TBT400 - GUIDE SFTP

IPLS

Version 4.60, 04/02/2022

Notice

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.

Table des matières

1. Gestion du document

Version	Commentaires	Statut	Auteurs	Date
0.1	Initialisation	Initié	Philippe CASSARD	12/04/2022

2. Description de l'option SFTP de TBT/400

2.1. 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 circulants 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.

2.2. Implémentation dans TBT/400

Le module **SFTP** de **TBT/400** propose depuis la version 611m17 les algorithmes suivants :

- algorithmes de signature :
  - RSA,
- algorithmes d'échange de clés :
  - diffie-hellman-group-exchange-sha1,
  - diffie-hellman-group-exchange-sha256,
  - ecdh-sha2-nistp192 (SHA256),
  - ecdh-sha2-nistp224 (SHA256),
  - ecdh-sha2-nistp256 (SHA256),
  - ecdh-sha2-nistp384 (SHA384),
  - ecdh-sha2-nistp521 (SHA512)
- algorithmes d'authentification HMAC :
  - hmac-sha2-256,
  - hmac-sha2-512,
  - hmac-sha1,
  - hmac_sha1_96,
  - hmac-md5,
  - hmac_md5_96,

- algorithmes de chiffrement :
    - aes256-ctr,
    - aes192-ctr,
    - aes128-ctr,
    - aes256-cbc,
    - aes192-cbc,
    - aes128-cbc,
    - 3des-cbc,
- algorithmes de compression :
    - zlib@openssh.com,
    - zlib.

Les derniers algorithmes d'échanges de clés ecdh-sha2-nistpxxx de type Elliptic Curve Diffie-Hellman sont disponibles à partir de la version V7R2 de l'OS.

3. Installation

3.1. 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). ???

3.2. Gestion des clés publiques/privées

En protocole SSH le client et le serveur peuvent s'identifier au moyen 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.).

3.2.1. 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) ①
                 TYPRES($$$$SFTP)
                 NOMLOG(EDIPSFTUSRATOB) ②
                 CRTCTX(*LOCSSLSRV) ③
                 IFSOBJ('/tmp/cert.p12') ④
                 PASSWO(PASSWORD) ⑤
```

① CRTFNC : Code de fonction, ici « *INTDEL » (ajout d'un certificat local)

② NOMLOG : Correspondant défini dans l'annuaire **TBT/400**

③ CRTCTX : Contexte d'utilisation du certificat, ici « *LOCSSLSRV » (SSL/SSH)

④ IFSOBJ : Chemin d'accès du certificat à importer

⑤ 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.

3.2.2. 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).

3.2.3. 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.

3.3. 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.

4. Initialisation d'une connexion SFTP

4.1. 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. ① ②
```

① xxx : nom d'hôte du serveur

② 111.222.333.444 : 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 ①
```

① « Hôte inconnu » indique l'incapacité du client DNS à résoudre le nom d'hôte.

4.2. Correspondants par défaut

4.2.1. Modèles SFTP

TBT/400 dispose dans son annuaire d'entrées des 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 recréées avec leur valeur par défaut à chaque mise à jour de **TBT/400**.

4.2.2. 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.

Ne pas supprimer le correspondant *IPLS$$$$PROFIL*

4.3. 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.

4.3.1. Détail d'un correspondant

```
DANG   9941 Nrcu         Détail d'un correspondant          IPLSPC     IPLSD
 Type d'annuaire . . . . $$$$SFTP   F4              Portée . . . . *GLOBAL
 Nom du correspondant  . EDIPSFTUSRATOB            Type réseau  . $$$$SFTP F4 ①
 Libellé correspondant . TBT400 Mta 200708110801         Inactif   N
 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 ②
 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 0 C 0 S 0 Cm 0
 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 :

① *Nom du correspondant*

② *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

4.3.2. Détail d'un correspondant SFTP

```
DSFT   9949 Nrcu       Détail d'un correspondant SFTP       IPLSPC     IPLSD
Type d'annuaire . . . . $$$$SFTP                   Portée . . . . *GLOBAL
Nom du correspondant  . EDIPSFTUSRATOB             Type réseau  . $$$$SFTP
Libellé correspondant . TBT400 Mta 200708110801
Client  User  . . . . . EDIPSFTUSRA                    Suplec N Ccsid    1252 ① ② ③
        Password  . . . PSW                            Compat N Pubkey N      ④ ⑤ ⑥
Serveur User  . . . . . EDIPSFTUSRB                             Ccsid    1252 ⑦ ⑧
        Password  . . . PSW                            Délai. N Pubkey N      ⑨ ⑩ ⑪
Dir / Emis  . ⑫
Dsn / Emis  .  FIC.&CPTUS09.&NOMFIL ⑬

Dsn / New . . ⑭
Dir / Scrut . ⑮
Dsn / Scrut . ⑯
Exchange  . . ⑰
Hmac  . . . . ⑱
Chiffrement . ⑲
Compression . ⑳
Exchange Max                          Sftsiz  32768
Sélection d'application A 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 :

① *Client User* : Nom du client SFTP tel que déclaré sur le serveur

② *Suplec* : Suppression du fichier distant après lecture

③ *Ccsid* : Code page des commandes SFTP en mode client

④ *Password* : Mot de passe du client SFTP tel que déclaré sur le serveur

⑤ *Compat* : Mode client uniquement permet la compatibilité avec serveur SSH 1.99

⑥ *Pubkey* : Mode client permet l'authentification par clé publique

⑦ *Serveur User* : Nom du serveur SFTP à remettre au client

⑧ *Ccsid* : Code page des commandes SFTP en mode serveur

⑨ *Password* : Mot de passe du serveur SFTP à remettre au client

⑩ *Delai* : Mode serveur : Ecriture différé en fin de connexion

⑪ *Pubkey* : Mode serveur permet l'authentification par clé publique

⑫ *Dir / Emis* : Répertoire distant pour émission

⑬ *Dsn / Emis* : Nom de fichier distant pour émission

⑭ *Dsn / New* : Nom de fichier distant pour renommage après émission

⑮ *Dir / Scrut* : Répertoire distant pour scrutation

⑯ *Dsn / Scrut* : Nom de fichier distant pour scrutation

⑰ *Exchange* : Algorithmes d'échange de clés dans l'ordre de préférence

⑱ *Hmac* : Algorithmes de validation HMAC dans l'ordre de préférence

⑲ *Chiffrement* : Algorithmes de chiffrement dans l'ordre de préférence

⑳ *Compression* : Algorithmes de compression dans l'ordre de préférence

Exchange Max : Quantité max de données échangées avant un renouvellement de clés

Sftsiz : Taille maximale de paquet

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.

4.3.3. 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   . ①

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 :

① *Utilisation Ssl* : N

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

4.3.4. 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           *Détail des certificats *          IPLS08     IPLSD
Type d'annuaire . . . . $$$$SFTP                   Portée . . . . *GLOBAL
Nom du correspondant  . EDIPSFTUSRATOB             Type réseau  . $$$$SFTP
Libellé correspondant . TBT400 Mta 150605090234

Certificat local  Ssl .                      K ①


Certificat remote Ssl .                      K ②
Certificat local  Aut .                      K
Certificat remote Aut .                      K
Certificat local  Sig . *S                   K ③
Certificat remote Sig . *S                   K ④
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 :

① *Certificat local Ssl*

② *Certificat remote Ssl*

③ *Certificat local Sig*

④ *Certificat remote Sig*

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

4.3.5. 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                    Apprentissage . O O,N
Ssl obligatoire . . . .   Option   Protocole       Cipher
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.

5. Importation de la clé publique dans le DCM TBT400

En mode client ou serveur, utilisez la commande suivante pour appliquer la clé publique du partenaire distant au correspondant de TBT400.

```
IPLSP/IPSCERTIFS CRTFNC(*DELSSH) ①
                 TYPRES($$$$SFTP)
                 NOMLOG(IPLS)
                 CRTCTX(*REMSSL) ②
                 IFSOBJ('/home/cert/ssh_rsa_4096.pub') ③
```

Avec:

① *CRTFNC(*DELSSH)* : « Delivery mode », ajoute la clé publique (si déjà présent l'ancien est conservé, ou « replace mode » **REPSSH* pour remplacer la clé publique existante,

② *CRTCTX(*REMSSL)* : Clé publique de cryptage distant (remote)

③ *IFSOBJ('xxxxxxxx')* : Chemin IFS de la clé publique à importer

La clé publique doit être au format standard SSH et formater sans entête ssh-rsa ou BEGIN/END, ni tout autre commentaire comme indiqué ci-dessous.

5.1. Entête BEGIN/END

```
Browse : /home/cert/ssh_rsa_4096.begin
 Record :       1   of       4 by  18            Column :    1     64 by 131
 Control :

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
 ************Beginning of data**************
-----BEGIN CERTIFICATE-----
AAAAB3NzaC1yc2EAAAABJQAAAIEAw+/hLuduVH+/rag+wQeDC/l4DViihWBs8dij
zIU59A21wU0bJ6YTKhryQjtwNM9a6en5aJPHW2kxhMaLLwoOHcrbqZQD/VmAM7qP
qpxFHlYxu5NNanmBM/qIEGiqJPMgoP1kIPX97YN22DSKvH79QaltuA12Nf4VMhF1
QALpNp0=
-----END CERTIFICATE-----
 ************End of Data*******************
```

```
Browse : /home/cert/ssh_rsa_4096.beginssh2
 Record :       1   of       4 by  18            Column :    1     64 by 131
 Control :

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
 ************Beginning of data**************
-----BEGIN SSH2 PUBLIC KEY-----
Comment: “rsa-key-20201208”
AAAAB3NzaC1yc2EAAAABJQAAAIEAw+/hLuduVH+/rag+wQeDC/l4DViihWBs8dij
zIU59A21wU0bJ6YTKhryQjtwNM9a6en5aJPHW2kxhMaLLwoOHcrbqZQD/VmAM7qP
qpxFHlYxu5NNanmBM/qIEGiqJPMgoP1kIPX97YN22DSKvH79QaltuA12Nf4VMhF1
QALpNp0=
-----END SSH2 PUBLIC KEY-----

 ************End of Data*******************
```

5.2. Entête ssh-rsa

```
Browse : /home/cert/ssh_rsa_4096.sshrsa
 Record :       1   of       4 by  18            Column :    1     64 by 131
 Control :

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
 ************Beginning of data**************
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAw+/hLuduVH+/rag+wQeDC/l4DViihWBs8dij
zIU59A21wU0bJ6YTKhryQjtwNM9a6en5aJPHW2kxhMaLLwoOHcrbqZQD/VmAM7qPr4ztztyz
qpxFHlYxu5NNanmBM/qIEGiqJPMgoP1kIPX97YN22DSKvH79QaltuA12Nf4VMhF15f3nz8b1
QALpNp0= technic@ipls
 ************End of Data*******************
```

5.3. Formatage nécessaire de la clé publique sans entête ni commentaire

```
Browse : /home/cert/ssh_rsa_4096.pub
 Record :       1   of       4 by  18            Column :    1     64 by 131
 Control :

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
 ************Beginning of data**************
AAAAB3NzaC1yc2EAAAABJQAAAIEAw+/hLuduVH+/rag+wQeDC/l4DViihWBs8dij
zIU59A21wU0bJ6YTKhryQjtwNM9a6en5aJPHW2kxhMaLLwoOHcrbqZQD/VmAM7qP
qpxFHlYxu5NNanmBM/qIEGiqJPMgoP1kIPX97YN22DSKvH79QaltuA12Nf4VMhF1
QALpNp0=
 ************End of Data*******************
```

La clé publique contenue dans le fichier « home/cert/ssh_rsa_4096.pub » est maintenant stocké dans le DCM TBT sous le nom prévu dans l'annuaire AS2 pour le correspondant IPLS.

Vous devrez trouver le message suivant pour valider l'application de la clé publique.

```
16563254 IPS IPLSGE *TBT/400* TBT : Fonction IPSPGCERTS exécutée correctement : Job=456300/IPLSPC/IPLSGE
Pgm=QSYS/QCMD Mi=456
Tbt=IPSPGCERTS code retour 0 - Cpu=4 msec Tot=75 msec Stg=17530880
```

6. 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 précisés se verront donc affectés les choix fixés dans IPLS$$$PROFIL.

7. Transférer un fichier

7.1. 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 qualifiants 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.

7.2. 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)
```

8. Commandes spécifiques

8.1. 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 . . . . . CRLFIN
Suppression blancs . . . . . SPADEM
Traduction ASCII . . . . . ASCDEM
Ccsid demandé  . . . . . CCSID
Enregistrements / segment  . . . RECSEG
Application de réception . . . . APPREC
```

8.2. 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
```

9. Exemple d'implémentation de la commande *IPSNDSFTP*

9.1. 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.

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.

9.2. 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

9.3. 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.

9.4. 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.

9.5. 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, dés 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.

9.6. Sources des programmes CL

La bibliothèque des programmes IPLSP (par défaut) doit être en ligne pour compiler ces sources.

9.6.1. *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          */
/*                          : (déja 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... */
             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
```

9.6.2. *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)
            IF         COND(&RTNCDP *NE 0) THEN(GOTO CMDLBL(ERREUR))
/* Pas d'erreur... */
            SNDPGMMSG  MSG('Appel *TBT/400* pour réception OK')
            GOTO       CMDLBL(FIN)
 ERREUR:    SNDPGMMSG  MSG('Erreur de traitement...')
            SNDPGMMSG  MSG('MSGTXT: ' *CAT &MSGTXT)
            GOTO       CMDLBL(FIN)
/* Autres traitements d'erreurs... */
 FIN:       ENDPGM
```

9.6.3. *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(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(&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')
            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)
/*************************************************************/
/* 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 transmision 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(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
```

9.6.4. *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:      SNDPGMMSG  MSG('Traitement SFTP OK') TOUSR(QSYSOPR)
             SNDPGMMSG  MSG('NOMLOG: ' *CAT &NOMLOG) TOUSR(QSYSOPR)
             GOTO       CMDLBL(FIN)
 FINKO:      SNDPGMMSG  MSG('Traitement SFTP en erreur...') TOUSR(QSYSOPR)
             SNDPGMMSG  MSG('NOMLOG: ' *CAT &NOMLOG) TOUSR(QSYSOPR)
             SNDPGMMSG  MSG('ACKTBT: ' *CAT &ACKTBT) TOUSR(QSYSOPR)
             SNDPGMMSG  MSG('LIBTBT: ' *CAT &LIBTBT) TOUSR(QSYSOPR)
             GOTO       CMDLBL(FIN)
/* Autres traitements d'erreurs... */
 FIN:        ENDPGM
```

9.6.5. *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')
             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)
/*************************************************************/
/* 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 transmision 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
```

9.6.6. *SFTTRTMSG*

```
/*-------------------------------------------------------------------*/
/*                                                                   */
/*  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) /* +
                          Libéllé d'acheminement (zone de retour) */
             DCL        VAR(&MSGCMD) TYPE(*CHAR) LEN(64)
/* Le fichier reçu par *TBT/400* est identifié par les champs:                */
/* - &OBJLIB: Bibliotèque de reception,                                   */
/* - &OBJFIL: Fichier     de reception,                                   */
/* - &OBJMBR: Membre      de reception.                                   */
/*                                                                        */
/* 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 exitant - *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'appeller 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 = Libéllé 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 = Libéllé d'acheminement négatif                        */
/*                                                                   */
 ERREUR:     CHGVAR     VAR(&ACKTBT) VALUE('KO')
             CHGVAR     VAR(&LIBTBT) VALUE('Traitement en erreur - +
                          voir : ' *BCAT &MSGCMD)
             GOTO       CMDLBL(FIN)
 FIN:        ENDPGM
```