



Connect:Express® UNIX

Mise en Oeuvre de
l'Utilitaire de Notification

Version 1.4.6

Connect:Express UNIX
Mise en Oeuvre de l' Utilitaire de Notification

Version 146
Première édition

La présente documentation a pour objet d'aider les utilisateurs autorisés du système Connect:Express (ci-après le « Logiciel de Sterling Commerce »). Le Logiciel de Sterling Commerce, la documentation correspondante ainsi que les informations et le savoir-faire qu'il contient, sont la propriété de Sterling Commerce Inc. et sont confidentiels. Ils constituent des secrets commerciaux de cette dernière, de ses sociétés affiliées ou de ses/leurs concédants (ci-après dénommés collectivement « Sterling Commerce »). Ils ne peuvent pas être utilisés à des fins non autorisées ni divulgués à des tiers sans l'accord écrit préalable de Sterling Commerce. Le Logiciel de Sterling Commerce ainsi que les informations et le savoir-faire qu'il contient ont été fournis conformément à un contrat de licence qui inclut des interdictions et/ou des limitations quant à la copie, la modification et l'utilisation. La reproduction, en tout ou partie, si et lorsqu'elle est autorisée, devra inclure la présente notice d'information et la légende de copyright de Sterling Commerce Inc. Lorsqu'un Logiciel de Sterling Commerce ou un Logiciel Tiers est utilisé, reproduit ou divulgué par ou à une administration des États-Unis ou un cocontractant ou sous-traitant d'une telle administration, le Logiciel est assorti de DROITS LIMITES tels que définis au Titre 48 CFR 52.227-19 et est régi par les dispositions suivantes : Titre 48 CFR 2.101, 12.212, 52.227-19, 227-7201 à 227.7202-4, FAR 52.227-14 (g) (2) (6/87) et FAR 52.227-19 (c) (2) et (6/87), et le cas échéant, la licence habituelle de Sterling Commerce, tel que cela est décrit au Titre 48 CFR 227-7202-3 concernant les logiciels commerciaux et la documentation des logiciels commerciaux, y compris le DFAR 252-227-7013 (c) (1), 252.227-7015 (b) et (2), DFAR 252.227-7015 (b) (6/95), DFAR 227.7202-3 (a), selon le cas.

Le Logiciel de Sterling Commerce et la documentation correspondante sont concédés « EN L'ETAT » ou assortis d'une garantie limitée, telle que décrite dans le contrat de licence de Sterling Commerce. A l'exception des garanties limitées accordées, AUCUNE AUTRE GARANTIE EXPRESSE OU IMPLICITE N'EST CONCEDEE, Y COMPRIS LES GARANTIES DE QUALITE MARCHANDE ET DE CONVENANCE A UN USAGE PARTICULIER. La société Sterling Commerce concernée se réserve le droit de revoir cette publication périodiquement et d'effectuer des modifications quant à son contenu, sans obligation d'en informer qui que ce soit, personne physique ou personne morale.

Les références faites dans le présent manuel aux produits, logiciels ou services Sterling Commerce ne signifient pas que Sterling Commerce a l'intention de les commercialiser dans tous les pays dans lesquels elle a des activités.

Imprimé aux États-Unis.

Copyright © 2004,2010. Sterling Commerce, Inc. Tous droits réservés.

Connect:Express est une marque déposée de Sterling Commerce. Les noms des Logiciels Tiers sont des marques ou des marques déposées de leurs sociétés respectives. Tous (toutes) autres marques ou noms de produit sont des marques ou des marques déposées de leurs sociétés respectives.

Table des matières

PREFACE.....	V
INTRODUCTION.....	VII
L'UTILITAIRE DE NOTIFICATION	1
PARAMÉTRAGE DE L'UTILITAIRE	1
<i>Liste des paramètres de configuration de l'utilitaire.....</i>	2
<i>Logging des process tom_notify.....</i>	5
COMMANDES OPÉRATEUR	6
<i>Format général</i>	6
<i>Demande d'arrêt.....</i>	6
<i>Traitement immédiat</i>	6
<i>Activation/Désactivation de la trace</i>	6
FONCTION TRACE INTÉGRÉE	7
FONCTION DE SAUVEGARDE	7
<i>Conservation des notifications.....</i>	8
<i>Traitement de la reprise.....</i>	8
DIFFÉRENTS TYPES DE TRAITEMENT	9
<i>Traitement en tâche de Fond</i>	9
<i>Traitement par lots.....</i>	10
<i>Traitement au fil de l'eau.....</i>	11
DIFFÉRENTS MODES D'UTILISATION	12
<i>Utilisation en batch / à partir de la ligne de commande.....</i>	12
<i>Utilisation depuis un programme C.....</i>	12
UTILISATION EN ENVIRONNEMENT APPLICATIF	1
ENVOI D'UN FICHER	1
<i>Paramétrage</i>	1
<i>Gestion des echecs</i>	2
ENVOI D'UN LOT DE FICHIERS.....	3
<i>Paramétrage</i>	3
<i>Gestion des echecs</i>	4
ENVOI D'UN TEXTE SIMPLE.....	4
<i>Paramétrage</i>	4
<i>Mots Clés dynamiques</i>	5
<i>Gestion des Echecs</i>	5
NOTIFICATIONS DE CONNECT:EXPRESS.....	1
UTILISATION DE LA FONCTION JOURNAL	1
<i>SYSIN</i>	1
<i>Fichier symbolique (STERM).....</i>	2
<i>Requête de transfert (STERM).....</i>	2
<i>Requête de transfert (p1b8preq).....</i>	3
<i>Paramétrage</i>	3
<i>Gestion des echecs</i>	4
CONSTRUCTION DYNAMIQUE D'UNE NOTIFICATION	4
<i>Paramétrage</i>	5
<i>Variables dynamiques.....</i>	5
<i>Gestion des echecs</i>	6
INSTALLATION ET UTILITAIRES DE MISE AU POINT.....	1
ELEMENTS INSTALLÉS	1

UTILITAIRE D'ARCHIVAGE DU FICHIER NOTIF.LOG	2
UTILITAIRE DE MISE AU POINT : DISPLAYNOT	3
UTILITAIRE DE MISE AU POINT : TESTREBAS	4
UTILITAIRE DE MISE AU POINT : RECVNOT	5
RECEPTION DES NOTIFICATIONS DE CONNECT:EXPRESS	1
UTILISATION DU CLIENT CONNECT:EXPRESS WINDOWS	1
ANNEXE 1.....	1
STRUCTURE DE LA NOTIFICATION CONNECT:EXPRESS	1
ANNEXE 2.....	1
NOTIF.H	1

Préface

Ce document décrit la mise en œuvre de la fonction généralisée de notifications livrée avec le produit Connect:Express UNIX. Les notifications sont émises vers un outil d'administration qui est en charge du traitement de l'information. Cet outil peut être l'utilitaire de réception des notifications de Connect:Express Windows.

La fonction de notification est prise en charge par un utilitaire qui peut être configuré de multiples façons et intégré dans n'importe quel environnement applicatif.

Le chapitre 1 donne une description générale de l'utilitaire.

Le chapitre 2 décrit la mise en œuvre dans un environnement applicatif quelconque, y compris Connect:Express UNIX.

Le chapitre 3 décrit la mise en œuvre dans l'environnement spécifique de Connect:Express UNIX.

Le chapitre 4 décrit les éléments logiciels installés avec Connect:Express UNIX.

Le chapitre 5 présente un exemple de plate-forme d'administration.

Les annexes 1 et 2 décrivent la structure de la notification journal.

Ce document est destiné à un administrateur du produit Connect:Express, ayant déjà mis en œuvre les interfaces d'intégration de Connect:Express.

Introduction

L'administration des transferts de fichiers effectués par le moniteur Connect:Express peut s'inscrire dans un contexte plus large incluant les applications qui produisent et exploitent les données échangées. L'utilitaire livré avec le produit Connect:Express a donc été conçu pour une utilisation généralisée.

La remontée d'informations vers une plate forme d'administration concerne aussi bien les évènements propres aux traitements que des évènements liés à l'organisation de ces traitements. L'utilitaire décrit dans ce document prend en compte les choix multiples qu'on peut effectuer dans la nature des notifications et dans la façon de notifier l'administrateur:

Différents types de traitements peuvent être mis en œuvre :

- Traitement en tâche de fond
- Traitement par lot
- Traitement au fil de l'eau

Les fonctions proposées sont les suivantes :

- Envoi d'une notification au format Connect:Express
- Envoi du contenu d'un fichier
- Envoi d'un message
- Construction dynamique de la notification
- Sauvegarde/reprise
- Trace interne
- Interface de commandes Opérateur

L'agent peut être utilisé de la façon suivante :

- Exécution en mode batch
- Exécution en tâche de fond
- Exécution par Connect:Express en début ou en fin de transfert

Dans le cas particulier de l'environnement Connect:Express, la notification est construite à partir de l'enregistrement journal (fichier des « en cours »).

L'utilitaire construit un message unique - à partir d'un fichier, d'un texte, ou de l'enregistrement journal de Connect:Express – et l'envoie à l'adresse indiquée dans sa configuration.

Les communications avec la plate forme d'administration sont basées sur le réseau TCP/IP.

Un mécanisme de sauvegarde des notifications peut être mis en place afin de se protéger contre les défaillances des communications avec le serveur d'administration.

Les différentes fonctions sont décrites une à une dans la suite après une présentation générale des paramètres et des principes de fonctionnement de l'utilitaire.

Chapitre 1

Ce chapitre décrit l'ensemble des fonctionnalités de l'utilitaire, les paramètres de configuration et les messages émis pendant son fonctionnement.

L'Utilitaire de notification

La mise en oeuvre du flux des notifications liées aux transferts de fichiers conduit à faire des choix parmi les types de traitements, les fonctions supportées et les modes d'utilisation proposés.

Le paramétrage est basé sur l'utilisation conjointe des paramètres d'exécution et d'un fichier de configuration (par défaut notcfg.xml). Les paramètres d'exécution ont priorité sur les paramètres définis dans le fichier notcfg.xml. Le choix du mode de paramétrage dépend du mode d'utilisation choisi: batch ou tâche de fond.

La fonction de notification est déterminée par un paramètre '**msg**' auquel est associé l'ensemble des paramètres nécessaires à l'exécution de la fonction choisie.

Parmi ces paramètres, les options de trace et de sauvegarde/reprise après incident peuvent être activées.

Paramétrage de l'utilitaire

Il est possible de définir la configuration soit uniquement par les paramètres d'exécution, soit uniquement dans le fichier notcfg.xml, soit par une fusion des deux. Une fois l'initialisation terminée, l'absence d'un paramètre obligatoire se traduira par une interruption de l'utilitaire.

Utilisation des paramètres d'exécution:

```
$ ./tom_notify -msg J -srv 1.2.3.4 -prt 4000 -rec Y
```

Utilisation du fichier notcfg.xml (configuration pour l'envoi d'un texte):

```
<msg>T</msg>
<srv>10.20.30.40</srv>
<prt>5000</prt>
<tmr>300</tmr>
<occ>2</occ>
<txt>PROCEDURE XXX ENDED ON %DA AT %TI</txt>
<trc>Y</trc>
```

Chaque mode de fonctionnement nécessite un jeu précis de paramètres, dont certains peuvent être pris par défaut, d'autres sont obligatoires. Le contrôle de la configuration ne s'attache qu'à vérifier que tous les paramètres nécessaires à l'exécution de la fonction déterminée par l'option "**msg**" sont présents. Les paramètres inutiles sont ignorés. Si un paramètre est fourni en double, c'est la première valeur qui est enregistrée.

La liste des paramètres de configuration ci-dessous indique pour chaque mot clé, les différentes valeurs acceptées et leur signification. Les valeurs par défaut sont indiquées en tête et soulignées. Les combinaisons valides de paramètres sont indiquées aux chapitres 2 et 3 dans la description de chacun des modes de fonctionnement.

Liste des paramètres de configuration de l'utilitaire

Mot Clé	Valeur	Description
msg	F L T J	<p>Paramètre obligatoire</p> <p>Mode de fonctionnement choisi:</p> <ul style="list-style-type: none"> • F = Fichier: envoi du fichier indiqué par le paramètre fnm • L = Lot: envoi des fichiers du répertoire indiqué par le paramètre ind • T = Texte: envoi du message indiqué par le paramètre txt • J = Journal: utilisation de la fonction journal de Connect:Express UNIX <p>Chacune des fonctions est décrite dans les chapitres 2 et 3.</p>
srv	1 à 15 c.	<p>Paramètre obligatoire</p> <p>Adresse TCP/IP du serveur TCP/IP de la plate forme d'administration, de la forme habituelle, 4 nombres de 1 à 3 chiffres séparés par des points (nnn.nnn.nnn.nnn).</p> <p><i>Exemple: <src >0.20.30.40</src></i></p>
prt	1 à 5 c.	<p>Paramètre obligatoire</p> <p>Port d'écoute du serveur TCP/IP de la plate forme d'administration.</p> <p><i>Exemple: <prt>7500</prt></i></p>
txt	1 à 80 c.	<p>Paramètre obligatoire avec la fonction msg=T.</p> <p>Chaîne de caractères alphanumériques. Quand il est passé en paramètre de la commande et s'il contient des blancs, inclure entre doubles quotes. Le texte peut</p>

Mot Clé	Valeur	Description
		<p>contenir des variables qui seront résolues avant envoi, selon le contexte. En environnement Connect:Express (msg=J), le jeu des variables fournies permet de structurer des informations liées au transfert.</p> <p><i>Exemple :</i></p> <p>\$ tom_notify -msg J -txt "Le traitement %FILENAM a été terminé avec succès le %DA à %TI"</p>
ind	1 à 512 c.	<p>Paramètre obligatoire avec les fonctions msg=L.</p> <p>Répertoire contenant les fichiers à transmettre ou motif décrivant un ensemble de fichiers à transmettre.</p> <p><i>Exemples :</i> <ind>\$TOM_DIR/notif</ind> <ind>\$TOM_DIR/rci/N_*_*</ind></p>
rci	1 à 19 c.	<p>Paramètre obligatoire avec l'option rec=Y. (Sauf pour msg=F et msg=L)</p> <p>Paramètre utilisé pour la recovery avec les fonctions msg=J, msg=T et msg=F si rec=Y .</p> <p>Répertoire contenant les fichiers de sauvegarde.</p> <p><i>Exemple :</i> <rci>\$TOM_DIR/rci</rci></p> <p>IMPORTANT : Ce répertoire doit impérativement se situer sur le même système de fichier que Connect:Express.</p>
fmt		<p>Paramètre optionnel: valeur par défaut 'N'.</p> <p>Ce paramètre n'est pas utilisé.</p>
adm	1 à 8 c.	<p>Paramètre optionnel : Il est fixé par défaut à la valeur '<u>ADMIN</u>'.</p> <p>Nom logique de la plate forme d'administration destinataire des notifications.</p> <p><i>Exemple :</i> <adm>TRACK01</adm></p>
wai	1 à 3 c.	<p>Paramètre optionnel, utilisé avec la fonction msg=L.</p> <p>Caractères numériques indiquant le nombre de minutes entre deux traitements. Ce paramètre est utilisé dans le traitement par lot (msg=L) et détermine un traitement par lot en tâche de fond. Ce paramètre vaut 0 par défaut.</p> <p><i>Exemple :</i> <wai>10</wai></p>
rec	<u>Y</u> N	<p>Paramètre optionnel: valeur par défaut 'Y'.</p> <ul style="list-style-type: none"> • Yes: Conservation de la notification pour ré-essai ultérieur (le paramètre rci est alors obligatoire si msg=J ou msg=T). • No: pas de recovery.

del	<u>N</u> Y	<p>Paramètre optionnel: valeur par défaut 'N'. (msg=F)</p> <ul style="list-style-type: none"> • Yes: Si la notification est envoyée avec succès, le fichier émis est supprimé automatiquement. Si l'envoi de la notification échoue, le fichier n'est supprimé que si rec=Y. • No: Le fichier n'est pas supprimé par tom_notify
trc	<u>N</u> Y	<p>Paramètre optionnel: valeur par défaut 'N'.</p> <ul style="list-style-type: none"> • No: pas de trace. • Yes: Activation de la trace interne <p>Note: Le paramètre STRACE=1 du fichier config/sysin de Connect:Express conditionne aussi la sortie d'éléments supplémentaires de traces.</p>
fnm	1 à 512 c.	<p>Paramètre optionnel: utilisé avec l'option msg=F.</p> <p>Nom complet du fichier transmis avec l'option msg=F.</p> <p><i>Exemple:</i> <fnm>\$TOM_DIR/sdir/filename.txt</fnm></p>
occ	<u>3</u> 1 à 3 c.	<p>Paramètre optionnel: valeur par défaut 3.</p> <p>Caractères numériques indiquant le nombre de tentatives de connexion avec le serveur TCP/IP de la plate forme d'administration avant d'abandonner l'émission de la notification.</p> <p><i>Exemple:</i> <occ>5</occ></p>
tmr	<u>100</u> 1 à 3 c.	<p>Paramètre optionnel: valeur par défaut 100 (1 seconde)</p> <p>Caractères numériques indiquant le nombre de secondes entre deux tentatives de connexion avec le serveur TCP/IP de la plate forme d'administration.</p> <p><i>Exemple:</i> -tmr 5</p>
que	<u>Num</u>	<p>Paramètre optionnel: utilisé avec l'option msg=L. Valeur par défaut 0.</p> <p>Dans le cas où tom_notify est lancé en tâche de fond ce paramètre permet d'indiquer le n° d'une message queue IPC associée à la tâche. Ce n° de message queue doit être indiqué dans la commande notcmd stop lors de l'arrêt du process tom_notify.</p> <p>La valeur par défaut est 0. S'il n'y a qu'un seul process tom_notify lancé en tâche de fond, il n'est par conséquent pas nécessaire d'utiliser ce paramètre.</p> <p><i>Exemple:</i> <que>0</que></p>
ati	<u>HHMM</u>	<p>Paramètre optionnel: utilisé par Connect:Express. Valeur par défaut 0030.</p> <p>Ce paramètre indique l'heure d'archivage du fichier log des notifications notif.log par le processus archnotlog lancé en tâche de fond par Connect:Express.</p> <p><i>Exemple:</i> <ati>2330</ati></p>

Logging des process tom_notify

Les différents processus tom_notify effectuent leur logging dans un fichier commun \$TOM_DIR/notif/notif.log.

Chaque ligne du fichier log indique après la date, le process id du processus tom_notify ayant généré le message. Les différents paramètres de chaque lancement de tom_notify sont indiqués.

```
0506131609 5996 -msg J -fnm N_J_E16400004
0506131609 5996 Notification N_J_E16400004 sent successfully
0506131609 5996 File deleted /u1/pgs/TOM140/tom1/notif/N_J_E16400004
0506131609 5998 -msg J -fnm N_J_E16400003
0506131609 5998 Notification N_J_E16400003 sent successfully
0506131609 5998 File deleted /u1/pgs/TOM140/tom1/notif/N_J_E16400003
0506131610 6012 -msg J -fnm N_J_E16400006
0506131610 6012 Connection refused
0506131610 6012 Error connecting to server. errno = 111
0506131610 6014 -msg J -fnm N_J_E16400005
0506131610 6014 Connection refused
0506131610 6014 Error connecting to server. errno = 111
0506131611 6017 -msg L -ind ../rci/N_*
0506131611 6017 File sent successfully ../rci/N_J_E16400006
0506131611 6017 File deleted ../rci/N_J_E16400006
0506131611 6017 File sent successfully ../rci/N_J_E16400005
0506131611 6017 File deleted ../rci/N_J_E16400005
0506131611 6017 2 file(s) sent
```

Commandes opérateur

Lorsque l'utilitaire est lancé en tâche de fond (msg=L avec wai non nul), il peut recevoir des commandes de l'opérateur.

Format général

Le format général d'une commande est le suivant:

```
$ ./notcmd commande [n°-de-message-queue]
```

Le n° de message queue correspond au numéro de message queue du processus tom_notify auquel on s'adresse.

Note: S'il n'y a qu'un seul processus tom_notify lancé en tâche de fond, ce paramètre peut être omis au lancement de tom_notify et lors de la commande notcmd (Dans ce cas la message queue 0 est utilisée par défaut).

Demande d'arrêt

L'arrêt peut être demandé par la commande:

```
$ ./notcmd stop
```

Si une demande d'arrêt est reçue en cours de traitement, elle ne provoque pas l'interruption du traitement mais l'utilitaire s'arrêtera une fois le traitement courant terminé.

Traitement immédiat

La commande suivante est utilisée pour l'activation immédiate du traitement:

```
$ ./notcmd send
```

Cette commande permet de déclencher la fonction de traitement par lot sans attendre le déclenchement automatique.

Activation/Désactivation de la trace

Les commandes suivantes permettent d'activer/désactiver la trace dans notif.log :

```
$ ./notcmd traceon  
$ ./notcmd traceoff
```

Fonction trace intégrée

Le paramètre `trc=Y` active une trace interne qui se matérialise par des messages enregistrés dans le fichier `notif.log`.

Fonction de sauvegarde

Une notification peut être construite à partir d'un fichier (`msg=F`, `msg=L`), d'un message (`msg=T`) ou d'une information éphémère (`msg=J`).

Msg = J:

Les notifications de type journal (`msg=J`) sont enregistrées dans le répertoire `$TOM_DIR/notif`, avant envoi vers la plate-forme d'administration, dans des fichiers dont le nom peut avoir l'une des formes suivantes

`msg=J` et paramètre `txt` non présent: `N_J_x<n°de requête>`

`msg=J` et paramètre `txt` présent: `N_JT_x<n°de requête>`

où `x` peut prendre les valeurs `I`, `E`, `R` suivant qu'il s'agit d'une notification de début de transfert, de fin de transfert ou d'erreur de transfert.

Par exemple:

- `$TOM_DIR/notif/N_J_E16400003` indique une notification au format Journal Connect:Express pour la fin du transfert de la requête n° 16400003.
- `$TOM_DIR/notif/N_JT_E16400003` indique une notification au format texte avec substitution de variables du journal de Connect:Express pour la fin du transfert de la requête n° 16400003.

Dans le fonctionnement normal, une fois la notification envoyée à la plate forme d'administration, l'information d'origine disparaît.

Msg = T:

Les notifications de type `T`, sont sauvegardées dans le répertoire défini par `rci` si l'option `rec=Y` dans des fichiers ayant un nom formé de la manière suivante:

`N_T_AAAAMMJJHHMMSS_<pid>`

Où `pid` est le process-id du processus `tom_notify`.

Msg = F:

Les nom des fichiers associés aux notifications sont ceux des fichiers à émettre choisis par l'utilisateur.

Ces fichiers sont supprimés après envoi de la notification si `del=Y`.

Si il y a erreur dans l'envoi de la notification:

- le fichier reste inchangé si `del=N`
- le fichier est déplacé vers le répertoire `rci`, si `rec=Y`, en prenant le nouveau nom `N_F_AAAAMMJJHHMMSS_<pid>` où `pid` est le process-id du processus `tom_notify` effectuant l'opération

Msg = L:

Les fichiers dont la notification est envoyée sans erreur sont systématiquement supprimés.

Les fichiers dont la notification n'a pu être envoyée sont conservés pour réémission ultérieure si `rec=Y` et supprimés sinon.

Conservation des notifications

`rec=Y`

En cas de dysfonctionnement, les notifications peuvent être conservées pour réémission ultérieure. Soit le fichier d'origine est conservé (`msg=F`, `msg=L`), soit la notification est sauvegardée dans un fichier de sauvegarde (`msg=J`).

Cette fonction peut être désactivée par l'option `rec=N`.

Dans le cas de la sauvegarde (`msg=J` ou `F`), le fichier qui n'a pu être envoyé est déplacé vers le répertoire indiqué par le paramètre `rci`.

Traitement de la reprise

Le traitement de reprise est effectué automatiquement et de manière répétitive dans le mode fonctionnant en tâche de fond (`msg=L` avec `wai=` non nul). La reprise consiste à effectuer un traitement par lot, soit sur des fichiers d'origine conservés, soit sur des fichiers de sauvegarde créés automatiquement. L'intervalle entre 2 traitements successifs étant alors de `<wai>` minutes.

Si le traitement par lot ne s'effectue pas en tâche de fond, il est possible à tout moment de lancer un traitement par lot (`msg=L`, `wai=0`) qui ne s'exécutera qu'une seule fois.

Le paramètre `ind` permet d'indiquer un motif simple permettant la sélection d'un ensemble de fichiers du répertoire `rci`:

Par exemple `-ind $TOM_DIR/rci/N_*_164*`, permettra de n'envoyer que les notifications des requêtes du quantième 164.

La mise en oeuvre de la reprise est indiquée au cas par cas aux chapitres 2 et 3 dans la description de chacun des modes de fonctionnement.

Ré-émission des Fichiers d'origine (`msg=F`)

Si l'on a paramétré `del=N`, les notifications journal ne sont pas déplacées dans le répertoire de recovery. Elles ne sont pas par ailleurs supprimées du répertoire notif.

Il est donc possible de les rémettre par un traitement par lot, avec un filtre adéquat:

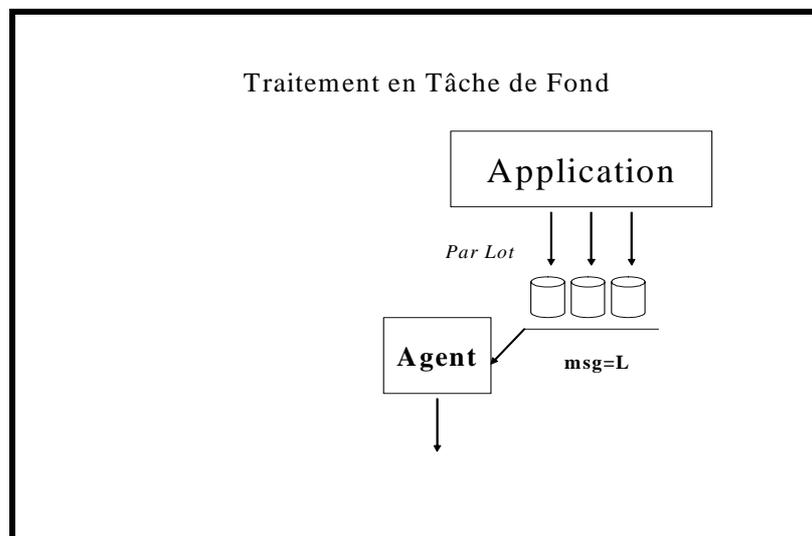
```
./tom_notify -msg L -ind $TOM_DIR/sdir/file???.txt    (pour un groupe de fichier)
./tom_notify -msg L -ind $TOM_DIR/sdir/file.txt       (pour un fichier unique)
```

Différents types de traitement

L'utilitaire peut s'adapter aux différentes situations engendrées par des environnements applicatifs variés. Suivant la fonction utilisée, il peut être nécessaire de passer par un fichier intermédiaire.

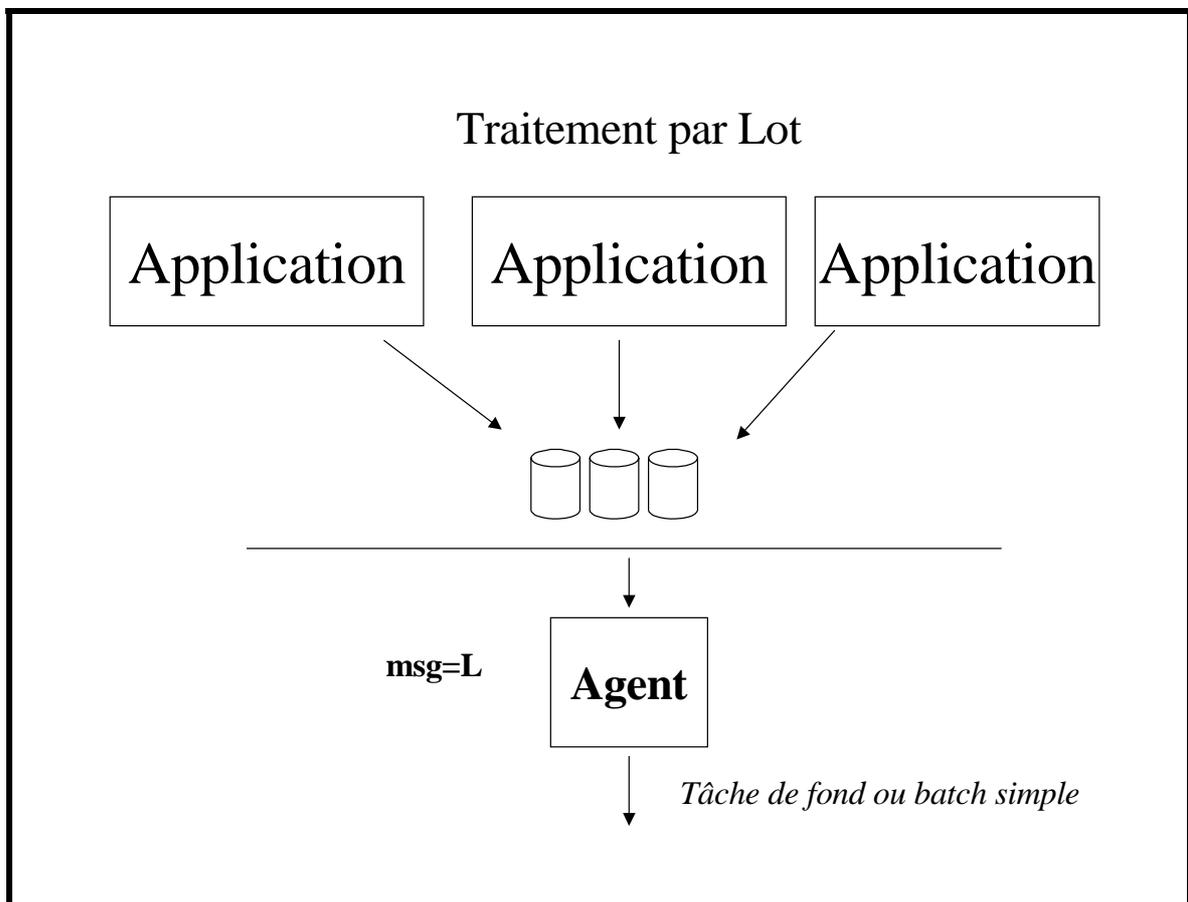
Traitement en tâche de Fond

Lancé en tâche de fond, l'utilitaire permet le traitement périodique d'un lot de fichiers d'origine quelconque (option msg=L avec wai non nul). La reprise après échec est gérée automatiquement ou peut être déclenchée par l'opérateur.



Traitement par lots

Le traitement par lot permet de traiter les informations mises à disposition par les applications. Il peut être lancé successivement pour un seul traitement ou en tâche de fond activée régulièrement. Les noms des fichiers à traiter sont caractérisés par un répertoire commun. Ce mode de traitement peut être utilisé pour mettre en place une procédure de reprise après interruption des communications avec la plate forme d'administration.

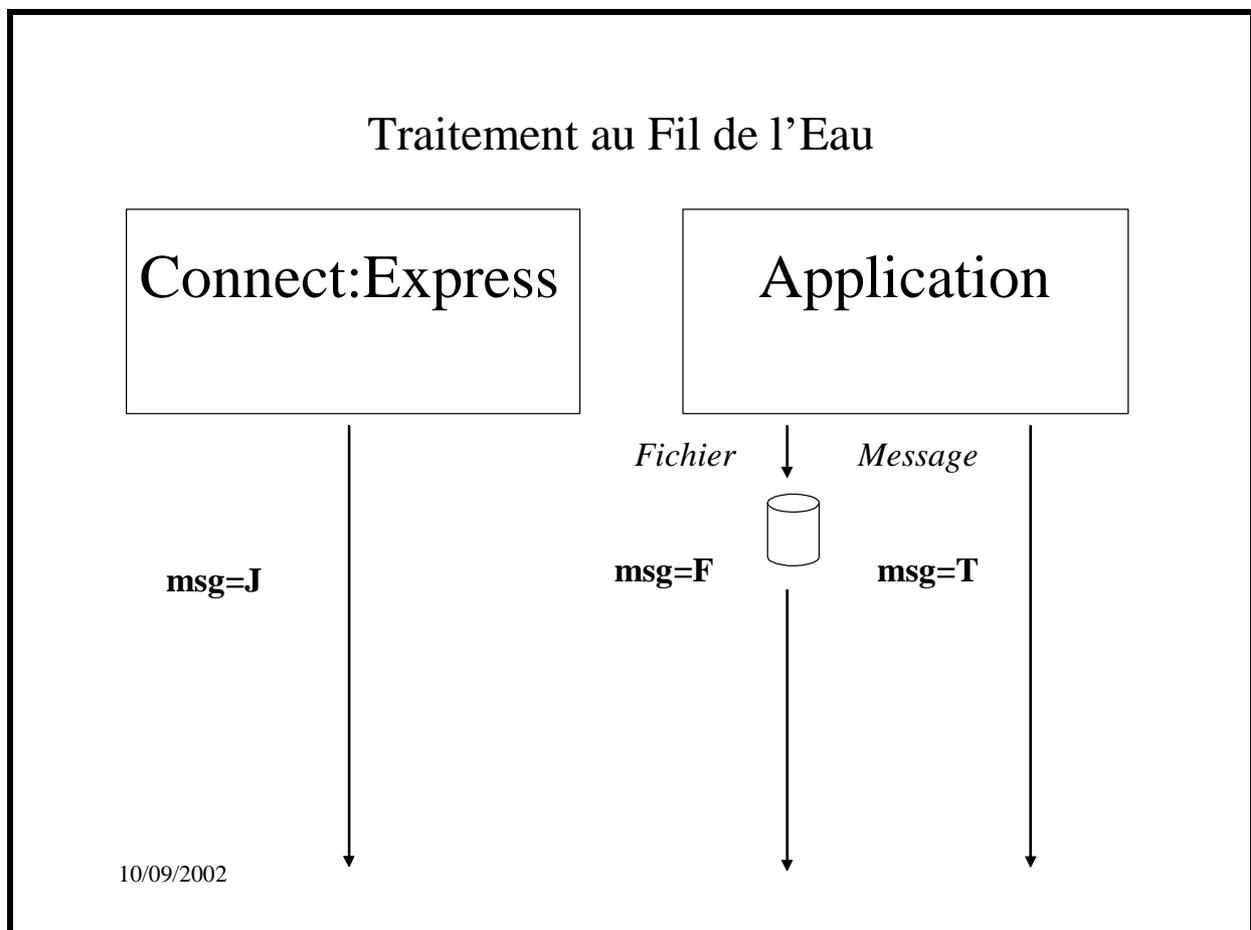


Traitement au fil de l'eau

Envoyer les notifications au fur et à mesure de leur création peut être réalisé en intégrant l'utilitaire dans le processus applicatif. En environnement Connect:Express, l'utilitaire peut être déclenché en début, en fin de transfert ou en cas d'erreur afin d'envoyer une notification journal du transfert.

Dans un contexte applicatif général (l'applicatif pouvant être Connect:Express lui-même) l'utilitaire peut être intégré dans une procédure pour envoyer un fichier ou un message.

L'utilitaire peut aussi être intégré dans un programme utilisateur pour envoyer un fichier ou un message.



Différents modes d'utilisation

L'utilitaire peut s'intégrer dans différents types de processus.

Utilisation en batch / à partir de la ligne de commande

Le paramétrage peut être complètement assuré par un fichier notcfg.xml. Cette utilisation convient dans un enchaînement de tâches: tom_notify envoie un fichier préparé par une étape précédente.

Utilisation depuis un programme C

L'agent peut être appelé depuis un programme utilisateur C. Les fonctions msg=F, msg=T et msg=L/wai=0 sont les seules adaptées à ce mode d'utilisation. Les paramètres d'exécution complémentaires sont issus du fichier notcfg.xml.

On peut utiliser la fonction system :

*int system(const char *string)*

en lui fournissant en paramètre la commande tom_notify à exécuter. On peut également lancer un script shell qui lui-même lancera tom_notify

Par exemple:

```
int ret;
char command[] = "./tom_notify -msg T -txt \"Send this text\"";          /* Execution synchrone */

if (system(command)==-1) {
    perror("");
    printf("Error executing command %s\n",command);
    ....
}
```

On peut également utiliser les appels systèmes fork puis execlp, pour lancer la commande de manière asynchrone.

Utilisation dans une commande de début, fin ou erreur de transfert

Une commande associée au transfert permet l'envoi d'un fichier (msg=F) et l'envoi d'un texte (msg=T). Ces fonctions sont décrites au chapitre 2.

Chapitre 2

Ce chapitre présente les modes d'utilisation qui peuvent être mis en oeuvre quelque soit l'environnement.

Utilisation en environnement applicatif

Les fonctions d'envoi d'un ou de plusieurs fichiers ou d'un message peuvent être mises en oeuvre en dehors de l'environnement de Connect:Express. Connect:Express peut lui même être considéré comme toute autre application et les paramètres indiqués dans ce chapitre peuvent être pris en compte dans l'environnement du moniteur (commandes de début et fin de transfert par exemple).

Envoi d'un fichier

(msg=F)

L'utilitaire permet l'envoi d'un fichier unique.

Paramétrage

La liste type des paramètres a la forme suivante:

```
$ tom_notify -msg F -srv 1.2.3.4 -prt 4000 -occ 5 -tmr 100 \  
> -fnm $TOM_DIR/sd/f.txt -del Y -rec Y -rci $TOM_DIR/rci
```

Le paramètre msg=F est caractéristique de ce type de fonctionnement. Les paramètres srv et prt sont obligatoires, les paramètres occ et tmr sont optionnels.

Si le paramètre del=Y et si la notification a été envoyée avec succès, le fichier est automatiquement supprimé suite à l'envoi de celle-ci. En cas d'échec, il n'est conservé que si rec=Y.

Remarque: Les fichiers sont envoyés en format binaire. Si le fichier émis est un fichier texte unix, c'est à la plate-forme d'administration de traiter les différents séparateurs LF des données reçues. Par ailleurs si l'application réceptrice ne s'attend qu'à recevoir des fichiers textes, il faut veiller à ce que seuls des fichiers texte d'un répertoire soient sélectionnés lors de l'utilisation en mode msg=L

Utilisation du fichier notcfg.xml (Configuration pour l'envoi d'un fichier) :

```
<msg>F</msg>
<srv>1.2.3.4</srv>
<prt>5000</prt>
<rci>${TOM_DIR}/rci</rci>
<rec>Y</rec>
<del>N</del>
<tmr>300</tmr>
<occ>2</occ>
```

Le fichier à envoyer est indiqué dans la commande :

```
$ tom_notify -fnm ${TOM_DIR}/sdir/f.txt
```

Dans l'exemple ci-dessus, les paramètres nécessaires non fournis dans la commande sont soit pris dans le fichier notcfg.xml, soit pris par défaut.

Les données du fichier indiqué seront lues et constitueront une notification envoyée à l'adresse indiquée dans les paramètres.

Gestion des echecs

Si il y a erreur dans l'envoi de la notification:

- le fichier reste inchangé si del=N
- le fichier est déplacé vers le répertoire rci si rec=Y et prend le nouveau nom N_F_AAAAMMJJHHMMSS_<pid> où pid est le process-id du processus tom_notify effectuant l'opération

Envoi d'un lot de fichiers

(msg=L)

L'utilitaire permet l'envoi par lot des fichiers situés dans un répertoire donné. Il peut être lancé pour un seul traitement ou lancé en tâche de fond pour un traitement périodique.

Paramétrage

La liste type des paramètres a l'une des formes suivantes :

```
$ ./tom_notify -msg L -srv 1.2.3.4 -prt 4000 -tmr=100 -wai=5 -ind $DIRECTORY/sdir -rec Y
```

```
$ ./tom_notify -msg L -srv 1.2.3.4 -prt 4000 -tmr=100 -wai=5 -ind $DIRECTORY/sdir/<expression> -rec Y
```

Dans la dernière forme ci-dessus, <expression> définit un motif simple permettant de caractériser un sous-ensemble des fichiers du répertoire sdir.

Par exemple `-ind $DIRECTORY/sdir/N*` correspondra à l'émission de tous les fichiers dont le nom commence par N dans le répertoire sdir.

Les paramètres `msg=L`, `ind` et `wai` sont caractéristiques de ce type de fonctionnement. Les paramètres `ind`, `srv` et `prt` sont obligatoires, les paramètres `occ` et `tmr` sont optionnels. Le paramètre `wai` est optionnel. S'il n'est pas fourni, l'utilitaire se lance, traite les fichiers trouvés et s'arrête.

Si le paramètre `wai` est fourni, non nul, l'utilitaire reste en mémoire jusqu'à ce que l'opérateur lui passe une commande `notcmd` d'arrêt. Il traite tous les fichiers présents, tous les intervalles de temps indiqués par ce paramètre.

Une fois l'utilitaire lancé, si le paramètre `wai` est fourni et non nul, il est possible de lui passer l'une des commandes opérateur décrites au chapitre 1.

Si l'on veut lancer plusieurs processus `tom_notify` en tâche de fond, agissant chacun sur des répertoires différents, il est nécessaire d'ajouter en paramètre de lancement supplémentaire, un n° de message queue (`-que`) afin que la commande d'arrêt `notcmd` puisse distinguer les `tom_notify` à laquelle elle s'adresse. Si un seul processus seulement doit rester en mémoire, l'utilisation du paramètre `-que` n'est pas nécessaire.

Par exemple:

```
./tom_notify -msg L -wai 10 -ind $DIRECTORY1 -que 1
./tom_notify -msg L -wai 20 -ind $DIRECTORY2 -que 2
./notcmd stop 2
```

pour lancer le 1er process `tom_notify`
pour lancer le 2ème process `tom_notify`
pour arrêter le 2ème process `tom_notify`

Remarque: Il est nécessaire que 2 processus `tom_notify` lancés en tâche de fond opèrent sur des répertoires différents.

Utilisation du fichier notcfg.xml (Configuration pour l'envoi d'un lot de fichiers):

```
<msg>L</msg>
<srv>1.2.3.4</srv>
<prt>5000</prt>
<ind>${DIR}/sdir</ind>
<tmr>300</tmr>
<rec>Y</rec>
<occ>2</occ>
<wai>10</wai>
```

Gestion des echecs

L'utilitaire traite tous les fichiers sélectionnés à partir du répertoire ind. En cas d'échec le processus continue jusqu'à la fin de la liste des fichiers trouvés.

Les fichiers correctement émis sont supprimés.

Un fichier qui n'a pu être envoyé n'est conservé que si rec=Y. Pour effectuer une nouvelle tentative, il suffit dans ce cas de relancer le traitement. Lorsque le traitement est permanent (wait non nul), le mécanisme de reprise est effectué systématiquement.

Envoi d'un texte simple

(msg=T)

L'utilitaire permet la construction dynamique d'un message de notification à envoyer. Le message peut être traité par lui même (Option msg=T et paramètre txt= message à envoyer) ou par la combinaison d'un texte (txt=) et des fonctions intégrées à Connect:Express (msg=J). Ce dernier cas d'utilisation sera décrits au chapitre 3.

La taille maximale d'un texte est de 512 caractères.

Paramétrage

La liste type des paramètres a la forme suivante:

```
$ ./tom_notify -msg T -srv 1.2.3.4 -prt 4000 -occ 5 -rec Y \
> -tmr 100 -txt "message" -rci "répertoire des fichiers de sauvegarde"
```

Les paramètres msg=T et txt sont caractéristiques de ce type de fonctionnement. Les paramètres txt, srv et prt sont obligatoires, les paramètres rec, occ et tmr sont optionnels. Si le paramètre rec est 'Y', alors le paramètre rci est obligatoire.

En fonction de la taille du texte à émettre, il peut être préférable d'utiliser le paramètre <txt> du fichier notcfg.xml ou pas.

Utilisation du fichier notcfg.xml (Configuration pour l'envoi d'un texte non associé au journal de Connect:Express):

```
<msg>T</msg>
<txt>Message à envoyer</msg>
<srv>1.2.3.4</msg>
<prt>5000</msg>
<tmr>300</msg>
<occ>2</msg>
<rec>Y</msg>
<del>Y</del>
<rci>répertoire-des-fichiers-de-sauvegarde</msg>
```

Mots Clés dynamiques

L'inclusion de variables symboliques dans le corps du message, permet d'envoyer des informations contextuelles.

Variable	Taille	Description
%DATE ou %DA	10 c.	Le champ sera résolu avec la date système, de la forme AAAA/MM/JJ
%TIME ou %TI	8 c.	Le champ sera résolu avec l'heure système, de la forme HH :MM :SS

Gestion des Echecs

Selon la nature des informations, les échecs peuvent être gérés de différentes façons : pour une information sans données variables la reprise peut consister à relancer le traitement. Si les données contiennent des informations contextuelles, il est nécessaire de sauvegarder la notification en l'état pour un nouvel essai.

En cas d'échec pendant l'envoi de la notification, si l'option de reprise est active (rec=Y), le message est sauvegardé dans un fichier. Pour effectuer une nouvelle tentative, il suffit de lancer un traitement par lot (msg=L) avec le paramètre ind équivalent au paramètre rci du batch initial.

Note : Les textes sont sauvegardés pour réémission dans le répertoire rci sous le nom N_T_AAAAMMJJHHMMSS_<pid> où pid est le process-id du tom_notify effectuant l'opération.

Chapitre 3

Dans le contexte spécifique de Connect:Express, il est possible de produire des notifications exploitables par des outils existants.

Notifications de Connect:Express

La mise en œuvre de tom_notify dans le contexte de Connect:Express peut être faite comme pour n'importe quelle application. Mais s'ajoute l'utilisation de la fonction journal (msg=J).

La notification envoyée peut être dirigée vers l'outil de réception des notifications de Connect:Express Windows ou vers une plate-forme d'administration.

Utilisation de la fonction journal

(msg=J et txt non renseigné)

L'utilitaire permet l'envoi d'une notification construite à partir de l'enregistrement journal de Connect:Express (Enregistrement du fichier RENC des "en cours").

Pour un transfert donné, les notifications journal peuvent être envoyées

- en début de transfert,
- en fin de transfert,
- en début et en fin d'un transfert,
- en cas d'erreur seulement
- pas du tout

selon configuration.

SYSIN

Le paramètre NOTIFY du fichier sysin de Connect:Express permet d'autoriser (NOTIFY=1) ou d'interdire (NOTIFY=0 ou non présent), globalement toute notification journal pour Connect:Express.

\$TOM_DIR/config/sysin

<pre> ... NOTIFY=1 * NOTIFY OPTION (1:ON 0:OFF) ... </pre>

Les décisions concernant l'envoi d'une notification pour un transfert donné se font dans la définition du fichier symbolique et/ou dans les paramètres du dépôt de la requête proprement dite.

Le paramétrage des notifications pour un fichier symbolique avec STERM s'effectue en valorisant le champ NOT avec l'une des valeurs ci-dessous:

- 0: pas de notification
- 1: notification en début de transfert
- 2: notification en fin de transfert
- 3: notification en début et en fin de transfert
- 4: notification en erreur
- 5: notification en début et en erreur de transfert
- 6: notification en fin et en erreur de transfert
- 7: notification en début, en fin et en erreur de transfert

Si une valeur est mentionnée lors du dépôt de la requête, celle ci est prise en compte pour la requête, sinon c'est la valeur définie dans le fichier symbolique qui est utilisée par défaut.

Fichier symbolique (STERM)

```

C:E UNIX 146-1 ----- FILES DIRECTORY ----- tom1
OPTION ==>
SYMBOLIC NAME : FILE03
INITIALIZATION STATUS : E E:ENABLE H:DISABLE
DIRECTION : R T:TRANSMIT R:RECEIVE *:EITHER
RECEIVING PARTNER : $$ALL$$ 'NAME',#LISTE, $$ALL$$
TRANSMITTING PARTNER : $$ALL$$ '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 : $TOM_DIR/in/&REQNUMB.tmp
RECORD FORMAT : BU TF, TV, BF, BU, T*, B*, **
RECORD LENGTH : 04000 1-5 NUMERIC CHARAC.
REMOTE DSN (FTP) :
TYPE/STRUCTURE/MODE FTP : E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) : Y/N FA : N Y/N NOT: 2 (0-7)
OPTION : VIEW UPD : 05/10/12 14:59 mverz1
-ENTER- NEXT FIELD -F3- CANCEL -F8- COMPLETION

```

Requête de transfert (STERM)

```

C:E/UNIX 146-1 ----- TRANSFER REQUEST ----- tom1
OPTION ==>
FILE : FILE03 DIRECTION : T (T/R)
PARTNER : PARTNER.
DPCSID ALIAS : MYNAME.. DPCPSW ALIAS : MYPASSW.
ORIGIN : DESTINATION :
SENDER : RECEIVER :
PHYSICAL NAME : $TOM_DIR/out/send.txt
REMOTE PHYSICAL NAME :
LABEL:
RECORD FORMAT : BU TF, TV, BF, BU
RECORD LENGTH : 04000
TYPE/STRUCTURE/MODE FTP : *** E/A/I/*,F/R/*,B/S/*
STORE UNIQUE (FTP) : N Y/N FA : Y/N NOT: 2 (0-7)
TYPE : N (N/I/H)
TYPE OF CONNECTION : T (X/P/T)
PRIORITY : 0 (0/1/2)
DATE : 19980728101604 (YYYYMMDDHHMMSS)
API FIELD (ETEBAC3 : 80 CHARACTERS FOR CARD)
1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0
DO YOU WANT TO GO ON ?
-ENTER- NEXT FIELD -F3- CANCEL -F8- COMPLETION

```

Requête de transfert (p1b8preq)

Lors du dépôt d'une requête par p1b8preq, il est possible de mentionner les caractéristiques de notification pour le transfert en utilisant /NTF=x (x = 0-7) dans le premier ensemble de paramètres de la commande.

Par exemple:

```
$TOM_DIR/itom/plb8preq "/SFN=FILE03/DIR=T/SPN=PARTNER/SID=MYNAME/PWD=MYPASSW/NTF=2" \
"/DSN=\$TOM_DIR/out/send.txt"
```

L'utilitaire est lancé automatiquement en début, en fin de transfert ou en cas d'erreur.

L'enregistrement journal correspondant au transfert est transformé en une notification au format standard commun à toutes les plates-formes Connect:Express et envoyé à la plate forme d'administration.

Paramétrage

La liste type des paramètres a la forme suivante:

```
$ tom_notify -msg J -srv 1.2.3.4 -prt 000' -occ 5 -tmr 100 -rec Y -rci
\$TOM_DIR/recov
```

Le paramètre msg=J est caractéristique de ce type de fonctionnement. Les paramètres srv et prt sont obligatoires, les paramètres rec , occ et tmr sont optionnels. Si rec=Y, le paramètre rci est obligatoire.

L'ajout d'un paramètre txt permet d'envoyer un message construit à partir de l'enregistrement Journal. Cette utilisation est décrite dans les paragraphes suivants.

Utilisation du fichier notcfg.xml (Configuration pour l'envoi d'une notification journal) :

```
<msg>J</msg>
<srv>1.2.3.4</srv>
<prt>5000</prt>
<tmr>300</tmr>
<occ>2</occ>
<rec>Y</rec>
<rci> 'répertoire des fichiers de sauvegarde'</rci>
```

Les notifications de type journal (msg=J avec txt non renseigné) sont enregistrées dans le répertoire \$TOM_DIR/notif, avant envoi vers la plate-forme d'administration, dans des fichiers ayant un nom de la forme suivante :

N_J_x<n° de requête>

où x peut prendre les valeurs I, E, R suivant qu'il s'agit d'une notification de début de transfert, de fin de transfert ou d'erreur de transfert.

Par exemple:

- \$TOM_DIR/notif/N_J_E16400003 indique une notification au format Journal Connect:Express pour la fin du transfert de la requête n° 16400003.

Gestion des echecs

Dans le fonctionnement normal, une fois la notification envoyée à la plate forme d'administration, l'information d'origine disparaît.

En cas d'échec pendant l'envoi de la notification, si l'option de reprise est active (rec=Y), le message est sauvegardé dans un fichier placé dans le répertoire rci.

Lorsque le service est rétabli et si un traitement par lot périodique sur le répertoire rci a été lancé en tâche de fond, les fichiers sauvegardés sont envoyés automatiquement.

Cependant il est possible de passer à l'utilitaire une commande de reprise :

\$ notcmd SEND

Cette commande provoque le traitement par lot des fichiers sauvegardés.

Construction dynamique d'une notification

(msg=J et txt renseigné)

L'utilitaire permet la construction dynamique d'un message de notification à envoyer. Le message peut être traité par la combinaison d'un texte (txt=) et de l'utilisation de la fonction journal de Connect:Express (msg=J).

Paramétrage

```
$ tom_notify -msg J -srv 1.2.3.4 -prt 4000 \  
> -occ 5 -tmr 100 -rec Y -rci $TOM_DIR/recov -txt "texte"
```

Les paramètres msg=J et txt sont caractéristiques de ce type de fonctionnement. Les paramètres srv et prt sont obligatoires, les paramètres rec, occ et tmr sont optionnels. Si rec=Y, le paramètre rci est obligatoire.

La taille maximale du message est de 512 caractères après résolution.

Les variables contenues dans le message sont résolues à partir de l'enregistrement journal de Connect:Express.

Compte tenu de la taille du message, il peut être nécessaire d'utiliser le paramètre <txt> du fichier notcfg.xml :

```
<msg>J  
<txt>'texte contenant des variables à résoudre'</txt>  
<srv>1.2.3.4</srv>  
<prt>5000</prt>  
<tmr>300</tmr>  
<occ>2</occ>  
<rec>Y</rec>  
<rci> 'répertoire des fichiers de sauvegarde'</rci>
```

Note: Si le texte contenu entre les tags <txt> et </txt> contient des line feeds (LF) (Séparateur de texte), ceux-ci sont transmis dans la notification.

Variables dynamiques

L'inclusion de variables symboliques dans le corps du message, permet d'envoyer des informations contextuelles liées au transfert de fichier courant.

Chaque variable est représentée par un mot clé long et un mot clé court.

Variable	Taille	Description
%DATE ou %DA	10 c.	Date système, de la forme AAAA/MM/JJ
%TIME ou %TI	8 c.	Heure système, de la forme HH:MM:SS
%PARTNID ou %PA	8 c.	Nom symbolique de partenaire
%FILENAM ou %FI	8 c.	Nom symbolique de fichier
%REQNUMB ou %RQ	8 c.	Numéro de requête de transfert
%TRC ou %TC	4 c.	Code retour Connect:Express
%PRC ou %PC	4 c.	Code retour Connect:Express
%SRC ou %SC	4 c.	Code retour Connect:Express

Variable	Taille	Description
%ORG ou %OR	8 c.	Origine du transfert
%DST ou %DS	8 c.	Destination du transfert
%PHYNAM ou %PN	8 c.	Nom physique du fichier
%BEGDAT ou %BD	10 c.	Date de début de transfert, de la forme AAAA/MM/JJ
%BEGTIM ou %BT	8 c.	Heure de début de transfert, de la forme HH:MM:SS
%ENDDAT ou %ED	10 c.	Date de fin de transfert, de la forme AAAA/MM/JJ
%ENDTIM ou %ET	8 c.	Heure de fin de transfert, de la forme HH:MM:SS
%MONID ou %MN	8 c.	Identifiant du moniteur Connect:Express effectuant la notification (Paramètre DPCSID du fichier sysin)
%UDATS ou %US	254 c.	Données utilisateur émises (Pi99)
%UDATR ou %UR	254 c.	Données utilisateur reçues (Pi99)
%PI61	24 c.	Emetteur du transfert
%PI62	24 c.	Recepteur du transfert

Gestion des echecs

En cas d'échec pendant l'envoi de la notification, si l'option de reprise est active (rec=Y), un fichier contenant la notification après substitution est créé dans le répertoire rci avec le nom :

N_JT_x<n°de requête>

où x peut prendre les valeurs I, E, R suivant qu'il s'agit d'une notification de début de transfert, de fin de transfert ou d'erreur de transfert.

Par exemple:

- \$TOM_DIR/rci/N_JT_E16400003 indique une notification au format texte avec substitution de variables du journal de Connect:Express pour la fin du transfert de la requête n° 16400003.

La reprise peut être faite automatiquement ou manuellement par un traitement par lot sur le répertoire rci.

Chapitre 4

Ce chapitre décrit les éléments de Connect:Express installés concernant les notifications ainsi que quelques outils d'aide à la mise en oeuvre.

Installation et utilitaires de mise au point

Une fois installé Connect:Express, tous les objets nécessaires à la mise en oeuvre de la fonction de notification sont disponibles. Il sera nécessaire de personnaliser l'environnement et de prendre en compte les fichiers de travail créés par le logiciel. Plusieurs utilitaires sont fournis afin de faciliter la mise au point de la génération des notifications, indépendamment de la mise en place du serveur destiné à traiter ces notifications.

Elements installés

Les éléments concernant les notifications sont tous situés à l'installation de Connect:Express dans le répertoire \$TOM_DIR/notif.

Il s'agit de :

tom_notify	Exécutable effectuant les notifications
notcmd	Commande de gestion des tom_notify lancés en tâche de fond
archnotlog	Exécutable d'archivage du fichier notif.log
displaynot	Utilitaire de mise au point permettant l'affichage d'une notification au format journal Connect:Express
recvnot	Utilitaire de mise au point permettant de tester sur un système UNIX la réception d'une notification émise par un Connect:Express distant
start_archnotlog.sh	Script shell utilisé par Connect:Express à son démarrage pour lancer le process d'archivage archnotlog en tâche de fond
stop_archnotlog.sh	Script shell utilisé par Connect:Express lorsqu'il s'arrête pour arrêter le process d'archivage archnotlog
notify.sh	Script shell utilisé par Connect:Express pour lancer un process tom_notify pour les notifications journal
testrebas	Petit utilitaire permettant de vérifier qu'un nom de fichier vérifie une expression à indiquer dans le paramètre ind d'une commande tom_notify -msg L
notcfg.xml	Exemple de fichier de configuration à adapter
notif.h	Include C décrivant la structure de la notification journal

Par ailleurs si un répertoire de sauvegarde rci est utilisé, il est nécessaire (suite à l'installation) de créer manuellement ce répertoire. Celui-ci doit obligatoirement se situer sur le même système de fichiers que Connect:Express. On peut créer par exemple \$TOM_DIR/rci.

Outre les notifications sauvegardées dans le répertoire rci, le logiciel génère automatiquement les fichiers suivant dans le répertoire notif :

notif.log	Fichier log commun à tous les process tom_notify
archnotlog.trc	Fichier trace de l'exécutable archnotlog. Généré si le paramètre t est mentionné au lancement de archnotlog.
queue0,1, ...	Fichiers vides permettant de définir différentes messages queues IPC pour les process tom_notify en tâche de fond
anlqueue	Fichier vide permettant de définir la message queue IPC du process archnotlog
anl_out.<process-id>	Fichiers trace du lancement par Connect:Express de archnotlog. Ces fichiers ne sont générés que si STRACE=1 dans \$TOM_DIR/config/sysin
No_xQQQNNNNN	Fichier trace du lancement par Connect:Express d'un process tom_notify pour un transfert n° QQQNNNNN. X peut prendre les valeurs I, E ou R suivant que l'on est en début, en fin de transfert ou en erreur. Ces fichiers ne sont générés que si STRACE=1 dans \$TOM_DIR/config/sysin
Répertoire log	Répertoire qui contient les fichiers notif.log archivés. Ce répertoire est créé automatiquement sous \$TOM_DIR/notif par archnotlog s'il n'existe pas.
log/LAAAAMMJJ	Fichier log archivé à la date AAAA/MM/JJ. Au maximum 7 fichiers d'archive sont conservés. Les plus anciens sont purgés automatiquement..

Utilitaire d'archivage du fichier notif.log

Si l'on utilise la fonction journal de Connect:Express, un process archnotlog est automatiquement lancé en tache de fond lors du lancement de Connect:Express et arrêté à l'arrêt de ce dernier. Ce lancement et cet arrêt se font par le biais des scripts start_archnotlog.sh et stop_archnotlog.sh.

La commande archnotlog admet les paramètres suivants :

```
./archnotlog <start|stop> [r][t]
```

```
start : lancement
stop:   arrêt
t :     trace dans archnotlog.trc demandée
r :     processus en tâche de fond
```

Par exemple :

```
./archnotlog start r
...
./archnotlog stop
```

Lance le processus en tâche de fond puis l'arrête.

```
./archnotlog start
```

Lance l'archivage en batch. Le process s'arrête une fois le traitement effectué.

Lorsque archnotlog est lancé en tâche de fond, les archivages se font chaque jour à l'heure indiquée par le paramètre <ati>HHMM</ati> du fichier notcfg.xml. L'heure par défaut est 0 heure 30 mn. Les fichiers notif.log sont archivés dans le répertoire \$TOM_DIR/notif/log avec le nom LAAAAMMJJ. Au maximum 7 fichiers sont conservés. Les plus anciens sont purgés automatiquement.

Utilitaire de mise au point : displaynot

Cet utilitaire permet d'afficher le détail d'une notification journal

```
./displaynot <filename>
```

Par exemple :

```
./displaynot ../rci/N_J_E16400005
```

Affiche :

```
-----
notificationType      = U02
clientName            = ADMIN
requestNumber         = 16400005
registrationDate      = 2005/06/13
registrationTime      = 16:10:25
status               = E
trc                  = 0000
prc                  = 0000
src                  = 0000
nrc                  = 0000
transferDirection    = T
fileName              = FILE03
physicalName          = $TOM_DIR/config/sysin.txt
remotePhysicalName    =
label                 = $TOM_DIR/config/sysin.txt
transferOrigin        = BOUCLE
transferDestination  = BOUCLE
localName             = BOUCLE
partnerName           = BOUCLE
typeOfFile            = BU
typeOfData            = B
fileRecordLength     = 04000
fileNumberOfRecords  = 000000000001
networkBytes          = 000000001047
notificationLink      = T
externalRequestNumber =
transferIdentifier    = A40005
requestor             = pga
tcpipRc               = 0000
x25Rc                 =
x25Cause              =
x25Diag               =
```

```

requestStatus          = N
typeOfUser             = I
typeOfRequest          = N
priority               = 0
typeOfLink             = 2
originalPhysicalName   =
fileOrganization       = S
realCompression        =
networkMessageSize     = 4096
typeOfPartner          = O
protocol               = E
tcpipPort              = 06677
tcpipAddress           = 127.0.0.1
tcpipHost              =
x25RemoteAddress       =
x25LocalAddress        =
x25LocalPort           =
x25UserDataField       =
x25Facilities          =
transferBeginningDate  = 2005/06/13
transferBeginningTime  = 16:10:25
transferEndDate        = 2005/06/13
transferEndTime        = 16:10:25
retryNumber            = 00
fileOpenOption         = R
requestDate            = 2005/06/13
requestTime            = 16:10:25
userDataSent           =
userDataReceived       =
monitorIdentification  = CEXP01

```

Utilitaire de mise au point : testrebas

Ce petit utilitaire permet de déterminer si un nom simple de fichier satisfait à un critère de sélection à indiquer dans un paramètre ind.

Avec le fichier test.txt contenant

```
string=N_J_E16400005,expression=N_J_E*
```

./testrebas test.txt affichera :

```
regexec MATCH string=N_J_E16400005,expression=^N_[\.\?]*_E.*$
```

Indiquant que N_J_E16400005 satisfait au motif N_J_E*

Les motifs valides sont des motifs simples acceptés par exemple par la commande ls.

? signifie : 1 caractère quelconque

* signifie : 0 ou plus caractères quelconques

Utilitaire de mise au point : recvnot

Ce petit utilitaire permet de tester la réception des notifications, donc de simuler l'application d'administration.

```
./recvnot <port>
```

Par exemple :

```
./recvnot 9000
```

L'utilitaire se met à l'écoute sur le port 9000. Les notifications reçues du tom_notify distant sont écrites dans des fichiers RNOT1, RNOT2, ..., dans le répertoire courant.

Selon la plate-forme UNIX utilisée, l'utilitaire s'arrête par CTRL-C ou par un kill -9 <process-id>

Chapitre 5

Ce chapitre présente une façon de recevoir les notifications émises par l'utilitaire.

Réception des notifications de Connect:Express

Les notifications envoyées par l'utilitaire sont destinées à être affichées et à permettre de déclencher des traitements ou des alertes.

Utilisation du client Connect:Express Windows

Une notification reçue par un client Connect:Express Windows bénéficie de toutes les fonctionnalités de l'outil de Réception des Notifications : affichage, déclenchement d'alerte, déclenchement de procédure. Se reporter au manuel '*Guide de l'installation et des Utilitaires de Connect:Express Windows*'

Seules sont supportées par cet outil les Notifications au format Connect:Express, donc avec les paramètres suivants: msg=J avec txt non renseigné.

Client/N° Requête	Moniteur	Date	Heure	Identifi...	Fichier	Partenaire	Etat	Nom physique	Nb. d'enreg...	Octets transférés	Typ
TESTNFY/ 00000786	TOM4	2002/10/01	11:04:37		TYLNFY1		Terminé	PSR\$TST.TYLNFY1.D021001.H1104...	0	0	
TESTNFY/ 00000784	TOM4	2002/10/01	11:04:37		TYLNFY1		Terminé	PSR\$REC.P5.F080.SHORT	0	0	
TESTNFY/ 00000785	TOM4	2002/10/01	11:04:38		F1DEL		Terminé	PSR\$REC.P5.F080.SHORT	0	0	

Pour l'aide, appuyez sur F1

Annexe 1

Cette annexe présente la structure de la notification journal de Connect:Express.

Structure de la notification Connect:Express

Champ	Ig	Description
"U02"	3	Type de message = notification de fin de transfert
Client	8	Nom du destinataire de la notification (ADMIN)
Numéro de requête	12	Identification du transfert au sens CONNECT:Express
Date de notification	10	Date courante
Heure de notif.	8	Heure courante
Identifiant applicatif	16	Identification du transfert attribué par l'application
Status du transfert	1	C:En cours,A:Attente,D:Différé,E:Terminé,O:Terminé anormalement, J:Reprise automatique,K:Attente reprise
Code retour TRC	4	Code retour CONNECT:Express
Code retour PRC	4	Code retour protocolaire
Code retour SRC	8	Code retour système
Filler	4	
Code retour NRC	4	Code retour réseau
Direction	1	T = émission, R = réception
Nom de fichier	8	Identifiant symbolique au répertoire de CONNECT:Express
Fichier physique	127	Nom sur disque du fichier transféré chez l'émetteur de la notification
Fichier distant	44	Nom du fichier chez le partenaire
Label de fichier	80	Identifiant complémentaire du fichier (Pi37 PeSIT)
Origine	8	Identifiant de l'émetteur de la demande de transfert
Destination	8	Identifiant du récepteur de la demande de transfert
Nom du moniteur	8	Identifiant de CONNECT:Express en charge du transfert
Nom de partenaire	8	Identifiant symbolique au répertoire de CONNECT:Express
Type de fichier	2	TF = Texte fixe, TV = texte variable, BF = binary fixe, BU = Binary undefined
Type de données	1	A = Ascii, E = Ebcdic, B = Binaire
Taille d'article	5	En octets
Nombre d'articles	12	Nombre d'enregistrements du fichier
Nombre d'octets	12	Nombre d'octets du fichier
Type de lien notification	1	Type de lien pour la notification (T :TCP/IP)
Filler	128	
Filler	30	
Num.de requ. externe	8	Identification du transfert chez le partenaire, valide seulement en PeSIT

Champ	Ig	Description
		chez le serveur récepteur.
Filler	16	
Ident. transfert	6	Identification (numérique) du transfert échangé entre partenaire, valide seulement en PeSIT.
Nom utilisateur	8	Origine de la requête de transfert
Filler	24	
Code retour TCP/IP	4	Code retour spécifique à TCP/IP
Filler	4	
Filler	8	
Code retour X25	4	Code retour spécifique à X25
Cause X25	2	
Diagnostic X25	2	
Status requête	1	Requête purgée Y = oui , N = non
Filler	1	
Type d'utilisateur	1	I = Interne, E = Externe
Type de requête	1	N = Normal, I = Interrogaton, H = Attente
Filler	1	
Priorité	1	0 = Urgent, 1 = Normale , 2 = Lent
Type de lien	1	1 = X25, 2 = TCP/IP
Filler	135	
Fichier d'origine	44	Nom de fichier proposé au partenaire
Filler	124	
Filler	14	
Filler	14	
Filler	32	
Organisation	1	S = Sequentielle, I = Indexée, R = Relative
Filler	1	
Filler	1	
Compression	2	Type de compression utilisé (Protocole Pesit): Blank = non, H : Horizontale, V : Verticale, M : Mixte
Filler	7	
Taille du fichier	12	Nombre d'octets du fichier
Taille de message	4	
Filler	16	
Type de partenaire	1	T = CONNECT:Express, O = Autre
Protocole	1	D = PeSIT-D, E= PeSIT-E, 3 = ETEBAC3, F = FTP
Filler	5	
Port TCP/IP	5	Adresse du partenaire distant
Adresse TCP/IP	15	Adresse du partenaire distant
Nom de host TCP/IP	127	Adresse du partenaire distant
Filler	8	

Champ	Ig	Description
Filler	8	
Filler	64	
Adresse distante X25	15	Adresse du partenaire distant
Adresse locale X25	15	Adresse locale
Port X25	2	Adresse locale
Don. utilisateur X25	8	Adresse du partenaire distant
Facilités X25	32	Adresse du partenaire distant
Date de début	10	Début du transfert
Heure de début	8	Début du transfert
Date de fin	10	Fin du transfert
Heure de fin	8	Fin du transfert
Nombre de reprises	2	
Action à l'ouverture	1	C = Création d'un nouveau fichier R = Remplacement d'un fichier existant O = Ouverture d'un fichier existant
Filler	27	
Nomb. de send/receive	12	Messages réseau
Date de la requête	10	Dépôt de la requête de transfert
Heure de la requête	8	Dépôt de la requête de transfert
Filler	92	
Données utilisateur émises	254	Pi99 émis
Données utilisateur reçues	254	Pi99 reçu
Identification moniteur	8	Paramètre dpcsid du fichier sysin
Emetteur	24	Emetteur Pi61
Recepteur	24	Recepteur Pi62
0D0A	2	

Annexe 2

Cette annexe présente l'include C notif.h décrivant la notification journal de Connect:Express.

notif.h

```
typedef struct tag_snot {
    char    notificationType[3];        /* Notification version (value U02) */
    char    clientName[8];              /* Client name */
    char    requestNumber[12];          /* Request number */
    char    registrationDate[10];        /* Registration date (YYYY/MM/DD) */
    char    registrationTime[8];        /* Registration time (HH:MM:SS) */
    char    userRequestId[16];          /* Application identification */
    char    status;                     /* Transfer status (C/A/D/E/O/J/K) */
    char    trc[4];                      /* C:X return code */
    char    prc[4];                      /* Protocol return code */
    char    src[8];                      /* System return code */
    char    fil_erc[4];                  /* */
    char    nrc[4];                      /* Network return code */
    char    transferDirection;           /* Transfer direction (T/R) */
    char    fileName[8];                 /* Symbolic file name */
    char    physicalName[127];           /* Local physical file name */
    char    remotePhysicalName[44];      /* Remote physical file name */
    char    label[80];                   /* File label */
    char    transferOrigin[8];           /* Origin name */
    char    transferDestination[8];      /* Destination name */
    char    localName[8];                /* Monitor name */
    char    partnerName[8];              /* Partner name */
    char    typeOfFile[2];                /* Type of file:TF,TV,BF,BU */
    char    typeOfData;                  /* Type of data A=Ascii, E=Ebdcic,B=Binary */
    char    fileRecordLength[5];         /* Local record length */
    char    fileNumberOfRecords[12];     /* Number of records sent/received */
    char    networkBytes[12];            /* Number of bytes transferred */
    char    notificationLink;            /* Type of link for notifications (T:TCPIP) */
    char    filler01[128];               /* */
    char    filler02[30];                /* */
};
```

2 Mise en Oeuvre de l'Utilitaire de Notification

```
char    externalRequestNumber[8];    /* External request number */
char    filler03[16];                /* User identification */
char    transferIdentif[6];          /* Transfer identification */
char    requestor[8];                /* User name */
char    filler04[24];                /* */
char    tcpipRc[4];                  /* TCP/IP return code */
char    filler05[12];                /* */
char    x25Rc[4];                    /* X25 return code */
char    x25Cause[2];                 /* X25 cause code */
char    x25Diag[2];                  /* X25 diagnostic code */
char    requestStatus;                /* Status of request : Y:OK,N:Not OK */
char    filler06[1];                 /* */
char    typeOfUser;                  /* Requestor user type: I=Internal,E=External */
char    typeOfRequest;               /* Type of request: N=Normal,I=Inquiry,H=Hold */
char    filler07[1];                 /* */
char    priority;                    /* Priority:'0','1' or '2' */
char    typeOfLink;                  /* Type of link: '0'=LU6.2,'1'=X25,'2'=TCP/IP */
char    filler08[135];               /* */
char    originalPhysicalName[44];     /* Original Physical file name */
char    filler09[124];                /* */
char    fil_ccreate[14];              /* */
char    fil_majdate[14];              /* */
char    filler10[32];                /* */
char    fileOrganization;             /* File Organisation: S=sequential,R=relative,I=indexed */
char    filler11[2];                 /* */
char    fil_typeOfAllocation;         /* */
char    realCompression;              /* Compression mode: */
                                        /* Space= No,H=Horizontal,V=VerticalM=Mixed */
char    filler12[7];                 /* */
char    fileBytes[12];                /* */
char    networkMessageSize[4];        /* Max message size negotiated */
char    filler13[16];                /* */
char    typeOfPartner;                /* Partner type:T=TOM,O=Other */
char    protocol;                    /* Protocol:D=PeSIT-D,E=PeSIT-E,F:FTP,3:ETEBAC3 */
char    filler14[5];                 /* */
char    tcpipPort[5];                 /* TCP/IP: Port number */
char    tcpipAddress[15];             /* TCP/IP: address */
char    tcpipHost[127];               /* TCP/IP: host name */
char    filler15[80];                /* */
char    x25RemoteAddress[15];         /* X25: remote address */
char    x25LocalAddress[15];          /* X25: local address */
char    x25LocalPort[2];              /* X25: port number */
char    x25UserDataField[8];          /* X25: user data */
```

```

char    x25Facilities[32];           /* X25: facilities */
char    transferBeginningDate[10];  /* Begin of transfer date (YYYY/MM/DD) */
char    transferBeginningTime[8];   /* Begin of transfer time (HH:MM:SS) */
char    transferEndDate[10];        /* End of transfer date (YYYY/MM/DD) */
char    transferEndTime[8];         /* End of transfer time (HH:MM:SS) */
char    retryNumber[2];             /* Number of recall/restart made */
char    fileOpenOption;             /* Action made at open:          */
                                        /* C=Created,R=Replaced,O=Opened */
char    filler16[27];               /* */
char    sendReceiveCount[12];       /* Number of network I/O */
char    requestDate[10];            /* Date when the transfer is accepted */
                                        /* by Connect:Express (YYYY/MM/DD) */
char    requestTime[8];             /* Time when the transfer is accepted */
                                        /* by Connect:Express (HH:MM:SS) */
char    filler17[92];               /* */
char    userDataSent[254];           /* Pi99 sent */
char    userDataReceived[254];      /* Pi99 received */
char    monitorIdentification[8];   /* sysin DPCSID parameter */
char    pi61[24];                   /* User Sender Pi61 */
char    pi62[24];                   /* User Receiver Pi62 */
char    crlf[2];                    /* 0x0D 0x0A */
} snotT;

```

