
Connect:Express™ Unix for SWIFTNet

Delivery notifications
Store and Forward

Version 2.0.5

Revision Date: 30 Juin 2005

Sterling Commerce
An IBM Company

Connect:Express Unix For SWIFTNet
Delivery Notifications et Store and Forward

Version 2.0.5
Première édition

Le présent document a été rédigé pour aider les utilisateurs autorisés du système Connect:Express Unix for SWIFTNet de Sterling Commerce, Inc; son contenu ne pourra être utilisé dans un autre but sans autorisation écrite préalable. Le matériel visé aux présentes est fourni, sans aucune garantie d'aucune sorte et avec pour principe qu'il en sera fait une utilisation adéquate. Toute utilisation inhabituelle peut produire des résultats imprévisibles. En conséquence, Sterling Commerce ne pourra répondre et ne pourra être tenu pour responsable de quelque manière que ce soit du fait de la fourniture ou de l'utilisation de ce document ou du matériel qu'il vise.

Les références dans le présent manuel aux produits, programmes ou services de Sterling Commerce n'impliquent pas que Sterling Commerce a l'intention de les mettre à disposition dans tous les pays où Sterling Commerce exerce une activité.

Droits limités: L'utilisation, la reproduction ou la divulgation par le Gouvernement des Etats-Unis est soumise aux restrictions prévues dans le FAR 52.227-19.

© 2003 Sterling Commerce, Inc.

Tous les droits sont réservés, y compris les droits de reproduction de tout ou partie du présent document sous quelque forme que ce soit.

Connect:Express est une marque déposée de Sterling Commerce, Inc. Tous les autres noms de branche ou de produit sont des marques commerciales ou des marques déposées de leurs sociétés respectives.

TABLE DES MATIERES

CONNECT:EXPRESS™ UNIX FOR SWIFTNET	I
DELIVERY NOTIFICATIONS.....	I
STORE AND FORWARD	I
COMPOSITION DE CE MANUEL.....	IV
DELIVERY NOTIFICATIONS	1
CONFIGURATION DE LA SAG POUR GESTION DES DELIVERY NOTIFICATIONS	1
DELIVERY NOTIFICATION SUR PUT SORTANT	2
DELIVERY NOTIFICATION SUR PUT ENTRANT	3
DELIVERY NOTIFICATION SUR GET SORTANT	4
DELIVERY NOTIFICATION SUR GET ENTRANT.....	5
REMONTEE VERS LE BACK OFFICE DES NOTIFICATIONS ENTRANTES.....	6
<i>Fichier d'environnement environment.xml</i>	6
<i>Fichiers symboliques</i>	6
REPRISE DES TRANSFERTS DES NOTIFICATIONS ENTRANTES VERS LE BACK OFFICE.....	9
STORE AND FORWARD	1
CONFIGURATION DE LA SAG POUR LE STORE AND FORWARD	1
<i>End points</i>	1
ACQUISITION DE LA QUEUE SNF	3
PUT ET DELIVERY NOTIFICATIONS ENTRANTS	4
PUT SORTANTS.....	5

Préface

Cette documentation décrit les éléments permettant d'utiliser les mécanismes de « delivery notifications » et de « store and forward » de SWIFT avec FTI de la SWIFTAlliance Gateway et Connect:Express Unix for SWIFTNet.

Elle est destinée aux gestionnaires des transferts sur le réseau SWIFTNet et aux administrateurs de Connect:Express. Elle suppose connus le produit Connect:Express Unix for SWIFTNet et les fonctionnalités de la SAG.

FTI (File transfer interface) est décrit dans le document SWIFT « *SWIFTAlliance Gateway – File Transfer Interface Guide* ».

Remarque :

Les n° de version 5V2.0.5 et 5V2.0.4 mentionnés dans ce document désignent les n° de version de la partie spécifique SWIFTNet de Connect:Express. Connect:Express (Moniteur PeSIT) a, par ailleurs, son propre n° de version indépendant (Par exemple 141-3).

Composition de ce manuel

Les informations sont organisées dans ce manuel de la façon suivante :

Type d'information	Section
Delivery notifications	Chapitre 1 : Delivery notifications
Store and Forward	Chapitre 2 : Store and Forward

Chapitre 1

Delivery notifications

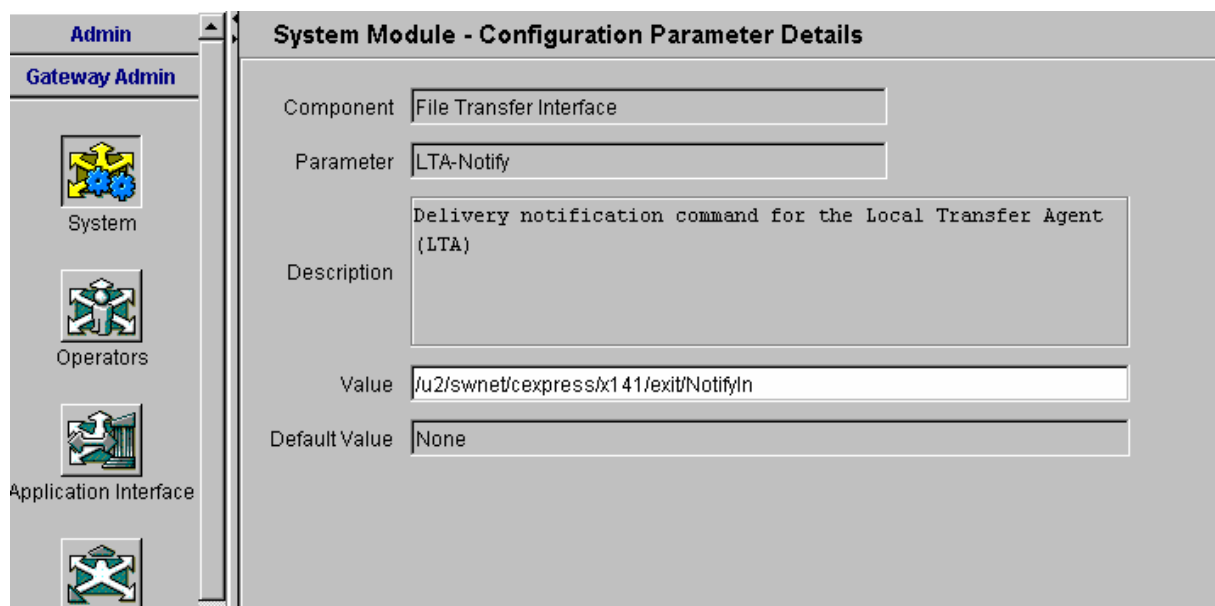
La gestion complète des « delivery notifications » n'est possible qu'à partir de la version 5V2.0.5 de la partie spécifique SWIFTNet de Connect:Express.

Pour les versions antérieures, même s'il est possible d'envoyer et de recevoir des delivery notifications via FTI, la réception de celles-ci n'est pas traitée complètement par Connect:Express.

Ce chapitre décrit l'utilisation des « delivery notifications » avec Connect:Express for SWIFTNet 5V2.0.5.

Configuration de la SAG pour gestion des delivery notifications

Afin que Connect:Express puisse prendre en charge les « delivery notifications » entrantes, il est nécessaire de valoriser, à l'aide de la WebStation, le paramètre LTA-Notify de la Gateway de la manière suivante :



NotifyIn est l'exécutable de Connect:Express prenant en charge les delivery notifications entrantes.

Delivery notification sur PUT sortant

Il est possible de demander, lors d'un put sortant, au correspondant distant de nous renvoyer une delivery notification confirmant qu'il a bien reçu le fichier.

Ceci se fait à l'aide d'un paramètre `<a18>TRUE</a18>` présent dans le fichier squelette associé au transfert.

Exemple de fichier squelette SwiftPut.xml

```
<Ft:PutFileRequest>
<a1>o=ptsfrmn,o=swift</a1>
<a2><1>,o=swift</a2>
<a3>swift.generic.fast!x</a3>
<a4>RequestType</a4>
<a5>cn=xxxxxx,o=ptsfrmn,o=swift</a5>
<a6><3></a6>
<a7>TransferInfo</a7>
<a8>UserReference</a8>
<a9></a9>
<a10><2></a10>
<a11>File Description</a11>
<a12>SwCompression=<4>,coding=ascii</a12>
<a13>TRUE</a13>
<a14>FALSE</a14>
<a18>TRUE</a18>
<a28>TRUE</a28>
</Ft:PutFileRequest>
```

Note : La valorisation à TRUE du tag `<a18>` peut aussi être faite dynamiquement par le Back Office, via un paramètre TRUE placé dans le Pi99 du transfert PeSIT SWIFTPUT.

A réception de la notification venant du correspondant, l'exécutable NotifyIn est activé. Ce dernier enregistre la notification dans le fichier swift.log de la manière suivante :

Exemple de delivery notification entrante (sur PUT sortant) dans swift.log

20050628174324 PUT CENTRAL 17900112	SWIFTPUT started
20050628174324 PUT CENTRAL 17900112	No rule number indicated in Pi99. Using default rule number 000
20050628174324 PUT CENTRAL 17900112	RequestorDN : o=ptsufrmn,o=swift. ResponderDN :
o=ptsufrmn,o=swift.Delivery notification expected	
20050628174324 PUT CENTRAL 17900112	No compression applied to the file
/u2/swnet/cexpress/x141/in/put_B7900112.tmp	
20050628174335 NTF	NotifyIn started
20050628174335 NTF SNL01594D11119973406030036C	DeliveryStatus:Accepted.DeliveryTime:2005-06-28T15:44:19Z.FileDigest:PWNGNVdCb9pxXE/UCL6kI7EFvbc=.AckDirection:Incoming.SnF:FALSE.SessionId:.
20050628174340 PUT CENTRAL 17900112	Ft-put command is OK
20050628174340 PUT CENTRAL 17900112	Ft-Put OK. SNL Transfer Reference :
SNL01594D11119973406030036C. UserRef : UserReference.	
20050628174340 PUT CENTRAL 17900112	SWIFTPUT terminated successfully
20050628174340 NTF SNL01594D11119973406030036C	NotifyIn terminated successfully
20050628174340 CLA CENTRAL 17900114	CLEARACK started
20050628174340 CLA CENTRAL 17900114	CLEARACK terminated successfully

Les éléments suivants sont indiqués :

Référence SNL du transfert

DeliveryStatus: Accepted/Rejected

DeliveryTime

FileDigest

AckDirection : Incoming pour un put sortant

SnF Store and forward utilisé ou pas

SessionId Identificateur de session Store and Forward (si Snf=TRUE)

La corrélation entre la notification et la requête initiale peut être faite avec la référence SNL du transfert et la référence utilisateur (UserRef) fournie par le Back Office.

Delivery notification sur PUT entrant

Lors d'un PUT entrant, si le correspondant a demandé qu'on lui renvoie une delivery notification, celle-ci est automatiquement générée et envoyée sur SWIFTNet par FTI.

Une indication de la notification sortante est fournie par FTI, par activation de l'exécutable NotifyIn.

NotifyIn enregistre le contenu de la notification émise dans le fichier swift.log de la manière suivante :

Exemple de delivery notification sortante (sur PUT entrant) dans swift.log

20050628174331 API	ConnectAcceptPutInit started
20050628174331 API SNL01594D11119973406030036S o=ptscfrnn,o=swift	RequestorDN : o=ptscfrnn,o=swift. ResponderDN :
20050628174331 API SNL01594D11119973406030036S	Controlling access. There is no white list
20050628174331 API SNL01594D11119973406030036S	Controlling access. There is no black list
20050628174331 API SNL01594D11119973406030036S	Put file authorized
20050628174331 API SNL01594D11119973406030036S	ConnectAcceptPutInit terminated successfully
20050628174332 FWD	SWIFTFWD started
20050628174332 FWD	Parameter file location =
/SWIFTAlliance/Gateway/FT/data/xmlparamfile/filename.SNL01594D11119973406030036S_putend.par	
20050628174332 FWD SNL01594D11119973406030036S o=ptscfrnn,o=swift.Delivery notification expected	RequestorDN : o=ptscfrnn,o=swift. ResponderDN :
20050628174332 FWD SNL01594D11119973406030036S	File /u2/swnet/cexpress/x141/exit/FindRule.xml doesn't exist. Using default rule 000
20050628174332 FWD SNL01594D11119973406030036S	No uncompression applied to the file
/SWIFTAlliance/Gateway/FT/data/reception/SNL01594D11119973406030036S	
20050628174332 FWD SNL01594D11119973406030036S	File filename successfully forwarded to the back office
CENTRAL	
20050628174332 FWD SNL01594D11119973406030036S	SWIFTFWD terminated successfully
20050628174332 CLF CENTRAL 17900113	CLEARFWD started
20050628174332 CLF CENTRAL 17900113	Parameter file
/SWIFTAlliance/Gateway/FT/data/reception/SNL01594D11119973406030036S.par deleted	
20050628174332 CLF CENTRAL 17900113	stat for
/SWIFTAlliance/Gateway/FT/data/reception/SNL01594D11119973406030036S.par_err returns errno 2	
20050628174332 CLF CENTRAL 17900113	CLEARFWD terminated successfully
20050628174337 NTF	NotifyIn started
20050628174337 NTF SNL01594D11119973406030036S	DeliveryStatus:Accepted.DeliveryTime:2005-06-28T15:44:19Z.FileDigest:PWNGNVdCb9pxXE/UCL6kI7EFvbc=.AckDirection:Outgoing.SnF:FALSE.SessionId.:
20050628174337 NTF SNL01594D11119973406030036S	NotifyIn terminated successfully

Delivery notification sur GET sortant

Lors d'un GET sortant, le correspondant distant peut demander à ce qu'on lui fasse parvenir une delivery notification à l'issue du téléchargement. Ceci est fait automatiquement par FTI. FTI active NotifyIn, une fois la notification émise. NotifyIn indique dans le fichier swift.log les éléments de la notification envoyée.

Exemple de delivery notification sortante (sur GET sortant) dans swift.log

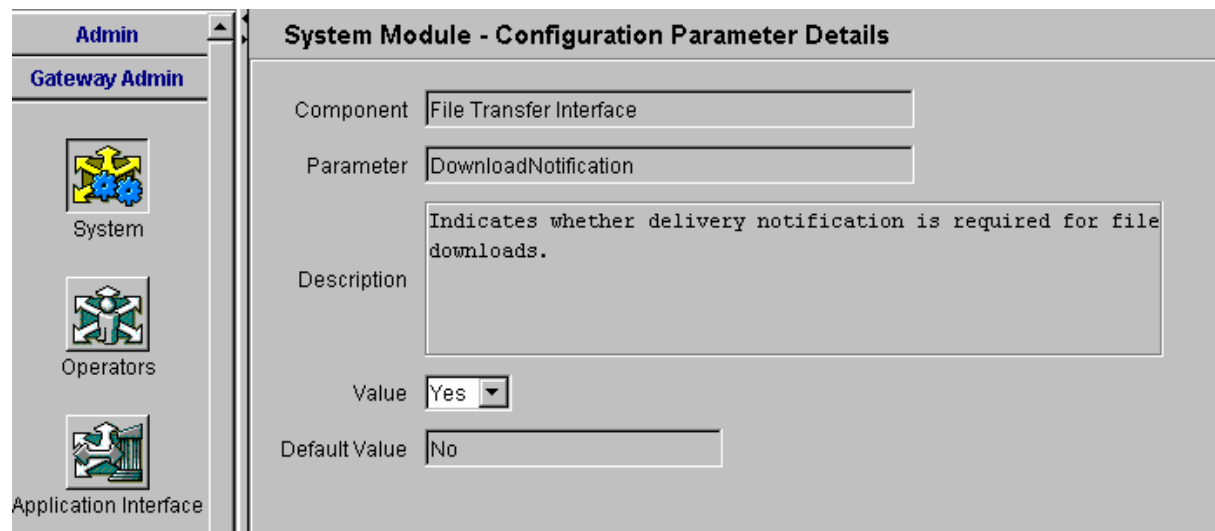
...	
20050628174147 NTF	NotifyIn started
20050628174147 NTF SNL01594D11119973287029899C	DeliveryStatus:Accepted.DeliveryTime:2005-06-28T15:42:25Z.FileDigest:ZidG3OpOuvBS5BN0iMby+CA51rc=.AckDirection:Outgoing.SnF:FALSE.SessionId.:
20050628174147 NTF SNL01594D11119973287029899C	NotifyIn terminated successfully
...	

La référence SNL permet de faire le lien avec les autres éléments du transfert présents dans swift.log.

Delivery notification sur GET entrant

Il est possible de demander aux correspondants qui viennent télécharger des fichiers chez nous de nous faire parvenir une notification pour chaque téléchargement.

Ceci se fait en renseignant avec la valeur « Yes » le paramètre DownloadNotification de la SAG, à l'aide de la WebStation, de la manière suivante :



Il faut noter que si ce paramètre est activé, toutes les demandes de téléchargement entrantes donnent lieu systématiquement à une demande de notification au correspondant.

Par ailleurs, il est nécessaire de paramétrer dans le fichier environment.xml de Connect:Express, le paramètre DownloadNotification :

\$TOM_DIR/exit/environment.xml

```
<TOM_DIR>/u2/swnet/cexpress/x141</TOM_DIR>
<CENTRAL>CENTRAL</CENTRAL>
<SAG_DIR>/SWIFTAlliance/Gateway</SAG_DIR>
<SAG_XML>/SWIFTAlliance/Gateway/FT/data/xmlparamfile</SAG_XML>
<SAG_RCV>/SWIFTAlliance/Gateway/FT/data/reception</SAG_RCV>
<SAG_DOWN>/SWIFTAlliance/Gateway/FT/data/download</SAG_DOWN>
<SAG_LOG>/SWIFTAlliance/Gateway/FT/data/log</SAG_LOG>
<SAG_BIN>/SWIFTAlliance/Gateway/bin</SAG_BIN>
<DownloadNotification>TRUE</DownloadNotification>
<NotifyBackOffice>FALSE</NotifyBackOffice>
```

Chaque notification envoyée par FTI est enregistrée par NotifyIn dans swift.log :

Exemple de delivery notification entrante (sur GET entrant) dans swift.log

```
...
20050628174144 NTF                               NotifyIn started
20050628174144 NTF SNL01594D11119973287029899S   DeliveryStatus:Accepted.DeliveryTime:2005-06-
28T15:42:25Z.FileDigest:ZidG3OpOuvBS5BN0iMby+CA51rc=.AckDirection:Incoming.SnF:FALSE.SessionId:.
20050628174144 NTF SNL01594D11119973287029899S   NotifyIn terminated successfully
...
```

La référence SNL permet de faire le lien avec les autres éléments du transfert présents dans swift.log.

Remontée vers le back office des notifications entrantes

Les delivery notification sortantes ne sont jamais remontées par Connect:Express vers le back office.

Par défaut, les delivery notifications entrantes ne sont pas remontées.

Pour remonter les delivery notifications venant des correspondants distants vers le back office, il est nécessaire d'ajouter les éléments de paramétrage de Connect:Express supplémentaires suivants :

Fichier d'environnement environment.xml

\$TOM_DIR/exit/environment.xml

```
<TOM_DIR>/u2/swnet/cexpress/x141</TOM_DIR>
<CENTRAL>CENTRAL</CENTRAL>
<SAG_DIR>/SWIFTAlliance/Gateway</SAG_DIR>
<SAG_XML>/SWIFTAlliance/Gateway/FT/data/xmlparamfile</SAG_XML>
<SAG_RCV>/SWIFTAlliance/Gateway/FT/data/reception</SAG_RCV>
<SAG_DOWN>/SWIFTAlliance/Gateway/FT/data/download</SAG_DOWN>
<SAG_LOG>/SWIFTAlliance/Gateway/FT/data/log</SAG_LOG>
<SAG_BIN>/SWIFTAlliance/Gateway/bin</SAG_BIN>
<DownloadNotification>TRUE</DownloadNotification>
<NotifyBackOffice>TRUE</NotifyBackOffice>
```

Fichiers symboliques

A l'aide de STERM, déclarer 4 fichiers symboliques supplémentaires NTPACK, NTPNAK, NTGACK et NTGNAK permettant d'effectuer des transferts PeSIT de delivery notification dans le sens SAG vers Back Office.

Pour le fichier symbolique NTPACK, la définition est la suivante :

Definition de Fichier NTPACK

```

C:X 141-3 ----- FILES DIRECTORY ----- TOM1
OPTION ==>
SYMBOLIC NAME :          NTPACK
INITIALIZATION STATUS . : E          E:ENABLE H:DISABLE
DIRECTION ..... :      T          T:TRANSMIT R:RECEIVE *:EITHER
RECEIVING PARTNER ..... : #CENTRAL 'NAME',#LISTE, $$ALL$$
TRANSMITTING PARTNER .. : #CENTRAL 'NAME',#LISTE, $$ALL$$
PRIORITY ..... :      0          0:URGENT 1:FAST 2:NORMAL
DEFINITION TYPE ..... :      D      D:DYNAMIC F:FIXED
PRESENTATION TABLE .... : 1        1 -> 9 PRESENTATION TABLE
PARAMETER CARDS FILE :      N          Y/N
SPACE TO RESERVE ..... :      N          Y/N
ALLOCATION RULE ..... :      0          0:INDIF., 1:PREALL., 2:TO CREATE
PHYSICAL NAME ..... :      $SAG_XML

RECORD FORMAT ..... :      TV          TF, TV, BF, BU, T*, B*, **
RECORD LENGTH ..... :      08192      1-5 NUMERIC CHARAC.
REMOTE DSN (FTP) ..... :
TYPE/STRUCTURE/MODE FTP :          E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) .... :          Y/N          FA : Y/N
OPTION : VIEW UPD : 98/03/13 14:59 pla
-ENTER- NEXT FIELD -F3- CANCEL -F8- COMPLETION
    
```

```

C:X 141-3 ----- FILES DIRECTORY ----- TOM1
OPTION ==>
SYMBOLIC NAME          :          NTPACK          DEFINITION : D DIRECTION : R
TRANSMISSION :
START EXIT ..... : .....
START COMMAND ..... : .....
END EXIT ..... : .....
END COMMAND ..... : CLEAR....
RECEPTION :
START EXIT ..... : .....
START COMMAND ..... : .....
END EXIT ..... : .....
END COMMAND ..... : .....
DO YOU WANT TO GO ON ? UPD : 19980722112010 C:E 140-1
-ENTER- NEXT FIELD -F3- CANCEL -F8- COMPLETION
    
```

Les définitions des fichiers NTPNAK, NTGACK et NTGNAK sont similaires à la précédente en substituant NTPACK par les valeurs respectives.

Les fichiers transmis au back office par ces fichiers symboliques sont des fichiers xml semblables aux fichiers d'acquittement SWIFTACK et SWIFTNAK déjà utilisés.

NTPACK et NTPNAK correspondent à des delivery notifications positives/négatives (Accepted/Rejected) sur PUT sortant.

NTGACK et NTGNAK correspondent à des delivery notifications positives/négatives (Accepted/Rejected) sur GET entrant.

Dans le cas du PUT sortant chaque fichier transféré contient des éléments permettant d'effectuer la corrélation avec la requête SWIFTPUT initiale.

Dans le cas d'un GET entrant, les caractéristiques de la demande de téléchargement du correspondant sont remontées.

Exemple de fichier transféré par NTPACK

```

<?xml version='1.0' encoding='UTF-8'?>
<Cx:StatusAttributes>
    
```

```

<Cx:Severity>Information</Cx:Severity>
<Cx:Code>OK</Cx:Code>
<Cx:Text>Notification received. Transfer accepted</Cx:Text>
<Cx:processingState>Ended</Cx:processingState>
<Cx:Component>NotifyIn</Cx:Component>
<Cx:ExpressRequestNumber>17900112</Cx:ExpressRequestNumber>
<Cx:Partner>CENTRAL</Cx:Partner>
<Cx:SNLTransferRef>SNL01594D11119973406030036C</Cx:SNLTransferRef>
<Cx>UserRef>UserReference</Cx>UserRef>
<Cx:FileName>filename</Cx:FileName>
<Cx:Operation>PUT</Cx:Operation>
<Cx:BPi99>o=ptscfrnn:filename:idt</Cx:BPi99>
<Cx:Details>
<Cx:NotificationContent>
  <?xml version='1.0' encoding='UTF-8'?>
  <Ft:Notify>
    <TransferRef>SNL01594D11119973406030036C</TransferRef>
    <DeliveryStatus>Accepted</DeliveryStatus>
    <DeliveryTime>2005-06-28T15:44:19Z</DeliveryTime>
    <FileDigest>PWNGNVdCb9pxXE/UCL6kI7EFvbc=</FileDigest>
    <AckDirection>Incoming</AckDirection>
    <SnF>FALSE</SnF>
    <SessionId></SessionId>
  </Ft:Notify>
</Cx:NotificationContent>
</Cx:Details>
</Cx>StatusAttributes>

```

Les valeurs des Pi37 (label) et Pi99 fournies lors de transferts NTPACK / NTPNAK sont identiques aux valeurs fournies pour les transferts correspondants SWIFACK / SWIFTNAK :

Pi37 : Nom logique de fichier (au sens SWIFTNet).

Pi99 : Tel que défini par la règle de RuleList.xml associée au transfert.

Exemple de fichier transféré par NTGACK

```

<?xml version='1.0' encoding='UTF-8'?>
<Cx>StatusAttributes>
  <Cx:Severity>Information</Cx:Severity>
  <Cx:Code>OK</Cx:Code>
  <Cx:Text>Notification received. Transfer accepted</Cx:Text>
  <Cx:processingState>Ended</Cx:processingState>
  <Cx:Component>NotifyIn</Cx:Component>
  <Cx:ExpressRequestNumber></Cx:ExpressRequestNumber>
  <Cx:Partner>CENTRAL</Cx:Partner>
  <Cx:SNLTransferRef>SNL01594D11119973287029899S</Cx:SNLTransferRef>
  <Cx>UserRef>UserRef</Cx>UserRef>
  <Cx:FileName>filename</Cx:FileName>
  <Cx:Operation>GET</Cx:Operation>
  <Cx:BPi99></Cx:BPi99>
  <Cx:Details>
  <Ft:GetInit>
    <TransferRef>SNL01594D11119973287029899S</TransferRef>
    <RequestorDN>o=ptscfrnn,o=swift</RequestorDN>
    <ResponderDN>o=ptscfrnn,o=swift</ResponderDN>
    <ServiceName>swift.generic.fa!x</ServiceName>
    <RequestType>RequestType</RequestType>
    <TransferDescription>idt</TransferDescription>
    <TransferInfo>SwCompression=ZIP,coding=ascii</TransferInfo>
    <UserRef>UserRef</UserRef>
    <FileName>filename</FileName>
    <FileDescription>File downloaded from responder o=ptscfrnn,o=swift to requestor
o=ptscfrnn,o=swift</FileDescription>
  </Ft:GetInit>
</Cx:Details>
</Cx>StatusAttributes>

```

```

<FileInfo></FileInfo>
<MaxFileSize>262144000</MaxFileSize>
<FileToken>1119973289</FileToken>
<Priority>Normal</Priority>
<Signed>TRUE</Signed>
</Ft:GetInit>
<Cx:NotificationContent>
  <?xml version='1.0' encoding='UTF-8'?>
  <Ft:Notify>
    <TransferRef>SNL01594D11119973287029899S</TransferRef>
    <DeliveryStatus>Accepted</DeliveryStatus>
    <DeliveryTime>2005-06-28T15:42:25Z</DeliveryTime>
    <FileDigest>ZidG3OpOuvBS5BN0iMby+CA51rc=</FileDigest>
    <AckDirection>Incoming</AckDirection>
    <SnF>FALSE</SnF>
    <SessionId></SessionId>
  </Ft:Notify>
</Cx:NotificationContent>
</Cx:Details>
</Cx:StatusAttributes>

```

La valeur du Pi37 (label) lors de transferts NTGACK / NTGNAK est le nom logique de fichier (<FileName>) au sens SWIFTNet.
Le Pi99 n'est pas renseigné.

Reprise des transferts des notifications entrantes vers le back office

La reprise des transferts vers le back office des notifications entrantes suit le même principe que la reprise des transferts PeSIT des fichiers proprement dits par RETRYFWD (Voir le document « *Reprise des transferts venant de SWIFTNet et non acheminés vers le Back Office* »). Cette reprise peut être nécessitée, par exemple, en cas d'arrêt temporaire du moniteur PeSIT du back office.

La reprise est gérée par les exécutables RETRYFWD, RETRYNTF et UEXLOG ainsi que par le script standard d'erreur de fin de transfert UEXERR.

Dans le cas d'une mise à jour de la version 5V2.0.4 à la version 5V205 de la partie spécifique SWIFTNet de Connect:Express, il est nécessaire de mettre à jour le fichier UEXERR existant avec les lignes suivantes :

\$TOM_DIR/exit/UEXERR

```
#-----  
# Insert the following lines if RETRYFWD is to be used  
#-----  
reqno=$1  
symbfile=$2  
filename=$4  
direction=$5  
if [ "$direction" = "T" ] ; then  
  if [ "$symbfile" = "NTPACK" ] || [ "$symbfile" = "NTGACK" ] || \  
    [ "$symbfile" = "NTPNAK" ] || [ "$symbfile" = "NTGNAK" ]; then  
    eval mv ${filename} ${filename}_err  
    $TOM_DIR/exit/UEXLOG $1 $2 $3 $4 $5 $6 $7 $8  
    $TOM_DIR/itom/p1b8ppur /REQ=$reqno  
    exit 0  
  fi  
  if [ "$symbfile" != "SWIFTACK" ] && [ "$symbfile" != "SWIFTNAK" ]; then  
    eval mv ${filename}.par ${filename}.par_err  
    $TOM_DIR/exit/UEXLOG $1 $2 $3 $4 $5 $6 $7 $8  
    $TOM_DIR/itom/p1b8ppur /REQ=$reqno  
    exit 0  
  fi  
fi  
#-----
```

Notes :

UEXERR est déclenché en cas d'erreur de transfert.

A titre indicatif :

mv effectue un renommage du fichier qui n'a pu être transféré pour reprise ultérieure par RETRYNTF.

UEXLOG affiche dans swift.log les valeurs TRC, PRC et SRC des codes d'erreur de transfert PeSIT.
p1b8ppur purge la requête en erreur du fichier RENC de Connect:Express.

Chapitre 2

Store and Forward

Bien qu'il soit possible d'émettre des fichiers en mode « Store and Forward » avec la version 5V2.0.4 de la partie spécifique SWIFTNet de Connect:Express, celle-ci n'offre pas l'assurance de l'acceptation du fichier par le destinataire.

En effet, côté émetteur (PUT sortant), le dépôt correct dans la « queue SnF » du correspondant chez SWIFT, n'implique pas forcément que le fichier ait été accepté et extrait de la queue SnF par ce dernier.

Ce n'est qu'à la réception de la delivery notification (positive/négative) en provenance du correspondant que l'on peut être sûr de l'état du transfert de bout en bout ;

Ce chapitre décrit l'utilisation du mode « store and forward » avec Connect:Express for SWIFTNet 5V2.0.5.

Configuration de la SAG pour le store and forward

La SAG doit être paramétrée ainsi qu'indiqué au chapitre 1 pour la réception des delivery notifications (Paramètre LTA-Notify = .../exit/NotifyIn).

Le service SWIFT utilisé pour les requêtes sera un service FileAct en mode store and forward (par exemple, service générique : swift.generic.fast)

Les delivery notifications venant des correspondants sont déposées dans notre propre « queue SnF » (par exemple ptscfrnn_generic).

Par ailleurs, nous désirons (ou pas) recevoir des fichiers sur notre propre queue SnF.

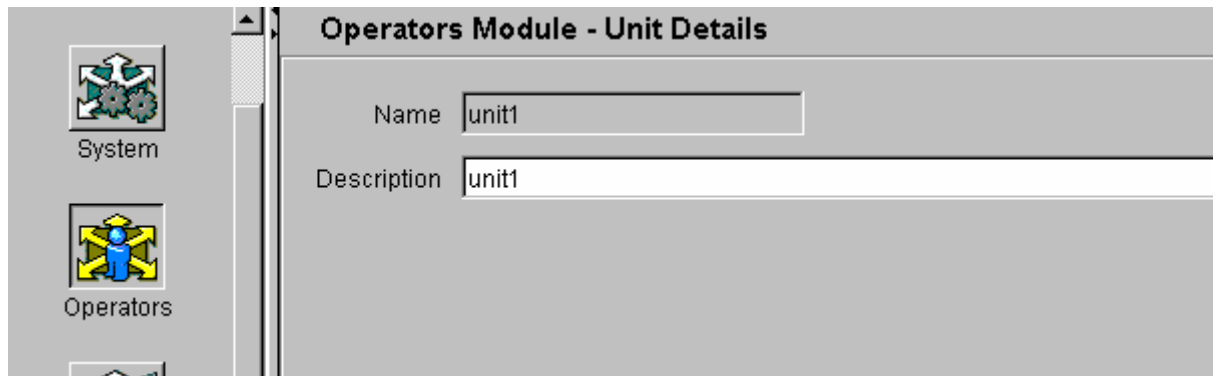
Le paramétrage ci-dessous décrit une possibilité de configuration de la SAG, pour émettre et recevoir des fichiers à la fois en temps réel et en «store and forward » avec FTI.

Pour précisions complémentaires concernant votre propre configuration, consulter la documentation SWIFT ou contacter le support de SWIFT.

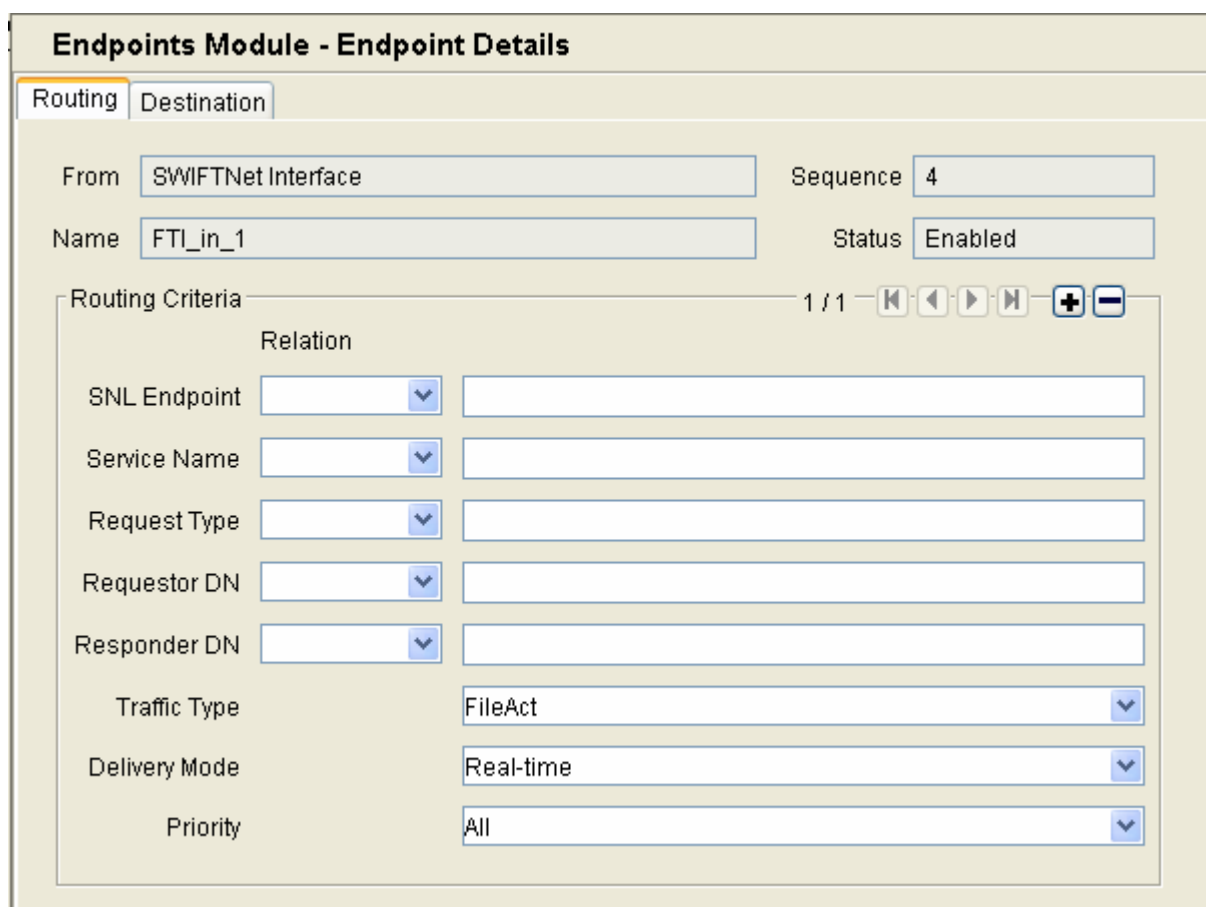
End points

Deux « endpoints modules » sont définis dans la SAG à l'aide de la WebStation, l'un FTI_in_1 pour les transferts en « temps réel », l'autre FTI_in_2 pour les transferts en « store and forward ».

Comme ces 2 endpoints FTI ne peuvent être associés à la même unité (« unit »), on crée l'unité « unit1 » en plus de l'unité par défaut « none ».



Puis on modifie le « endpoint » existant FTI_in_1 de la manière suivante :



Note :

SNL Endpoint = snl_sft

Service Name = swift.generic.fa

Puis on crée le « endpoint » FTI_in_2 suivant :

Endpoints Module - Endpoint Details

Routing
Destination

From

Name

Sequence

Status

Routing Criteria 1 / 1

	Relation	
SNL Endpoint	<input type="text" value="v"/>	<input style="width: 95%;" type="text"/>
Service Name	<input type="text" value="v"/>	<input style="width: 95%;" type="text"/>
Request Type	<input type="text" value="v"/>	<input style="width: 95%;" type="text"/>
Requestor DN	<input type="text" value="v"/>	<input style="width: 95%;" type="text"/>
Responder DN	<input type="text" value="v"/>	<input style="width: 95%;" type="text"/>
Traffic Type		<input style="width: 95%;" type="text" value="FileAct"/>
Delivery Mode		<input style="width: 95%;" type="text" value="Store-and-Forward"/>
Priority		<input style="width: 95%;" type="text" value="All"/>

Note :

SNL Endpoint = ftl_snfep

Service Name = swift.generic.fast

Acquisition de la queue SnF

Pour pouvoir recevoir, par store and forward, soit des fichiers soit des delivery notifications, il est nécessaire de faire au préalable l'acquisition de notre queue SnF.

Ceci est fait par la commande opérateur ft-acquire-queue :

Cette commande doit être lancée par un opérateur ou un script depuis le compte utilisateur UNIX de la SAG (swnet).

Par exemple :

\$TOM_DIR/exit/acq.sh_example

```
/SWIFTAlliance/Gateway/bin/ft-acquire-queue -a5 cn=xxxx,o=ptscfrnn,o=swift -a40 ptscfrnn_generic!x -a41 TRUE -o
```

-a5 est le DN de sécurité

-a40 est le nom de la queue SnF

-a41 TRUE précise que l'on force l'acquisition de la queue

On obtient :

```
Sending request...
Polling for server response...
Response received...
Analysing response...
-a44 ptscfrnn_generic!x:p:174760
-a47 1237
Queue acquisition accepted!
```

-a44 correspond à l'identifiant de session.

L'état de la queue peut être interrogé par la commande ft-getqueestatus :

```
/SWIFTAlliance/Gateway/bin/ft-getqueestatus -a44 ptscfrnn_generic!x:p:174760 -o
```

On obtient :

```
Sending request...
Polling for server response...
Response received...
Analysing response...
-a46 Active
-a5 cn=xxxxxx,o=ptscfrnn,o=swift
-a50 Push
-a42 FIFO
-a43 FALSE
-a47 1237
-a48 2005-06-28T15:55:35Z
Succeeded to get queue status: session is Active
```

Remarque:

Ni la SAG, ni Connect:Express ne disposent de mécanismes automatiques d'acquisition des queues SnF.

Ceci doit être fait soit manuellement par opérateur, soit par lancement d'un script shell spécifique, fonction des conditions d'exploitation.

PUT et delivery notifications entrants

Lorsque l'acquisition de la queue SnF est effectuée, s'il existe des notifications ou des fichiers présents dans celle-ci, ils sont transférés automatiquement par FTI de la queue chez SWIFT vers la SAG.

Les exécutable de Connect:Express définis dans les paramètres LTA-PutInIt, LTA-PutEnd et LTA-Notif de la SAG sont exécutés exactement comme dans le mode « temps réel ».

PUT sortants

L'émission d'un fichier vers la queue SnF d'un correspondant par Connect:Express se fait à l'aide de paramètres SnF spécifiques à inclure dans les fichiers squelette (SwiftPut.xml ...) associés aux transferts par les règles du fichier RuleList.xml.

Les tags de squelettes concernés sont les suivants :

Tag	Name	Description	Commentaires
<a18>	Delivery Notification requested	Demande d'envoi de delivery notification Valeurs : TRUE/FALSE	Dans le cadre de SnF, si TRUE, une delivery notification sera reçue dans tous les cas Si FALSE, seules les delivery notifications négatives seront reçues
<a26>	DeliveryNotification ReceiverDN	DN à qui sera renvoyée la delivery notification C'est le requestorDN de la requête dans le cadre de Connect:Express C'est aussi la valeur par défaut.	Voir <a1>
<a28>	SnF	Indication d'utilisation du mode store and forward Valeurs : TRUE/FALSE	TRUE : SnF FALSE : Temps réel
<a29>	Identifiant queue SnF	C'est l'identifiant de notre queue SnF sur laquelle sont reçues les delivery notifications	Exemple : ptscfrnn_generic !x

Il est possible de distinguer les transferts en temps réel des transferts SnF :

- Soit en utilisant des règles différentes dans le fichier RuleList.xml, chaque règle pointant sur un fichier squelette adapté au mode de transfert
- Soit en valorisant de manière dynamique le tag <a28> à TRUE ou FALSE à l'aide du Pi99 construit par le back office.

(Voir « *Connect:Express for SWIFTNet – Guide d'implémentation* »)

Voici, ci-dessous, un exemple de fichier squelette SwiftPut.xml :

```
<Ft.PutFileRequest>
<a1>o=ptscfrmn,o=swift</a1>
<a2><1>,o=swift</a2>
<a3>swift.generic.fast!x</a3>
<a4>RequestType</a4>
<a5>cn=xxxxxx,o=ptscfrmn,o=swift</a5>
<a6><3></a6>
<a7>TransferInfo</a7>
<a8>UserReference</a8>
```

```
<a9></a9>  
<a10><2></a10>  
<a11>File Description</a11>  
<a12>SwCompression=<4>,coding=ascii</a12>  
<a13>TRUE</a13>  
<a14>FALSE</a14>  
<a18>TRUE</a18>  
<a28>TRUE</a28>  
<a29>ptscfrnn_generic!x</a29>  
</Ft:PutFileRequest>
```