/* APPEL DE LA COMMANDE DE RECEPTION */
/*****
/* CALL RECEIVE COMMAND */
/*****
      IPSRCVTBT  FNCDEM(R) DEBDEM(&DEBDEM) FINDEM(C) +
      EXCDEM(N) TRADEM(0) RTNCDP(&RTNCDP) +
      KEYTBT(&KEYTBT) KEYUSR(&KEYUSR) +
      ACKTBT(&ACKTBT) LIBTBT(&LIBTBT) +
      OBJLIB(&OBJLIB) OBJFIL(&OBJFIL) +
      OBJMBR(&OBJMBR) USRPRF(&USRPRF) +
      DATFPC(&DATFPC) HORFPC(&HORFPC) +
      DATFTR(&DATFTR) HORFTR(&HORFTR) +
      DATRPC(&DATRPC) HORRPC(&HORRPC) +
      DATRTR(&DATRTR) HORRTR(&HORRTR) +
      SUPDEM(&SUPDEM) COMUSR(&COMUSR) +
      NOMLOG(&NOMLOG) KEYEXT(&KEYEXT) /* Appel +
      des API de TBT/400 via la Command +
      IPSRCVTBT */

      IF          COND(&RTNCDP *NE 0) THEN(GOTO +
      CMDLBL(ENDPGM)) /* Plus rien dans la file +
      d'attente */

      CHGVAR      VAR(&DEBDEM)  VALUE('N')

      IPSSTP1     IFSOBJ(&IFSOBJ)

      SNDPGMSG    MSG('KEYTBT=' *CAT &KEYTBT)
      SNDPGMSG    MSG('KEYUSR=' *CAT &KEYUSR)
      SNDPGMSG    MSG('DATFPC=' *CAT &DATFPC)
      SNDPGMSG    MSG('HORFPC=' *CAT &HORFPC)
      SNDPGMSG    MSG('DATFTR=' *CAT &DATFTR)
      SNDPGMSG    MSG('HORFTR=' *CAT &HORFTR)
      SNDPGMSG    MSG('DATRPC=' *CAT &DATRPC)
      SNDPGMSG    MSG('HORRPC=' *CAT &HORRPC)
      SNDPGMSG    MSG('DATRTR=' *CAT &DATRTR)
      SNDPGMSG    MSG('HORRTR=' *CAT &HORRTR)
      SNDPGMSG    MSG('SUPDEM=' *CAT &SUPDEM)
      SNDPGMSG    MSG('COMUSR=' *CAT &COMUSR)
      SNDPGMSG    MSG('ACKTBT=' *CAT &ACKTBT)
      SNDPGMSG    MSG('LIBTBT=' *CAT &LIBTBT)
      SNDPGMSG    MSG('OBJLIB=' *CAT &OBJLIB)
      SNDPGMSG    MSG('OBJFIL=' *CAT &OBJFIL)
      SNDPGMSG    MSG('OBJMBR=' *CAT &OBJMBR)
      SNDPGMSG    MSG('USRPRF=' *CAT &USRPRF)
      SNDPGMSG    MSG('NOMLOG=' *CAT &NOMLOG)
      SNDPGMSG    MSG('KEYEXT=' *CAT &KEYEXT)
      SNDPGMSG    MSG('IFSOBJ=' *CAT %SST(&IFSOBJ 1 1000))

```

```

/*****/
/* INSERER L'APPEL DE VOS TRAITEMENTS ICI */
/* Brancher obligatoirement en MESOK si OK */
/* Brancher obligatoirement en MESKO si erreur */
/* Brancher obligatoirement en MESPC si statut inconnu */
/*
/* R E M A R Q U E : Ce programme de consommation est une */
/* maquette commune pour le traitement : */
/* - des fichiers en entrée */
/* - des acquittements de transmission reçus. */
/* Cependant, DANS LE CAS DES ACQUITTEMENTS, il n'est pas */
/* nécessaire de brancher la suite du traitement sur les */
/* étiquettes MESOK et MESKO car la valorisation des champs */
/* KEYUSR, ACKTBT, LIBTBT est sans conséquence sur le menu */
/* "Supervision de l'historique". */
/*
/* CALL USERBIB(USERPGM) */
/* MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO)) */
/*
/*****/

/*****/
/* INSERT APPLICATION PROCESS HERE */
/* Mandatory GOTO label MESOK si OK */
/* Mandatory GOTO label MESKO si Error */
/*
/* CALL YOURLIB(YOURPROGRAM) */
/* MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO)) */
/*****/
CALL PGM(IPZAS2TRAK) PARM(&ACKTBT &LIBTBT &KEYUSR +
&COMUSR &OBJLIB &OBJFIL &OBJMBR &IFSOBJ +
&NOMLOG &MSGCMD)

IF COND(&ACKTBT *EQ 'PC') THEN(GOTO CMDLBL(MESPC))
IF COND(&ACKTBT *NE 'OK') THEN(GOTO CMDLBL(MESKO))

MESOK:
/* ***** */
/* Traitement OK */
/* ***** */
GOTO CMDLBL(MESFIN)

MESPC:
/* ***** */
/* Traitement PC */
/* ***** */
GOTO CMDLBL(MESFIN)

MESKO:
/* ***** */
/* Traitement KO */
/* ***** */
GOTO CMDLBL(MESFIN)

MESAB:
/* ***** */
/* Traitement en exception */
/* ***** */
CHGVAR VAR(&ACKTBT) VALUE('AB')
CHGVAR VAR(&LIBTBT) VALUE('Exception rencontrée +
dans le programme')
GOTO CMDLBL(MESFIN)

MESFIN: CHGVAR VAR(&MSGACK) VALUE('TBT/400 - Interface +
applicative - Code retour=' *CAT &ACKTBT +
*CAT ' : ' *BCAT &LIBTBT *BCAT '- Pour +
visualiser le job utiliser la commande : +
' *BCAT &MSGCMD)
SENDMSG MSG(&MSGACK) TOUSR(&USRPRF)
MONMSG MSGID(CPF0000)
SENDMSG MSG(&MSGACK) TOUSR(*SYSOPR)
MONMSG MSGID(CPF0000)
IPSRVCTBT FNCDEM(P) DEBDEM(N) FINDEM(C) EXCDEM(O) +

```

```

TRADEM(0) RTNCDP(&RTNCDP) KEYTBT(&KEYTBT) +
KEYUSR(&KEYUSR) ACKTBT(&ACKTBT) +
LIBTBT(&LIBTBT) SUPDEM(&SUPDEM) +
COMUSR(&COMUSR)
GOTO CMDLBL(ITER)

/*****
/* INCIDENT HORS ITERATION */
/*****
/* ERROR OUT OF LOOP */
/*****
CPF0000: CHGVAR VAR(&ACKTBT) VALUE('AB')
CHGVAR VAR(&LIBTBT) VALUE('Exception rencontrée +
dans le programme')
CHGVAR VAR(&MSGACK) VALUE('TBT/400 - Interface +
applicative - Code retour=' *CAT &ACKTBT +
*CAT ' : ' *BCAT &LIBTBT *BCAT '- Pour +
visualiser le job utiliser la commande : +
' *BCAT &MSGCMD)
SNDMSG MSG(&MSGACK) TOUSR(&USRPRF)
MONMSG MSGID(CPF0000)
SNDMSG MSG(&MSGACK) TOUSR(*SYSOPR)
MONMSG MSGID(CPF0000)
SNDPGMMSG MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
MSGDTA(&MSGACK) MSGTYPE(*ESCAPE)
MONMSG MSGID(CPF0000)
ENDPGM: ENDPGM

```

### 10.6.3. IPZAS2TRAK

Programme de traitement des acquittements entrants (appelé par « IPZAS2PCAK »).

Si ACKTBT est différent de ' ' (blanc) ou 'PC', le traitement est considéré en erreur.

Dans tous les cas un message est émis dans la message queue QSYSOPR.

La commande WRKJOB JOB(SAAS2APP) permet de visualiser le résultat d'exécution de ce programme.

```

/*-----*/
/*          TRAITEMENT DES ACQUITTEMENTS AS2          */
/*-----*/
PGM          PARM(&ACKTBT &LIBTBT &KEYUSR &COMUSR &OBJLIB +
               &OBJFIL &OBJMBR &IFSOBJ &NOMLOG &MSGCMD)

DCL          VAR(&ACKTBT) TYPE(*CHAR) LEN(2) /* Code +
           acquittement */
DCL          VAR(&LIBTBT) TYPE(*CHAR) LEN(128) /* Libellé +
           d'acheminement */
DCL          VAR(&KEYUSR) TYPE(*CHAR) LEN(16) /* Clé +
           utilisateur */
DCL          VAR(&COMUSR) TYPE(*CHAR) LEN(128) /* +
           Commentaire utilisateur */
DCL          VAR(&OBJLIB) TYPE(*CHAR) LEN(10) /* Nom de +
           bibliothèque */
DCL          VAR(&OBJFIL) TYPE(*CHAR) LEN(10) /* Nom de +
           fichier */
DCL          VAR(&OBJMBR) TYPE(*CHAR) LEN(10) /* Nom de +
           membre */
DCL          VAR(&IFSOBJ) TYPE(*CHAR) LEN(5000) /* Nom +
           d'objet si IFS */
DCL          VAR(&NOMLOG) TYPE(*CHAR) LEN(20) /* Nom du +
           correspondant */
DCL          VAR(&MSGCMD) TYPE(*CHAR) LEN(64) /* Commande +
           formatée par l'appelant */

/* ----- */
/* Valeur du champ ACKTBT considérée comme positive: */
/* - &ACKTBT = ' ' => OK (OK et PSR reçu ou non demandé) */
/* - &ACKTBT = 'PC' => Pris en compte (OK mais en attente du MDN) */
/* ----- */

IF          COND(&ACKTBT *EQ ' ') THEN(GOTO CMDLBL(FINOK))
IF          COND(&ACKTBT *EQ 'PC') THEN(GOTO CMDLBL(FINPC))

/* Autres tests... */

/* Sinon: Traitement en erreur... */
GOTO       CMDLBL(FINKO)

FINPC:     SNDPGMSG MSG('Émission AS2 OK (attente MDN) - NOMLOG: ' *TCAT +
               &NOMLOG *TCAT ' - KEYUSR: ' *TCAT +
               &KEYUSR) TOUSR(QSYSOPR)

GOTO       CMDLBL(FIN)

FINOK:     SNDPGMSG MSG('Émission AS2 OK - NOMLOG: ' *TCAT +
               &NOMLOG *TCAT ' - KEYUSR: ' *TCAT +
               &KEYUSR) TOUSR(QSYSOPR)

GOTO       CMDLBL(FIN)

FINKO:     SNDPGMSG MSG('*** Émission AS2 en erreur *** NOMLOG: +
               ' *TCAT &NOMLOG *TCAT ' - KEYUSR: ' *TCAT +
               &KEYUSR) TOUSR(QSYSOPR)
SNDPGMSG   MSG('ACKTBT: ' *CAT &ACKTBT) TOUSR(QSYSOPR)

GOTO       CMDLBL(FIN)

/* Autres traitements d'erreurs... */

FIN:       ENDPGM

```

## 10.6.4. IPZAS2PCMG

Programme de consommation lié à la file d'attente de type M et appelé par TBT400 pour chaque message reçu.

A noter que les champs « &ACKTBT » et « &LIBTBT » sont valorisés par le programme appelé (IPZAS2TRMG).

```

/*****
/* Ceci est le source du programme "dummy" de consommation      */
/* d'une file d'attente. Il est destiné à servir de modèle.     */
/* Ne remplacez pas le programme IPSPADUMMY dans la bibliothèque */
/* du progiciel (IPLSP). Une version plus complete est fournie  */
/* sous le nom IPSPADUMMC.                                     */
/*****
/* This is the source of the "IPSPADUMMY" program. It must be  */
/* used as a skeleton program and duplicated in customer library */
/* for modifications.                                         */
/*****

PGM
DCL      VAR(&DEBDEM) TYPE(*CHAR) LEN(1) VALUE(O)
DCL      VAR(&RTNCDP) TYPE(*DEC) LEN(11)
DCL      VAR(&KEYTBT) TYPE(*CHAR) LEN(16)
DCL      VAR(&KEYUSR) TYPE(*CHAR) LEN(16)
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(&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(&KEYEXT) TYPE(*CHAR) LEN(32)
DCL      VAR(&USRPRF) TYPE(*CHAR) LEN(16)
DCL      VAR(&IFSOBJ) TYPE(*CHAR) LEN(5000)

DCL      VAR(&JOB) TYPE(*CHAR) LEN(10)
DCL      VAR(&USER) TYPE(*CHAR) LEN(10)
DCL      VAR(&NBR) TYPE(*CHAR) LEN(6)
DCL      VAR(&MSGCMD) TYPE(*CHAR) LEN(64)
DCL      VAR(&MSGACK) TYPE(*CHAR) LEN(256)

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

RTVJOBA  JOB(&JOB) USER(&USER) NBR(&NBR)
CHGVAR   VAR(&MSGCMD) VALUE('WRKJOB JOB(' *TCAT &NBR +
      *TCAT '/' *TCAT &USER *TCAT '/' *TCAT +
      &JOB *TCAT ' '))

ITER:
/*****
/* APPEL DE LA COMMANDE DE RECEPTION                          */
/*****
/* CALL RECEIVE COMMAND                                       */
/*****

IPSRVTBT  FNCDEM(R) DEBDEM(&DEBDEM) FINDEM(C) +
      EXCDEM(N) TRADEM(0) RTNCDP(&RTNCDP) +
      KEYTBT(&KEYTBT) KEYUSR(&KEYUSR) +
      ACKTBT(&ACKTBT) LIBTBT(&LIBTBT) +
      OBJLIB(&OBJLIB) OBJFIL(&OBJFIL) +
      OBJMBR(&OBJMBR) USRPRF(&USRPRF) +
      DATFPC(&DATFPC) HORFPC(&HORFPC) +
      DATFTR(&DATFTR) HORFTR(&HORFTR) +
      DATRPC(&DATRPC) HORRPC(&HORRPC) +
      DATRTR(&DATRTR) HORRTR(&HORRTR) +
      SUPDEM(&SUPDEM) COMUSR(&COMUSR) +

```

```

NOMLOG(&NOMLOG) KEYEXT(&KEYEXT) /* Appel +
des API de TBT/400 via la Command +
IPSRCVTBT */

IF COND(&RTNCDP *NE 0) THEN(GOTO +
CMDLBL(ENDPGM)) /* Plus rien dans la file +
d'attente */

CHGVAR VAR(&DEBDEM) VALUE('N')

IPSSTP1 IFSOBJ(&IFSOBJ)

SNDPGMMSG MSG('KEYTBT=' *CAT &KEYTBT)
SNDPGMMSG MSG('KEYUSR=' *CAT &KEYUSR)
SNDPGMMSG MSG('DATFPC=' *CAT &DATFPC)
SNDPGMMSG MSG('HORFPC=' *CAT &HORFPC)
SNDPGMMSG MSG('DATFTR=' *CAT &DATFTR)
SNDPGMMSG MSG('HORFTR=' *CAT &HORFTR)
SNDPGMMSG MSG('DATRPC=' *CAT &DATRPC)
SNDPGMMSG MSG('HORRPC=' *CAT &HORRPC)
SNDPGMMSG MSG('DATRTR=' *CAT &DATRTR)
SNDPGMMSG MSG('HORRTR=' *CAT &HORRTR)
SNDPGMMSG MSG('SUPDEM=' *CAT &SUPDEM)
SNDPGMMSG MSG('COMUSR=' *CAT &COMUSR)
SNDPGMMSG MSG('ACKTBT=' *CAT &ACKTBT)
SNDPGMMSG MSG('LIBTBT=' *CAT &LIBTBT)
SNDPGMMSG MSG('OBJLIB=' *CAT &OBJLIB)
SNDPGMMSG MSG('OBJFIL=' *CAT &OBJFIL)
SNDPGMMSG MSG('OBJMBR=' *CAT &OBJMBR)
SNDPGMMSG MSG('USRPRF=' *CAT &USRPRF)
SNDPGMMSG MSG('NOMLOG=' *CAT &NOMLOG)
SNDPGMMSG MSG('KEYEXT=' *CAT &KEYEXT)
SNDPGMMSG MSG('IFSOBJ=' *CAT %SST(&IFSOBJ 1 1000))

/*****
/* INSERER L'APPEL DE VOS TRAITEMENTS ICI */
/* Brancher obligatoirement en MESOK si OK */
/* Brancher obligatoirement en MESKO si erreur */
/* Brancher obligatoirement en MESPC si statut inconnu */
/*
/* R E M A R Q U E : Ce programme de consommation est une */
/* maquette commune pour le traitement : */
/* - des fichiers en entrée */
/* - des acquittements de transmission reçus. */
/* Cependant, DANS LE CAS DES ACQUITTEMENTS, il n'est pas */
/* nécessaire de brancher la suite du traitement sur les */
/* étiquettes MESOK et MESKO car la valorisation des champs */
/* KEYUSR, ACKTBT, LIBTBT est sans conséquence sur le menu */
/* "Supervision de l'historique". */
/*
/* CALL USERBIB(USERPGM) */
/* MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO)) */
/*
*****/

/*****
/* INSERT APPLICATION PROCESS HERE */
/* Mandatory GOTO label MESOK si OK */
/* Mandatory GOTO label MESKO si Error */
/*
/* CALL YOURLIB(YOURPROGRAM) */
/* MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO)) */
*****/

CALL PGM(IPZAS2TRMG) PARM(&ACKTBT &LIBTBT &KEYUSR +
&COMUSR &OBJLIB &OBJFIL &OBJMBR &IFSOBJ +
&NOMLOG &MSGCMD)
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(MESAB))

IF COND(&ACKTBT *EQ 'PC') THEN(GOTO CMDLBL(MESPC))
IF COND(&ACKTBT *NE 'OK') THEN(GOTO CMDLBL(MESKO))

MESOK:
/* ***** */
/* Traitement OK */
/* ***** */
GOTO CMDLBL(MESFIN)

```

```

MESPC:
/* ***** */
/* Traitement PC */
/* ***** */
      GOTO      CMDLBL(MESFIN)

MESKO:
/* ***** */
/* Traitement KO */
/* ***** */
      GOTO      CMDLBL(MESFIN)

MESAB:
/* ***** */
/* Traitement en exception */
/* ***** */
      CHGVAR    VAR(&ACKTBT) VALUE('AB')
      CHGVAR    VAR(&LIBTBT) VALUE('Exception rencontrée +
dans le programme')
      GOTO      CMDLBL(MESFIN)

MESFIN:
      CHGVAR    VAR(&MSGACK) VALUE('TBT/400 - Interface +
applicative - Code retour=' *CAT &ACKTBT +
*CAT ' : ' *BCAT &LIBTBT *BCAT '- Pour +
visualiser le job utiliser la commande : +
' *BCAT &MSGCMD)
      SNDMSG    MSG(&MSGACK) TOUSR(&USRPRF)
      MONMSG    MSGID(CPF0000)
      SNDMSG    MSG(&MSGACK) TOUSR(*SYSOPR)
      MONMSG    MSGID(CPF0000)
      IPSRCVTBT FNCDEM(P) DEBDEM(N) FINDEM(C) EXCDEM(O) +
TRADEM(O) RTNCDP(&RTNCDP) KEYTBT(&KEYTBT) +
KEYUSR(&KEYUSR) ACKTBT(&ACKTBT) +
LIBTBT(&LIBTBT) SUPDEM(&SUPDEM) +
COMUSR(&COMUSR)
      GOTO      CMDLBL(ITER)

/*****
/* INCIDENT HORS ITERATION */
/*****
/* ERROR OUT OF LOOP */
/*****
CPF0000:
      CHGVAR    VAR(&ACKTBT) VALUE('AB')
      CHGVAR    VAR(&LIBTBT) VALUE('Exception rencontrée +
dans le programme')
      CHGVAR    VAR(&MSGACK) VALUE('TBT/400 - Interface +
applicative - Code retour=' *CAT &ACKTBT +
*CAT ' : ' *BCAT &LIBTBT *BCAT '- Pour +
visualiser le job utiliser la commande : +
' *BCAT &MSGCMD)
      SNDMSG    MSG(&MSGACK) TOUSR(&USRPRF)
      MONMSG    MSGID(CPF0000)
      SNDMSG    MSG(&MSGACK) TOUSR(*SYSOPR)
      MONMSG    MSGID(CPF0000)
      SNDPGMMSG MSGID(CPF9898) MSGF(QSYS/QCPFMSG) +
MSGDTA(&MSGACK) MSGTYPE(*ESCAPE)
      MONMSG    MSGID(CPF0000)
ENDPGM:      ENDPGM

```

## 10.6.5. IPZAS2TRMG

Programme de traitement des messages entrants (appelé par « IPZAS2PCMG »).

Le fichier reçu par TBT400 et identifié par les champs « &OBJLIB », « &OBJFIL » et « &OBJMBR » est copié dans QTEMP puis imprimé.



La valeur des champs « &ACKTBT » et « &LIBTBT » valorisés par ce programme seront directement visibles dans l'historique de TBT400. Cela signifie que **son exécution doit être totalement maîtrisée** sous peine de provoquer des erreurs de traitement ou d'interprétation de résultat lors de son exploitation dans l'historique.

La commande WRKJOB JOB(SMAS2APP) permet de visualiser le résultat d'exécution de ce programme.

```

/*-----*/
/*
/* Appelé automatiquement par TBT/400 pour chaque réception
/* (via la notion d'application)
/*
/* Les variables &ACKTBT &LIBTBT &KEYUSR &COMUSR
/* sont des zones de retour qui
/* seront affichées dans l'historique de TBT/400.
/*
/* &ACKTBT='OK' => OK + ligne en vert dans l'historique
/* &ACKTBT='KO' => Erreur + ligne en rouge dans l'historique
/*
/*-----*/

PGM PARM(&ACKTBT &LIBTBT &KEYUSR &COMUSR &OBJLIB +
&OBJFIL &OBJMBR &IFSOBJ &NOMLOG &MSGCMD)

DCL VAR(&ACKTBT) TYPE(*CHAR) LEN(2) /* Code +
acquittement - KO par défaut (zone de
retour) */
DCL VAR(&LIBTBT) TYPE(*CHAR) LEN(128) /* +
Libellé d'acheminement (zone de retour optionnelle) */
DCL VAR(&KEYUSR) TYPE(*CHAR) LEN(16) /* Clé +
utilisateur (zone de retour optionnelle) */
DCL VAR(&COMUSR) TYPE(*CHAR) LEN(128) /* +
Commentaire utilisateur (zone de retour +
optionnelle) */
DCL VAR(&OBJLIB) TYPE(*CHAR) LEN(10) /* Nom de +
bibliothèque */
DCL VAR(&OBJFIL) TYPE(*CHAR) LEN(10) /* Nom de +
fichier */
DCL VAR(&OBJMBR) TYPE(*CHAR) LEN(10) /* Nom de +
membre */
DCL VAR(&IFSOBJ) TYPE(*CHAR) LEN(5000) /* Nom +
d'objet si IFS */
DCL VAR(&NOMLOG) TYPE(*CHAR) LEN(20) /* Nom du +
correspondant */
DCL VAR(&MSGCMD) TYPE(*CHAR) LEN(64) /* Commande +
formatée par l'appelant */

/* En erreur par défaut */
CHGVAR VAR(&ACKTBT) VALUE('KO')
CHGVAR VAR(&LIBTBT) VALUE('Traitement en erreur - +
voir : ' *BCAT &MSGCMD)

CHGVAR VAR(&KEYUSR) VALUE('DELINS Exemple')
CHGVAR VAR(&COMUSR) VALUE('Commentaire envoyé par +
le programme d'application')

/* Le fichier reçu par TBT/400 est identifié par les champs:
/* - &OBJLIB: Bibliothèque de réception,
/* - &OBJFIL: Fichier de réception,
/* - &OBJMBR: Membre de réception.
/* - &IFSOBJ: Nom d'objet IFS
/*
/* Exemple de copie du fichier reçu par TBT/400 vers un fichier utilisé par
/* l'applicatif final (copie sans contrôle: *NOCHK - avec remplacement du
/* membre existant: *REPLACE et création du fichier si besoin: CRTFILE(*YES))*/
/*
CPYF FROMFILE(&OBJLIB/&OBJFIL) +
TOFILE(QTEMP/&OBJFIL) MBROPT(*REPLACE) +
CRTFILE(*YES) FMTOPT(*NOCHK)
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(ERREUR))

/*
/* Cet exemple se contente d'imprimer le fichier copié dans QTEMP au lieu
/* d'appeler un programme utilisateur...
/*
/* CALL PGM(PGMUTIL) PARM(QTEMP &OBJFIL) */
/*
CPYF FROMFILE(QTEMP/&OBJFIL) TOFILE(*PRINT)
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(ERREUR))

```

```

/* Autres traitements... */

/*
/* Si tout s'est passé correctement:
/* - &ACKTBT = 'OK'
/* - &LIBTBT = Libellé d'acheminement positif
/*
/* Si &ACKTBT n'est pas à 'OK' à la fin de ce programme, ce dernier
/* sera considéré comme étant en erreur par TBT/400 (ligne en rouge
/* dans l'historique).
/*
      CHGVAR      VAR(&ACKTBT) VALUE('OK')
      CHGVAR      VAR(&LIBTBT) VALUE('Traitement réalisé +
                          correctement')

      GOTO        CMDLBL(FIN)

/* En cas d'erreur de traitement:
/* - &ACKTBT = 'KO' (=> ligne en rouge dans l'historique TBT/400)
/* - &LIBTBT = Libellé d'acheminement négatif
/*
ERREUR: /* D'autres traitements d'erreur peuvent être ajoutés ici. */

      GOTO        CMDLBL(FIN)

FIN:      ENDPGM

```