

## Table des matières

1 Description de l'option SFTP de TBT/400.....	2
1.1 Concepts.....	2
1.2 Implémentation dans TBT/400.....	2
2 Installation.....	3
2.1 Pré-requis.....	3
2.2 Gestion des clés publiques/privées.....	3
2.2.1 Importer un certificat.....	3
2.2.2 Créer un certificat.....	3
2.2.3 Sauvegarder les Certificats.....	4
2.3 Table des Hosts.....	4
3 Initialisation d'une connexion SFTP.....	4
3.1 Considération TCP/IP.....	4
3.2 Correspondants par défaut.....	5
3.2.1 Modèles SFTP.....	5
3.2.2 IPL\$\$\$\$\$PROFIL.....	5
3.3 Création d'un correspondant SFTP.....	5
3.3.1 Détail d'un correspondant.....	6
3.3.2 Détail d'un correspondant SFTP.....	7
3.3.3 Détail des paramètres TCP/IP.....	8
3.3.4 Détail des certificats.....	9
3.3.5 Détail des paramètres d'accès.....	9
4 Négociation SSH.....	10
5 Transférer un fichier.....	10
5.1 Utilisation.....	10
5.2 Exemple.....	11
6 Commandes spécifiques.....	12
6.1 Commande d'envoi.....	12
6.2 Commande de statut.....	13
7 Exemple d'implémentation de la commande IPSNDSFTP.....	13
7.1 Mise en situation.....	13
7.2 Terminologie.....	13
7.3 Émission.....	14
7.4 Réception.....	14
7.5 Paramétrage des applications et files d'attente.....	14
7.6 Sources des programmes CL.....	15
7.6.1 SFTSEND.....	16
7.6.2 SFTRECV.....	18
7.6.3 SFTPCACK.....	20
7.6.4 SFTTRTACK.....	24
7.6.5 SFTPCMSG.....	25
7.6.6 SFTTRTMSG.....	29

## 1 Description de l'option SFTP de TBT/400

### 1.1 Concepts

Le protocole **SFTP** permet une communication sécurisée entre deux sites après établissement d'un tunnel **SSH**.

Le module SFTP de **TBT/400** est basé sur les standards suivants :

- SFTP version 3:
  - <https://filezilla-project.org/specs/draft-ietf-secsh-filexfer-02.txt>,
- SSH :
  - <http://www.ietf.org/rfc/rfc4251.txt>
  - <http://www.ietf.org/rfc/rfc4252.txt>
  - <http://www.ietf.org/rfc/rfc4253.txt>
  - <http://www.ietf.org/rfc/rfc4254.txt>

De nombreuses options de configuration sont disponibles afin de garantir le maximum de sécurité pendant un transfert SFTP :

- Les données circulant dans ce tunnel peuvent être :
  - chiffrées,
  - compressées,
  - authentifiées,
- Le client authentifie le serveur en validant sa signature,
- Le serveur peut authentifier le client par mot de passe ou par validation d'une signature.

### 1.2 Implémentation dans TBT/400

Le module **SFTP** de **TBT/400** propose depuis la version 610m90 les algorithmes suivants :

- algorithmes de signature :
  - RSA,
- algorithmes d'échange de clés :
  - diffie-hellman-group-exchange-sha1,
  - diffie-hellman-group-exchange-sha256
- algorithmes d'authentification HMAC :
  - hmac-sha2-256,
  - hmac-sha2-512,
  - hmac-sha1,
  - hmac\_sha1\_96,
  - hmac-md5,
  - hmac\_md5\_96,
- algorithmes de chiffrement :
  - aes256-cbc,
  - aes192-cbc,
  - aes128-cbc,
  - 3des-cbc,
- algorithmes de compression :
  - [zlib@openssh.com](mailto:zlib@openssh.com),
  - zlib.

## 2 Installation

### 2.1 Pré-requis

- L'OS doit être au niveau V5R3M0 ou supérieur,
- Les produits suivants doivent être installés :
  - Crypto Access Provider 128-bit AC3 (pour version V5R3M0 de l'OS).

### 2.2 Gestion des clés publiques/privées

En protocole SSH le client et le serveur peuvent s'identifier au moyens d'un jeu de clés publiques/privées RSA. C'est la méthode retenue par **TBT/400**.

Bien que seules les données propres aux clés publiques et privées soient utiles, **TBT/400** propose d'utiliser un certificat électronique X509 complet.

Deux raisons à ce choix :

- Cela n'a aucun impact sur l'implémentation du protocole,
- Cela permet de s'appuyer sur les fonctions de gestion de certificats intégrées dans **TBT/400** et largement utilisées par tous les autres protocoles (AS2, EBICS, OFTPv2, etc.).

#### 2.2.1 Importer un certificat

Il est possible d'importer un certificat existant dans le répertoire IFS de **TBT/400** en utilisant la commande suivante :

```
IPLSP/IPSCERTIFS CRTFNC (*INTDEL)
                    TYPRES ($$$$SFTP)
                    NOMLOG (EDIPSF'TUSRATOB)
                    CRTCTX (*LOCSSLSRV)
                    IFSOBJ ('/tmp/cert.p12')
                    PASSWO (PASSWORD)
```

Avec :

- CRTFNC : Code de fonction, ici « \*INTDEL » (ajout d'un certificat local),
- NOMLOG : Correspondant défini dans l'annuaire **TBT/400**,
- CRTCTX : Contexte d'utilisation du certificat, ici « \*LOCSSLSRV » (SSL/SSH),
- IFSOBJ : Chemin d'accès du certificat à importer,
- PASSWO : Mot de passe d'accès au fichier PKCS12.

Le code fonction « \*INTDEL » est utilisé ici pour ajouter un certificat, si un certificat de même nom existe, il ne sera pas écrasé.

Pour remplacer un certificat existant, le code fonction « \*INTREP » peut être utilisé.

#### 2.2.2 Créer un certificat

Il est possible de créer un certificat en utilisant la commande suivante :

```
IPLSP/IPSCRTCERT CERLOCA ('Paris')
                    CERSTAT ('Ile de France')
                    CERCOMN ('Client')
```

Une fois la commande exécutée correctement TBT/400 propose une vue de l'IFS où sont stockés les certificats (répertoire /IFSTBTIPSC par défaut) :

- IPSTBTSUBS\_APP.p12 Certificat SSL

Le nom du certificat reprend par défaut le nom du sous-système de TBT/400 suffixé par les caractères « \_APP ».

Ce nom peut être modifier (utilisation de plusieurs certificats), en renseignant le champ CERNAME (nom physique du certificat). Les autres champs de cette commande ne sont pas à modifier pour une installation standard (et pour un premier certificat).

### 2.2.3 Sauvegarder les Certificats

Deux répertoires doivent être sauvegardés :

- /QIBM/USERDATA/ICSS qui contient tout le paramétrage SSL OS/400
- /IfstbtIPSC qui contient les certificats privées SFTP, ainsi que les certificats publics des partenaires.

## 2.3 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.

## 3 Initialisation d'une connexion SFTP

### 3.1 Considération TCP/IP

Le protocole SFTP 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 seront dans 99% des cas connues 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 DE HOST DU SERVEUR'

La commande PING doit renvoyer :

- Verifying connection to xxx at address 111.222.333.444.

Xxx étant le nom d'hôte du serveur 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 :

- Unkonw host (où « Hôte inconnu » indiquant l'incapacité du client DNS à résoudre le nom d'hôte).

## **3.2 Correspondants par défaut**

### **3.2.1 Modèles SFTP**

**TBT/400** dispose dans son annuaire d'entrées modèles permettant d'effectuer rapidement vos premiers tests :

- EDIPSFTUSRATOB
- EDIPSFTUSRBTOA

Ces deux entrées peuvent être modifiées à volonté lors de vos différents tests mais ne doivent en aucun cas être utilisés en production : elles seront automatiquement recréer avec leur valeur par défaut à chaque mise à jour de **TBT/400**.

### **3.2.2 IPLS\$\$\$\$PROFIL**

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

En particulier :

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

## **3.3 Création d'un correspondant SFTP**

Pour créer un nouveau correspondant, depuis une ligne de commande :

- IPLSP/IPS
- « Gestion de l'annuaire »,
- « Définition des correspondants »,
- Surchargez le nom logique du correspondant à dupliquer puis validez,
- F10 sur le nouveau correspondant.

### 3.3.1 Détail d'un correspondant

DANG 9941 Devt	<b>Détail d'un correspondant</b>				IPLS08	IPLSD
Type d'annuaire . . . .	\$\$\$\$SFTP	F4	Portée . . . .	*GLOBAL		
Nom du correspondant .	EDIPSFTUSRATOB		Type réseau .	\$\$\$\$SFTP	F4	
Libellé correspondant .	TBT400 Mta 150605090234					
Commentaire utilisateur	Crea &KEYTBT			&KEYL16		
&CURJOB		&ipsdatz				
Auteur . . . . .	Crea &ipsdatz					
Objet . . . . .	Crea &KEYTBT			&KEYL16		
&CURJOB		&ipsdatz		&ipsdatc		
		Suffixe N O,N	Trace	O,N		
A l'attention de . . .	Crea &ipsdatz		Impre.	O,N,C,B		
Référence du message .	File.&CPTUS09		Scrut.	O,N		
Emission mode puits . .	O,N	Messages demandés . . .		O,N,C,B		
Accusé demandé . . . .	O,N,C	Avis =====> Distri	O Lectur	Applic A		
Mode transparent . . .	O,N	Ajout CR/LF . . . .	O,N	Lrec		
Suppression des blancs	O,N,L	Transfert ASCII . . . .	Ccsi	65533		
Priorité réseau . . . .	N,U,H	Enreg. par segment . .	0 -	255		
R. txt Lr	Tr	Ty	C	Ec	R. bin Lr	Tr Ty C Ec
Identifiant réseau . .					Ha	C S Cm
F1=Hlp F3=Exi F6=Imp F7=Avn F8=Apr F9=Cmd F13=Hau F19=Gau F20=Dro F21=Def						
F24=Bas						

Les champs importants dans cet écran sont :

- Nom du correspondant,
- Transfert ASCII (émission ASCII/EBCDIC) = **ASCDEM (associé au CCSID)**,
- Les champs de la ligne 20 permettent de préciser un mode de réception (longueur d'enregistrements, gestion CR/LF, etc.).

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

### 3.3.2 Détail d'un correspondant SFTP

```

DSFT  9949 Devt          Détail d'un correspondant SFTP          IPLS08      IPLSD
Type d'annuaire . . . . $$$SFTP          Portée . . . . *GLOBAL
Nom du correspondant . EDIPSFTUSRATOB      Type réseau . $$$SFTP
Libellé correspondant . TBT400 Mta 150605090234
Client  User . . . . . EDIPSFTUSRA          Suplec N Ccsid      1252
      Password . . . . . PSW
Serveur User . . . . . EDIPSFTUSRB          Ccsid      1252
      Password . . . . . PSW
Dir / Emis .
Dsn / Emis .  FIC.&CPTUS09.&NOMFIL

Dsn / New . .
Dir / Scrut .
Dsn / Scrut .
Exchange . .
Hmac . . . .
Chiffrement .
Compression .
Exchange Max
Sélection d'application  A,D          Application par défaut  IPLSDemo 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

```

Renseigner les champs suivants :

- Client User : Nom du client SFTP tel que déclaré sur le serveur,
- Suplec : Suppression du fichier distant après lecture,
- Ccsid : Code page des commandes SFTP en mode client,
- Password : Mot de passe du client SFTP tel que déclaré sur le serveur,
- Serveur User : Nom du serveur SFTP à remettre au client,
- Ccsid : Code page des commandes SFTP en mode serveur,
- Password : Mot de passe du serveur SFTP à remettre au client,
- Dir / Emis : Répertoire distant pour émission,
- Dsn / Emis : Nom de fichier distant pour émission,
- Dsn / New : Nom de fichier distant pour renommage après émission,
- Dir / Scrut : Répertoire distant pour scrutation,
- Dsn / Scrut : Nom de fichier distant pour scrutation,
- Exchange : Algorithmes d'échange de clés dans l'ordre de préférence,
- Hmac : Algorithmes de validation HMAC dans l'ordre de préférence,
- Chiffrement : Algorithmes de chiffrement dans l'ordre de préférence,
- Compression : Algorithmes de compression dans l'ordre de préférence,
- Exchange Max : Quantité max de données échangées avant un renouvellement de clés,
- Sélection d'application : Choix de l'application de réception,
- Application par défaut : Application par défaut utilisée en réception.

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

### 3.3.3 Détail des paramètres TCP/IP

```

DTCP 9947 Devt      Détail des paramètres TCP/IP      IPLS08      IPLSD
Type d'annuaire . . . . $$$$SFTP      Portée . . . . *GLOBAL
Nom du correspondant . EDIPSFTUSRATOB      Type réseau . $$$$SFTP
Libellé correspondant . TBT400 Mta 150605090234

Hostname IP distant . . AS400D.ipls.local

Adresse IP distant . . 10:2:3:134

Port      IP distant . .

Hostname IP local . . . AS400D.ipls.local

Adresse IP locale . . . 10:2:3:34

Usage adresse . . . . .

Utilisation Ssl . . . . N Option      Protocole      Cipher      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

```

L'utilisation du mode SSL n'a aucun sens en protocole SFTP:

- Utilisation Ssl : N.

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



### 3.3.4 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.

DCRT	9973	Devt	<b>Détail des certificats</b>	IPLS08	IPLSD
Type d'annuaire . . . .	\$\$\$\$SFTP			Portée . . . .	*GLOBAL
Nom du correspondant .	EDIPSFTUSRATOB			Type réseau .	\$\$\$\$SFTP
Libellé correspondant .	TBT400 Mta 150605090234				
Certificat local Ssl .					K
Certificat remote Ssl .					K
Certificat local Aut .					K
Certificat remote Aut .					K
Certificat local Sig .	*S				K
Certificat remote Sig .	*S				K
Certificat local Cry .	*S				K
Certificat remote Cry .	*S				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 protocole SFTP les certificats utilisés sont :

- Certificat local Ssl :
- Certificat remote Ssl :
- Certificat local Sig :
- Certificat remote Sig :

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

### 3.3.5 Détail des paramètres d'accès

Dans cet écran vous pouvez affiner le paramétrage du mode serveur SFTP.

Appuyer sur F21 pour afficher les valeurs par défaut.

```

DSRV  9951 Devt      Détail des paramètres d'accès      IPLS08      IPLSD
Type d'annuaire . . . . $$$$SFTP      Portée . . . . *GLOBAL
Nom du correspondant . EDIPSFTUSRATOB      Type réseau . $$$$SFTP
Libellé correspondant . TBT400 Mta 150605090234      Protocole T T
Mot de passe d'accès . PSW      Ctrl. appelant O,N
Mot de passe ancien . .

Application par défaut IPLSDEMO
Sous adresse X25 admise
Numéro appelant admis .

Tad autorisée . . . . . O,N
Contrôle host . . . . . O O,N
Ssl obligatoire . . . . Option Protocole Ciper
Adresse IP .

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

```

Le paramétrage du correspondant est terminé.

## 4 Négociation SSH

En protocole SSH le choix des algorithmes utilisés est négocié entre le client et le serveur.

A l'initialisation d'une connexion, la négociation est à l'initiative du client, qui propose sa liste de possibilités en fonction de ses préférences. Le serveur répond par sa propre liste et les deux parties s'alignent sur les valeurs communes lorsque cela est possible.

Si la négociation échoue, la communication est avortée.

Au cours d'une connexion le client et le serveur peuvent à tout instant imposer une nouvelle négociation.

Dans **TBT/400**, le champs « Exchange Max » du correspondant permet d'imposer une quantité maximale de données transférée au delà de laquelle une renégociation est imposée.

**TBT/400** permet de bâtir la liste des algorithmes proposés de deux façons :

- au niveau local : en modifiant le correspondant cible,
- au niveau global : en modifiant le correspondant modèle IPLS\$\$\$PROFIL.

La modification au niveau global impacte les valeurs par défaut de tous les correspondants SFTP: ceux pour lesquels rien n'est précisé se verront donc affectés les choix fixés dans IPLS\$\$\$PROFIL.

## 5 Transférer un fichier

### 5.1 Utilisation

Il est possible de transférer un fichier :

- Par menu,
- Par commande.

Le mode classique dans **TBT/400** est l'utilisation de la commande d'émission IPSNDSFTP.

Plusieurs champs sont à initialiser pour l'usage de cette commande :

- Les qualificants du fichier à envoyer
  - OBJLIB, OBJFIL, OBJMBR pour un fichier natif OS/400,
  - OBJFIL(\*IFS), IFSDIR, IFSOBJ pour un fichier IFS,
- Le correspondant distant (NOMLOG),
- Les champs FTPDIR et FTPDSN permettant de préciser le nom du répertoire distant et du fichier distant pour une émission,
- Les champs FTPDIS et FTPDSS permettant de préciser le nom du répertoire distant et du fichier distant pour une scrutation.

## 5.2 Exemple

Le certificat du correspondant serveur EDIPSFTUSRBTOA doit être disponible avant de pouvoir exécuter ces deux exemples.

Exemple d'une émission de fichier SFTP vers le correspondant EDIPSFTUSRATOB :

```
IPLSP/IPSNDSFTP NOMLOG (EDIPSFTUSRATOB)
                  KEYUSR ('clé utilisateur')
                  OBJFIL (IPSSAMPLES)
                  OBJLIB (IPLSP)
                  OBJMBR (IPZIGBAN)
                  COMUSR ('commentaire')
```

Exemple d'une scrutation de fichier SFTP depuis le correspondant EDIPSFTUSRATOB :

```
IPLSP/IPSNDSFTP NOMLOG (EDIPSFTUSRATOB)
                  KEYUSR ('clé utilisateur')
                  OBJFIL (*DUMMY)
                  OBJLIB (*DUMMY)
                  OBJMBR (*DUMMY )
                  COMUSR ('commentaire')
                  FTPDIS (/dir1/dir2)
                  FTPDSS (file)
```

## 6 Commandes spécifiques

### 6.1 Commande d'envoi

#### Emission SFTP (IPSNDFTP)

```

Nom logique du correspondant . . . . . NOMLOG
Clé utilisateur . . . . . KEYUSR
Objet à traiter: Fichier . . . . . OBJFIL
Objet à traiter: Bibliothèque . . . . . OBJLIB
Objet à traiter: Membre . . . . . OBJMBR
Nom du spool à envoyer . . . . . SPLNAM
Travail ayant créé le Spool . . . . . SPLJOB
  Utilisateur . . . . .
  Numéro . . . . .
Numéro du spool à envoyer . . . . . SPLNUM
Suppression du spoolfile . . . . . SPLSUP
IFS - Répertoire . . . . . IFSDIR
IFS - Objet . . . . . IFSOBJ
Auteur du courrier . . . . . AUTHOR
Objet du courrier . . . . . OBJECT
A l'attention de . . . . . ATTENT
Référence du courrier . . . . . REFMSG
Format fichier . . . . . FMTFIC
Taille fichier . . . . . SIZFIC
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 . . . . . HAUPAG
Environnement demandé . . . . . SETENV
Code utilisateur souhaité . . . . . USRDEM
Mode synchrone . . . . . SYNDEM
Synchrone : Nbr Essais . . . . . SNBRRT
Synchrone : Intervalle Essai . . . . . SLAPSE
Répertoire d'arrivée . . . . . FTPDIR
Fichier d'arrivée . . . . . FTPDSN
Fichier final . . . . . FTPNDS
Répertoire à scruter . . . . . FTPDIS
Fichier scruté . . . . . FTPDSS
Suppression sur serveur distant . . . . . FTPSUP
Profil de transmission . . . . . SNDPRO
Ajout de CR/LF . . . . . CRLDEM
Ajout de CR/LF fin . . . . . CRLFEN
Suppression blancs . . . . . SPADEM
Traduction ASCII . . . . . ASCDEM
Ccsid demandé . . . . . CCSID
Enregistrements / segment . . . . . RECSEG

```

Application de réception . . . . APPREC

## 6.2 Commande de statut

Statut SFTP (IPSSTSFTP)

```
Répertoire d'arrivée . . . . . FTPDIR
Répertoire à scruter . . . . . FTPDIS
Fichier d'arrivée . . . . . FTPDSN
Fichier scruté . . . . . FTPDSS
Suppression sur serveur distan FTPSUP
Nom de fichier sur serveur dis FTPDSR
Nom de système du serveur . . . FTPSYS
Welcome du serveur . . . . . FTPWEL
Commande à exécuter . . . . . FTPCMD
Répertoire final . . . . . FTPNDI
Fichier final . . . . . FTPNDS
Fonction fin demandée . . FINDEM
Fonction exception demandée . . EXCDEM
Fonction trace demandée . . TRADEM
Code retour (Num. étendu) . . . RTNCDP
Libellé du compte rendu . . . . MSGTXT
```

## 7 Exemple d'implémentation de la commande IPSNDSFTP

### 7.1 Mise en situation

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

- Envoi d'un fichier vers un serveur distant,
- Scrutation d'un répertoire sur un serveur distant.

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

**Attention : Ces programmes ne sont que des exemples dont l'objectif principal est d'illustrer cette documentation. Il ne sont absolument pas destinés à être déployés en production en l'état.**

### 7.2 Terminologie

TBT/400 utilise les notions d'application, de files d'attente et de programmes de consommation.

- SFTPAPP : Application (au sens TBT/400),
- ASFTPAPP : File d'attente des acquittements,
- SFTPCACK : Programme de consommation des acquittements reçus (lie TBT/400 à SFTTRTACK),
- SFTTRTACK : Nom du CL de traitements des acquittements reçus (programme utilisateur),
- MSFTPAPP : File d'attente des messages,
- SFTPCMSG : Programme de consommation des messages reçus (lie TBT/400 à SFTTRTMSG),
- SFTTRTMSG : Nom du CL de traitements des messages reçus.

### 7.3 Émission

Déroulement du processus d'émission :

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

### 7.4 Réception

Déroulement du processus de réception :

- Phase n°1 : Le programme utilisateur SFTRECV appelle **TBT/400**
  - Soumission de la commande **utilisant l'application émettrice « SFTPAPP »**,
  - Traitement des codes retour (de la soumission),
  - Fin du programme SFTRECV avec messages d'erreurs si besoin.
- Phase n°2 : **TBT/400** appelle le programme SFTTRTACK (traitement acquittements **TBT/400**) pour chaque soumission effectuée
  - Test des codes retours,
  - Traitement des erreurs éventuelles,
  - Fin du programme SFTTRTACK avec messages d'erreurs si besoin.
- Phase n°3 : **TBT/400** appelle le programme SFTTRTMSG 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 SFTTRTMSG avec messages d'erreurs si besoin.

### 7.5 Paramétrage des applications et files d'attente

Pour créer l'application SFTPAPP :

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

Les valeurs par défaut sont suffisantes pour notre exemple.

Pour créer les files d'attentes ASFTPAPP et MSFTPAPP :

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

Idem pour la file d'attente ASFTPAPP en utilisant cette fois le modèle MSFTPAPP et le programme SFTPCACK.

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 MSFTPAPP le programme de consommation SFTPCMSG sera appelé avec toutes les variables nécessaire au traitement applicatif.

Il en va de même pour les acquittements.

## **7.6 Sources des programmes CL**

A noter : la bibliothèque des programmes IPLSP (par défaut) doit être en ligne pour compiler ces sources.

## 7.6.1 SFTSEND

```

/*-----*/
/*                                          */
/* Déroulement du CL:                      */
/* - Appel IPSNDSFTP en mode émission,      */
/* - Gestion du code retour de la commande */
/*                                          */
/*-----*/

      PGM          PARM(&OBJLIB &OBJFIL &OBJMBR)

      DCL          VAR(&OBJLIB) TYPE(*CHAR) LEN(10)
      DCL          VAR(&OBJFIL) TYPE(*CHAR) LEN(10)
      DCL          VAR(&OBJMBR) TYPE(*CHAR) LEN(10)
      DCL          VAR(&KEYTBT)  TYPE(*CHAR) LEN(16)
      DCL          VAR(&KEYUSR)  TYPE(*CHAR) LEN(16)
      DCL          VAR(&RTNCDP) TYPE(*DEC) LEN(11) VALUE(8) /* +
      8 = KO (par défaut) */
      DCL          VAR(&MSGTXT)  TYPE(*CHAR) LEN(256)

/*
/* La commande IPSNDSFTP soumet la demande d'émission de */
/* fichiers et IPSRCVTBT permet d'en contrôler l'exécution. */
/* Une autre solution consiste à positionner EXCDEM à OUI */
/* (IPSNDSFTP) pour demander la génération d'une exception */
/* en cas d'erreur et utiliser un MONMSG IPS0000 (à la place */
/* de IPSRCVTBT + test code retour). */
/*
/*
/* FINDEM(N)      : Ne pas clôturer l'environnement */
/* EXCDEM(N)      : Pas d'exception en cas d'erreurs */
/* APPEME(SFTPAPP): Les accusés et messages reçus seront */
/*                  envoyés à l'application SFTPAPP */
/* ACKDEM(O)      : Surveillance des accusés active */
/*                  (= appel l'application SFTPAPP à la */
/*                  réception d'un accusé) */
/*
/*
/* OBJLIB, OBJFIL et OBJMBR sont alimentés par l'applicatif */
/* appelant TBT/400. */
/*
/*
      IPSNDSFTP  NOMLOG(EDIPSFTUSRATOB) OBJFIL(&OBJFIL) +
                  OBJLIB(&OBJLIB) OBJMBR(&OBJMBR) +
                  FINDEM(N) +
                  EXCDEM(N) APPEME(SFTPAPP) ACKDEM('O')
      MONMSG     MSGID(CPF0000)

/* 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 IPSNDSFTP) */
/* 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... */
  SNDFGMMMSG MSG('Appel TBT/400 pour émission OK')
  GOTO      CMDLBL(FIN)

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

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

FIN:      ENDFGM
```

## 7.6.2 SFTRECV

```

/*-----*/
/*                                          */
/*  Déroulement du CL:                    */
/*  - Appel IPSNDSFTP en mode scrutation (récupération),      */
/*  - Gestion du code retour de la commande                    */
/*                                          */
/*-----*/

      PGM

      DCL      VAR(&KEYTBT)   TYPE(*CHAR) LEN(16)
      DCL      VAR(&KEYUSR)   TYPE(*CHAR) LEN(16)
      DCL      VAR(&RTNCDP)  TYPE(*DEC)  LEN(11) VALUE(8) /* +
                          8 = KO (par défaut) */
      DCL      VAR(&MSGTXT)   TYPE(*CHAR) LEN(256)

/*  La commande IPSNDSFTP soumet la demande de réception de  */
/*  fichiers et IPSRCVTBT permet d'en contrôler l'exécution. */
/*  Une autre solution consiste à positionner EXCDEM à OUI    */
/*  (IPSNDSFTP) pour demander la génération d'une exception  */
/*  en cas d'erreur et utiliser un MONMSG IPS0000 (à la place */
/*  de IPSRCVTBT + test code retour).                          */
/*                                          */
/*  FINDEM(N)          : Ne pas clôturer l'environnement     */
/*  EXCDEM(N)          : Pas d'exception en cas d'erreurs     */
/*  APPEME(SFTPAPP): Les accusés et messages reçus seront    */
/*                   envoyés à l'application SFTPAPP         */
/*  ACKDEM(O)          : Surveillance des accusés active      */
/*                   (= appel l'application SFTPAPP à la     */
/*                   réception d'un accusé)                   */
/*                                          */
/*  OBJLIB, OBJFIL et OBJMBR valent *DUMMY dans le cas d'une  */
/*  scrutation (*DUMMY = fichier "fictif").                    */
/*                                          */
/*                                          */
      IPSNDSFTP  NOMLOG(EDIPSFTUSRATOB) OBJFIL(*DUMMY) +
                OBJLIB(*DUMMY) OBJMBR(*DUMMY) +
                FINDEM(N) EXCDEM(N) APPEME(SFTPAPP) +
                ACKDEM('O')
      MONMSG     MSGID(CPF0000)

/*  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 IPSNDSFTP)                */
/*  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)

                COND(&RTNCDP *NE 0) THEN(GOTO CMDLBL(ERREUR))

IF

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

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

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

FIN:           ENDPGM
```

### 7.6.3 SFTPCACK

```

/*****
/* 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 (0)
      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 (&OBJJMBR)  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 (&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 +
IPSRVBT */

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

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

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)

/*****
/* 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        */
/* Mandatory GOTO label      MESPS si Unknown state */
/*                                                    */
/* CALL YOURLIB (YOURPROGRAM)              */
/* MONMSG      MSGID(CPF0000) EXEC (GOTO CMDLBL (MESKO)          */
/*****/
                CALL      PGM(BM/SFTTRTACK) PARM(&ACKTBT &LIBTBT &NOMLOG)
                MONMSG    MSGID(CPF0000) EXEC (GOTO CMDLBL (MESKO))

MESOK:          CHGVAR      VAR(&KEYUSR) VALUE ('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE ('Commentaire envoyé par +
                le programme d'application')
                CHGVAR      VAR(&ACKTBT) VALUE ('OK')
                CHGVAR      VAR(&LIBTBT) VALUE ('Message consommé avec +
                succès')
/*            CHGVAR      VAR(&SUPDEM) VALUE ('N')      Override valeur +
                initiale */
                GOTO        CMDLBL (MESFIN)

MESPC:          CHGVAR      VAR(&KEYUSR) VALUE ('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE ('Commentaire envoyé par +
                le programme d'application')
                CHGVAR      VAR(&ACKTBT) VALUE ('PC')
                CHGVAR      VAR(&LIBTBT) VALUE ('Message pris en compte')
/*            CHGVAR      VAR(&SUPDEM) VALUE ('N')      Override valeur +
                initiale */
                GOTO        CMDLBL (MESFIN)

MESKO:          CHGVAR      VAR(&KEYUSR) VALUE ('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE ('Commentaire envoye par +
                le programme d'application')
                CHGVAR      VAR(&ACKTBT) VALUE ('KO')
                CHGVAR      VAR(&LIBTBT) VALUE ('Message en erreur')
/*            CHGVAR      VAR(&SUPDEM) VALUE ('N')      Override valeur +
                initiale */
                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
```

### 7.6.4 SFTTRTACK

```

/*-----*/
/*          TRAITEMENT DES ACQUITTEMENTS SFTP          */
/*-----*/

      PGM          PARM(&ACKTBT &LIBTBT &NOMLOG)

      DCL          VAR(&ACKTBT) TYPE(*CHAR) LEN(2)
      DCL          VAR(&LIBTBT) TYPE(*CHAR) LEN(128)
      DCL          VAR(&NOMLOG) TYPE(*CHAR) LEN(20)

/* ----- */
/* 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 PSR) */
/* - &ACKTBT = 'BV' => Boite vide (OK mais rien à recevoir) */
/* ----- */

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

/* Autres tests... */

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

FINOK:      SNDPGMSG MSG('Traitement SFTP OK') TOUSR(QSYSOPR)
           SNDPGMSG MSG('NOMLOG: ' *CAT &NOMLOG) TOUSR(QSYSOPR)

           GOTO          CMDLBL(FIN)

FINKO:      SNDPGMSG MSG('Traitement SFTP en erreur...') TOUSR(QSYSOPR)
           SNDPGMSG MSG('NOMLOG: ' *CAT &NOMLOG) TOUSR(QSYSOPR)
           SNDPGMSG MSG('ACKTBT: ' *CAT &ACKTBT) TOUSR(QSYSOPR)
           SNDPGMSG MSG('LIBTBT: ' *CAT &LIBTBT) TOUSR(QSYSOPR)

           GOTO          CMDLBL(FIN)

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

FIN:        ENDPGM

```



### 7.6.5 SFTPCMSG

```

/*****
/* 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(0)
      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(&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')

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

/*****
/* 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                    */
/* Mandatory GOTO label      MESPS si Unknown state           */
/*                                                     */
/* CALL YOURLIB(YOURPROGRAM)                                    */
/* MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO))          */
/*****/
                CALL      PGM(BM/SFTTRTMSG) PARM(&ACKTBT &LIBTBT +
                &OBJLIB &OBJFIL &OBJMBR &MSGCMD)
MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(MESKO))

                IF      COND(&ACKTBT *NE ' ') THEN(GOTO CMDLBL(MESKO))

MESOK:      CHGVAR      VAR(&KEYUSR) VALUE('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE('Commentaire envoyé par +
                le programme d''application')
/*          CHGVAR      VAR(&ACKTBT) VALUE('OK')                */
/*          CHGVAR      VAR(&LIBTBT) VALUE('Message consommé avec + */
/*          succès')                                           */
/*          CHGVAR      VAR(&SUPDEM) VALUE('N')      Override valeur +
                initiale */
                GOTO      CMDLBL(MESFIN)

MESPC:      CHGVAR      VAR(&KEYUSR) VALUE('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE('Commentaire envoyé par +
                le programme d''application')
                CHGVAR      VAR(&ACKTBT) VALUE('PC')
                CHGVAR      VAR(&LIBTBT) VALUE('Message pris en compte')
/*          CHGVAR      VAR(&SUPDEM) VALUE('N')      Override valeur +
                initiale */
                GOTO      CMDLBL(MESFIN)

MESKO:      CHGVAR      VAR(&KEYUSR) VALUE('Userkey')
                CHGVAR      VAR(&COMUSR) VALUE('Commentaire envoye par +
                le programme d''application')
/*          CHGVAR      VAR(&ACKTBT) VALUE('KO')                */
/*          CHGVAR      VAR(&LIBTBT) VALUE('Message en erreur')   */
/*          CHGVAR      VAR(&SUPDEM) VALUE('N')      Override valeur +
                initiale */
                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(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

```

### 7.6.6 SFTTRMSG

```

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

PGM PARM(&ACKTBT &LIBTBT &OBJLIB &OBJFIL &OBJMBR &MSGCMD)

DCL VAR(&OBJLIB) TYPE(*CHAR) LEN(10)
DCL VAR(&OBJFIL) TYPE(*CHAR) LEN(10)
DCL VAR(&OBJMBR) TYPE(*CHAR) LEN(10)
DCL VAR(&ACKTBT) TYPE(*CHAR) LEN(2) VALUE('KO') +
/* Code retour - KO par défaut (zone de +
retour) */
DCL VAR(&LIBTBT) TYPE(*CHAR) LEN(128) /* +
Libellé d'acheminement (zone de retour) */
DCL VAR(&MSGCMD) TYPE(*CHAR) LEN(64)

/* Le fichier reçu par TBT/400 est identifié par les champs:
*/
/* - &OBJLIB: Bibliothèque de réception,
*/
/* - &OBJFIL: Fichier de réception,
*/
/* - &OBJMBR: Membre de réception.
*/
/*
*/
/* Exemple de copie du fichier reçu par TBT/400 vers un fichier utilisé par
*/
/* l'applicatif final (copie sans contrôle - *NOCHK - et avec remplacement du
*/
/* membre existant - *REPLACE).
*/
*/
*/

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(PGMCOMPTA) 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 = ' '
*/
/* - &LIBTBT = Libellé d'acheminement positif
*/
/*
*/
/* Si &ACKTBT n'est pas à ' ' 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(' ')
          CHGVAR          VAR(&LIBTBT) VALUE('Traitement réalisée +
                          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:    CHGVAR          VAR(&ACKTBT) VALUE('KO')
          CHGVAR          VAR(&LIBTBT) VALUE('Traitement en erreur - +
                          voir : ' *BCAT &MSGCMD)
          GOTO            CMDLBL(FIN)

FIN:      ENDPGM

```