



Connect:Express[®] Unix

Release Notes
Version 1.4.6-1

Table des matières

MODIFICATIONS ET AJOUTS DANS CONNECT:EXPRESS UNIX V146-1.....	4
NOUVELLES FONCTIONNALITES DE LA VERSION 1.4.6.....	5
PESIT MESSAGE	5
<i>Généralités.....</i>	5
<i>Les opérations associées au PeSIT Message</i>	6
Envoyer un message - le type de requête M	6
Recevoir un message – stocker les données	7
STORE AND FORWARD – ACQUITTEMENT DE BOUT EN BOUT.....	8
<i>Généralités.....</i>	8
<i>Acquittement de bout en bout - EERP.....</i>	8
Enregistrer les paramètres	8
Retrouver les paramètres	9
Envoyer un acquittement de bout en bout	10
Recevoir un acquittement de bout en bout	11
<i>Les opérations liées au Store and Forward</i>	11
Généralités.....	12
Utilisation du routage automatique	12
CONTROL DES DN	14
<i>Généralités.....</i>	14
<i>Mise en œuvre.....</i>	14
Configurer le contrôle des certificats.....	15
Contrôle des certificats d'un partenaire.....	16
IDENTIFICATION FTP ETENDUE	17
<i>Le fichier des identifiants FTP étendus.....</i>	17
<i>Le champ DPCSID ALIAS avec FTP.....</i>	18
L'UTILITAIRE P1B8PE2E	19
<i>Acquitter un transfert.....</i>	19
<i>Rediriger un transfert.....</i>	19
<i>Description de P1b8pe2e.....</i>	19
EERP - Requête.....	19
EERP - Définition d'un transfert.....	20
Rediriger une requête	21
Rediriger une définition de transfert.....	21
Codes Erreur.....	22
NOUVELLES VARIABLES, NOUVEAUX PARAMETRES ET CODES RETOUR	23
<i>Variable Symbolique.....</i>	23
<i>Paramètres des commandes utilisateur.....</i>	23
<i>Codes TRC.....</i>	24

Modifications et ajouts dans Connect:Express UNIX V146-1

28/01/2010 : **SR 15973** : Correction problème en SUN SOLARIS dans purge_queue.sh.
Modification de purge_queue.sh (tom/config).
Livraison de purge_queue.sh.

02/02/2010 : **SR 15943** : Ajout d'une nouvelle procédure check_httpn.sh qui assure la surveillance de la présence du process tom_httpn et sa relance si nécessaire (à lancer périodiquement via crontab).
Création de survey_httpn.sh (tom/httpn).
Livraison de survey_httpn.sh.

08/02/2010 : **case 135892** : Ajout d'une nouvelle procédure check_apm.sh qui assure la surveillance de la présence du process tom_apm sur un port d'écoute paramétrable et sa relance si nécessaire (à lancer périodiquement via crontab).
Création de survey_apm.sh (tom/config).
Livraison de survey_apm.sh.

10/02/2010 : **case 138833** : Correction problème sur code retour toujours à 1 même si le traitement est OK (si pas de fichier paramètre).
Modification de tom_prm.c (tom/prm).
Livraison de tom_prm.

17/02/2010 : **case 143439** : Prise en compte d'une nouvelle variable d'environnement \$THR_PI42 à ajouter dans le fichier profile. La valeur XX contenue dans cette variable, permet d'ajouter en réception une tolérance de XX % d'erreur sur la taille maxi du fichier annoncée par l'émetteur dans le pi42.
Modification de decl.h, declc.h et tom_apm.c (tom/strf).
Livraison de tom_apm.

24/02/2010 : **case 133791** : Actualisation des valeurs par défaut à la création des tables de session et de présentation.
Modification de oto0.c (tom/gtrf).
Livraison de tom_mon, RPRE et RTAB.

10/03/2010 : **case 152743** : Modification du contrôle protocolaire, en mode serveur, sur le premier message ajouté à partir de la version 145, qui s'avère être trop rigoureux vis-à-vis de partenaires demandeurs non Connect :Express.
Modification de r_tcp.c (tom/strf).
Livraison de tom_apm.

13/04/2010 : **case 164852** : Prise en compte d'une nouvelle variable d'environnement \$REC_EMPTY à ajouter dans le fichier profile. Si la variable \$REC_EMPTY existe et est égale à 1, les records vides (lg=0) , pour les fichiers en mode TV, sont remplacés par un caractère blanc (lg=1).
Modification de decl.h, declc.h et tom_apm.c (tom/strf).
Livraison de tom_apm.

Nouvelles Fonctionnalités de la Version 1.4.6

La Version 1.4.6 implémente un nouveau service PeSIT, une nouvelle fonctionnalité SSL et un nouveau système d'identification FTP. Ces fonctionnalités sont décrites dans ce qui suit, et les guides correspondants ont été mis à jour.

Le mode message PeSIT et le store and forward sont inclus dans le guide *Connect:Express Unix User and Installation* ainsi que toutes les modifications d'ensemble liées aux nouvelles fonctionnalités. Le contrôle des certificats a été inclus dans le guide *Connect:Express Unix Option SSL*. Le système d'identification FTP étendue est inclus dans le guide *Connect:Express Unix FTP Guide*. L'utilitaire p1b8pe2e est décrit dans le guide *Connect:Express Unix mise en œuvre des interfaces d'exploitation*.

PeSIT Message

Cette section décrit le service message de PeSIT, comment l'utiliser et le configurer dans Connect:Express Unix.

Généralités

Le PeSIT Message est un service protocolaire qui permet d'envoyer des données en une seule étape :

Message + données <-> AckMessage

Au lieu de l'enchaînement :

```
Create <-> AckCreate
      +
      Open <-> AckOpen
      +
      Write <-> AckWrite
      +
      .....
      N * Data
      /
      Sync <-> Async
      .....
      +
      DataEnd
      +
      TransEnd <-> AckTransEnd
      +
      Close <-> AckClose
      +
      Deselect <-> AckDeselect.
```

Vous pouvez mettre en œuvre cette fonctionnalité pour envoyer des messages courts, mais aussi des fichiers. Elle est utilisée également dans la mise en œuvre des acquittements de bout en bout de transferts de fichiers simples ou dans le cadre de mécanismes de store and forward. Un utilitaire appelé p1b8pe2e est fourni : il permet d'envoyer un acquittement ou de rediriger un fichier.

Recevoir un message – stocker les données

A la réception d'un message PeSIT, Connect:Express utilise le nom symbolique de fichier reçu dans le paramètre PeSIT pi12: si ce nom est défini au répertoire (RFIC) il l'utilise pour traiter la réception. Si ce nom n'existe pas, il recherche le nom symbolique \$\$MSGD\$\$: s'il le trouve, avec l'état 'E', il l'utilise. S'il ne le trouve pas ou si l'état est 'H', le transfert est rejeté.

Un message peut contenir soit des données, soit l'acquittement d'un transfert précédent. Le paramètre PeSIT pi11 indique la nature des informations : 'FFFF', en hexadécimal, indique que c'est un message, 'FFFE' indique que c'est l'acquittement d'un message. Tout autre valeur indique que c'est l'acquittement d'un transfert de fichier, la valeur du pi11 correspondant à celle utilisée pour le transfert initial.

On peut stocker le message reçu de deux façons : l'écrire dans un fichier ou seulement le stocker dans le fichier RENC. Connect:Express se reporte aux attributs du fichier symbolique concerné. Si un nom de fichier physique est fourni dans le profil, les données y seront écrites et les 254 premiers caractères seront stockés dans le champ r_pi99_254 du fichier RENC , affiché dans le champ MESSAGE << de STERM .

```
10/06/04 16:24:23 REQUEST 07200008 MSGFIC2 <- BOUCLE MESSAGE ACCEPTED STRF 0000011700
10/06/04 16:24:23 REQUEST 07200008 MSGFIC2 <- BOUCLE MESSAGE RECEIVED STRF 0000011700
10/06/04 16:24:23 REQUEST 07200008 254 first characters of the file
10/06/04 16:24:23 REQUEST 07200008 $TOM_DIR/msg/MSGFIC2_A7200008
```

Si aucun nom physique de fichier n'est fourni (le champ doit être égal à '-'), les données sont placées dans le champ r_pi99_254 du fichier RENC (254 caractères maximum).

```
10/06/04 16:24:23 REQUEST 07200008 MSGFIC2 <- BOUCLE MESSAGE ACCEPTED STRF 0000011700
10/06/04 16:24:23 REQUEST 07200008 MSGFIC2 <- BOUCLE MESSAGE RECEIVED STRF 0000011700
10/06/04 16:24:23 REQUEST 07200008 254 first characters of the file
```

L'écran de suivi de STERM affiche les données du message.

C:E/UNIX 146-1 ----- SUIVI DES REQUETES ----- tom1								
OPTION ===>								
NUM.REQ.	FICHER	AVEC	DIR.	PRI.	TYPE-REQ.	ETAT	STRF	ID
07200001	FICTEST1	EXPRESS1	T	0	N NORMAL	E	000001	10408
07200005	FICTEST2	DPX1	T	0	N NORMAL	E	000001	11441
07200006	FICTST	SID1	R	0	N NORMAL	E	000001	11698
07200007	FICTEST2	DPX1	T	0	N NORMAL	E	000001	11443
07200008	MSGFIC2	BOUCLE	R	0	M MESSAGE	E	000001	11700
07200009	FICSTSN	DPX1	T	0	N NORMAL	E	000001	11445
07200010	FIC22424	SID1	R	0	N NORMAL	E	000001	11702
07200011	FICTEST3	DPX1	T	0	N NORMAL	E	000001	11447
07200012	ARECEVOI	SID1	R	0	N NORMAL	E	000001	11704
07200013	FICTEST3	DPX1	T	0	N NORMAL	E	000001	11449
07200014	ARECEVOI	SID1	R	0	N NORMAL	E	000001	11706
07200015	AENVOYER	DPX1	T	0	N NORMAL	O	000001	11451
07200017	FICTEST4	DPX1	T	0	N NORMAL	E	000001	11456
07200018	FICTST2	SID1	R	0	N NORMAL	E	000001	12225
07200019	FICTEST1	EXPRESS1	T	0	N NORMAL	J	000001	11458

<- -F10- -F3- FIN -F7- ECRAN PRECEDENT -F8- ECRAN SUIVANT -F11- ->

```

C:E/UNIX 146-1 ----- SUIVI DES REQUETES ----- tom1

REQUETE : 09600815   DE   : BOUCLE       DIRECTION : T       AVEC : BOUCLE
ORIGIN  : BOUCLE    DESTINATION : BOUCLE    XFER ID : 06292271 *MESSAGE
EMETTEUR:                               RECEPTEUR :
USERID  : gcz       PID STRF   : 0000014700 FA : N       NOT : 0       SSL : N
FICHIER : MSGFIC2   ORIGINE CMD : I       SSLPARM :          ETAT TRANSFERT : X
PROTOCOLE PESIT   CRC : N       MULTI : N       TRANSLATION : 0 COMPRESSION : 0
DSNAME   : $TOM_DIR/msg/MSGFIC2_A7200008
MESSAGE << : 254 first characters of the file

DEBUT  : 20100406 16:24:23  FIN  : 20100406 16:24:23  REPRISE/MAX : 00/00
NRC    : 0000   SRC  : 0000   TRC  : 0000   PRC  : 0 000   SSLRC : 00000000
NOMBRE D'ENREGISTREMENTS : 0000000005   K.OCTETS   : 0000000019
FORMAT ENREGISTREMENT . : BU           LONGUEUR ENREGISTREMENT : 04096
***** TCP/IP *****

HOST TCP/IP (09) : localhost
ADRESSE TCP/IP (00) :
PORT : 05015

```

Store and Forward – Acquittement de bout en bout

Cette section décrit les mécanismes du store and forward (qui met en œuvre l'acquittement de bout en bout basé sur le PeSIT message), la façon de l'utiliser et de le configurer dans Connect:Express Unix. L'acquittement de bout en bout est d'abord décrit, en tant que processus simple. Suivent les mécanismes de store and forward, qui incluent l'acquittement de bout en bout.

Généralités

Dans ce qui suit, on utilise l'acronyme 'EERP' pour 'End to End Response'. L'acquittement de bout en bout valide la réception d'un fichier (ou d'un message) par l'application destinataire. Cet acquittement peut être simplement véhiculé du récepteur à l'émetteur, ou, au travers d'un mécanisme de store and forward, depuis le destinataire final vers l'origine.

Acquittement de bout en bout - EERP

Ce paragraphe décrit les différentes étapes du mécanisme de l'EERP. Le paragraphe suivant le place dans le contexte des mécanismes de store and forward.

- Première étape : En fin de réception, sauver le contexte et en particulier l'identifiant de transfert.
- Deuxième étape : Retrouver le contexte et soumettre une requête de transfert d'EERP.
- Troisième étape : Recevoir l'EERP et prendre les mesures adéquates.

La liste des paramètres passés aux commandes de début et fin de transfert contient le type de requête (paramètre \$25) pour permettre de définir une action spécifique à un transfert de fichier (Type= N ou I), à un transfert de message (Type=M) ou à un transfert de EERP (Type=E). Tous les paramètres nécessaires à l'identification d'un transfert, le contexte, sont fournis. La liste des paramètres nécessaires à la mise en œuvre du store and forward et de l'acquittement de bout en bout sont indiqués dans le tableau ci dessous.

Enregistrer les paramètres

La première étape est réalisée de façon normale par l'enregistrement de la requête dans le fichier RENC. La commande de fin de transfert permet de les enregistrer par ailleurs ou de les utiliser en ligne. La table ci dessous montre la correspondance entre les paramètres Connect:Express, les champs PeSIT, les champs du fichier RENC, Les paramètres de la requête de transfert normale et les paramètres passés aux commandes utilisateur.

Paramètre	PeSIT-Pi	RENC-trfpar	Requête Normale	Commande
Nom du partenaire	3	pi.ident	SPN=	\$3
Alias	4	pi.idser	SID=	\$13
EERP Context				
Nom du fichier	3bis	pi.user_org	ORG=	\$11
	4bis	pi.user_dst	DST=	\$12
	11	pi.tyf		\$26 (nouveau)
	12	pi.nof	SFN=	\$2
Identification du transfert	13	pi.idt		\$27 (nouveau)
Date-heure du fichier	51	pi.dhc		\$28 (nouveau)
Nom du 'client'	61	pi.user_snd	SND=	\$15
Nom du 'service'	62	pi.user_rcv	RCV=	\$16

Connect:Express peut accéder aux informations enregistrées dans le fichier RENC pour construire un EERP, à moins que les paramètres soient fournis au moment de la demande par l'utilitaire **p1b8pe2e**, décrit au paragraphe "L'utilitaire P1b8e2e".

Retrouver les paramètres

Dans la deuxième étape, la construction de l'EERP nécessite la restitution des paramètres du transfert à acquitter, auxquels peut s'ajouter un message de compte rendu. Il est possible soit de fournir le numéro de requête initial, soit de fournir tous les paramètres du transfert initial.

- Donner le numéro de requête et un message de compte-rendu – la requête doit être terminée et présente dans le fichier RENC.

Information attendue	Champ	Description	PeSIT
Numéro de requête	REQ=	Requête locale à acquitter	
Nom du partenaire	SPN=	Partenaire adjacent	Pi3 (Connect)
Alias	SID=	Nom local	Pi4 (Connect)
Notification	NTF=	0-7	
Priorité	PRT=	0-2	
Type de lien	LNK=		
Date d'exécution	DAT=		
Message utilisateur	ACK=	Compte-rendu applicatif, lg <= 254	Pi91

- Fournir les informations du transfert et un message de compte-rendu.

Information attendue	Champ	Description	PeSIT
Nom du partenaire	SPN=	Partenaire adjacent	Pi3 (Connect)
Alias	SID=	Identification locale	Pi4 (Connect)
Notification	NTF=	Nom local	
Priorité	PRT=	0-2	
Type de lien	LNK=		
Date d'exécution	DAT=		
EERP Context			
Nom du fichier	ORG=	L<=24	Pi3bis
	DST=	L<=24	Pi4bis
	P11=	L = 2 hexadécimal	Pi11
	P12=	L<=14	Pi12
Identification du transfert	P13=	L<=08 décimal	Pi13
Date-heure du fichier	P51=	L=12	Pi51
Nom du 'client'	P61=	L<=24	Pi61
Nom du 'service'	P62=	L<=24	Pi62
Feedback			
Message utilisateur	ACK=	Compte-rendu applicatif, lg <= 254	Pi91

Envoyer un acquittement de bout en bout

Pour envoyer un EERP, il faut soumettre une requête de transfert d'EERP à Connect:Express. Pour cela on utilise l'utilitaire batch p1b8pe2e, un programme, ou l'interface opérateur STERM. Connect:Express construit l'EERP soit directement à partir des paramètres fournis, soit en accédant à l'enregistrement RENC. La requête initiale doit être une réception, un fichier ou un message, terminée (status = 'E'). Un code TRC=2050, 2051, 2053 ou 2055 sera retourné si la requête est présente dans le fichier RENC et ne satisfait pas ces conditions.

Le transfert d'un EERP ne nécessite pas de définition de fichier symbolique spécifique pour s'exécuter. Si le profil \$\$EERP\$\$ est défini au répertoire RFIC, et si son état est 'E', le transfert sera exécuté sur la base de ce profil: par exemple les exits, les commandes utilisateur, le nom physique de fichier déclarés pour ce profil seront utilisés. Si le fichier de la requête est défini au répertoire, si il est actif, et si le profil \$\$EERP\$\$ est défini, mais inactif, c'est le profil du fichier de la requête qui sera utilisé.

Lorsque le transfert de l'EERP est réussi, l'état de la requête initiale passe de 'E' à 'X'.

Utilisation de STERM

On peut soumettre une requête de transfert d'EERP, sans compte rendu, par STERM. L'EERP est construit à partir de l'enregistrement de la requête dans le fichier RENC. Aucune donnée n'est associée au message PeSIT.

```
C:E/UNIX 146-1 ----- SUIVI DES REQUETES ----- tom1
OPTION ==>
  NUM.REQ.   FICHER  AVEC      DIR.  PRI.  TYPE-REQ.  ETAT  STRF ID
  REQ.NUM.   FILE    WITH DIR.  PRI.   REQ. TYPE STATE  STRF ID
07200001  FICTEST1 EXPRESS1 T      0      N NORMAL  E      0000010408
07200003  FICTEST1 EXPRESS1 T      0      E EERP    E      0000013254
07200005  FICTEST2 DPX1  T      0      N NORMAL  E      0000011441
07200006  FICTST  SID1  R      0      N NORMAL  E      0000011698
07200007  FICTEST2 DPX1  T      0      N NORMAL  E      0000011443
07200008  DOUDOU  SID1  T      0      M MESSAGE E      0000011700
07200009  FICSTSN DPX1  T      0      N NORMAL  E      0000011445
07200010  FIC22424 SID1  R      0      N NORMAL  E      0000011702
07200011  FICTEST3 DPX1  T      0      N NORMAL  E      0000011447
07200012  ARECEVOI SID1  R      0      N NORMAL  E      0000011704
E 07200013  FICTEST3 DPX1  R      0      N NORMAL  E      0000011449
07200014  ARECEVOI SID1  R      0      N NORMAL  E      0000011706
07200015  AENVOYER DPX1  T      0      N NORMAL  O      0000011451
07200017  FICTEST4 DPX1  T      0      N NORMAL  E      0000011456
07200018  FICTST2  SID1  R      0      N NORMAL  E      0000012225
07200019  FICTEST1 EXPRESS1 T      0      N NORMAL  J      0000011458

<- -F10- -F3- FIN -F7- ECRAN PRECEDENT -F8- ECRAN SUIVANT -F11- ->
```

Utilisation de l'utilitaire P1b8pe2e

Pour envoyer un message de compte rendu, il faut utiliser le programme batch p1b8pe2e, avec le paramètre /ACK=, ou /DSN= pour placer le compte rendu dans un fichier.

FUN=E signifie 'envoyer un EERP', REQ='numéro de requête', ACK='compte rendu': ce paramétrage provoque la recherche de la requête dans le fichier RENC, et la construction de l'EERP correspondant. Le compte rendu est envoyé dans le message PeSIT.

```
p1b8pe2e "/FUN=E/SPN=ident/REQ=xxxxxxxx" "/ACK='compte rendu'"
```

Si la requête a été purgée, il faut fournir tous les paramètres du transfert initial.

FUN=E, contexte EERP (/ORG=/DST=/P11=/P12=/P13=/P51=/P61=/P62=), ACK='compte rendu'.

```
p1b8pe2e "/FUN=E/SPN=ident" "'contexte EERP'" "/ACK='compte rendu'"
```

Utilisation de l'API L0b2z20

Pour soumettre une demande de transfert d'EERP depuis un programme C, utiliser la structure fournie dans le fichier d0b8z20.h, de la même façon que pour une demande de transfert, en fournissant les informations spécifiques.

```
struct st_sci {
  char dire[1];          /* Direction */
  char file[8];          /* Symbolic file name p1b8pe2e */
  char part[8];          /* Symbolic partner name p1b8pe2e */
  char dsnam[44];        /* Dsname */
  char prty[1];          /* Priority */
  char dat[8];           /* Date */
  char hour[6];          /* Hour */
  char lnk[1];           /* Link type */
  char udf[44];          /* User data file */
  char typ[1];           /* Request type = E p1b8pe2e */
  char sta[1];           /* State of Request */
  char dpcsid[8];        /* Dpcsid for Alias */
  char dpcpsw[8];        /* Dpcpsw for Alias */
  char format[2];        /* Record Format (TF TV BF BU) */
  char lrecl[5];         /* Record Length */
  char api[88];          /* Api Field */
  char tsm[3];           /* Type/Structure/Mode FTP */
  char stou[1];          /* Store Unique FTP */
  char fa[1];            /* flag File agent Y/N */
  char label[80];        /* Label */
  char s_pi99_254[254]; /* Feedback on 254 p1b8pe2e */
  char user_org[8];      /* User Origin p1b8pe2e */
  char user_dst[8];      /* User Destination p1b8pe2e */
  char user_snd[24];     /* User Sender pi61 p1b8pe2e */
  char user_rcv[24];     /* User Receiver pi62 p1b8pe2e */
  char quant_aa[2];      /* AA for Julian Date */
  char quant[3];         /* Julian Date */
  char notif[1];         /* Notification: space/0-7 */
  char noreq[8];         /* request number p1b8pe2e */
  char dhc[12];          /* File date Pi51 p1b8pe2e */
  char idt[8];           /* Pi13 p1b8pe2e */
  char ftype[4];         /* Pi11 p1b8pe2e */
  char filler[SIZE_RENC - 675];
};
```

Recevoir un acquittement de bout en bout

La réception d'un acquittement de bout en bout se fait au travers du mode message PeSIT, avec un pi11 différent de 'FFFF' ou 'FFFE'. Le nom symbolique de fichier est fourni dans le pi12. La description du PeSIT message s'applique .

Le transfert d'un EERP ne nécessite pas de définition de fichier symbolique spécifique pour s'exécuter. Si le profil \$\$EERP\$\$ est défini au répertoire RFIC, et si son état est 'E', le transfert sera exécuté sur la base de ce profil, dans tous les cas: par exemple les exits, les commandes utilisateur, le nom physique de fichier déclarés pour ce profil seront utilisés. Si le fichier de la requête est défini au répertoire, si il est actif, et si le profil \$\$EERP\$\$ n'est pas défini , ou est inactif, c'est le profil du fichier de la requête qui sera utilisé.

Quand il reçoit un EERP, Connect:Express recherche la requête correspondante. Ce doit être une émission, un fichier ou un message, elle doit être terminée. Le code TRC 2050 ou 2055 est généré si la requête est trouvée et ne satisfait pas ces conditions. Si la requête n'est pas trouvée, l'EERP est quand même accepté.

L'état de la requête initiale passe de 'E' à 'X' à la bonne fin du transfert de l'EERP.

```
09/07/03 16:00:22 REQUEST 00100026 FICMSG <- partner EERP:  org  dest  idt
09/07/03 16:00:22 REQUEST 00100026 FICMSG <- partner EERP  RECEIVED
09/07/03 16:00:22 REQUEST 00100026 User data
```

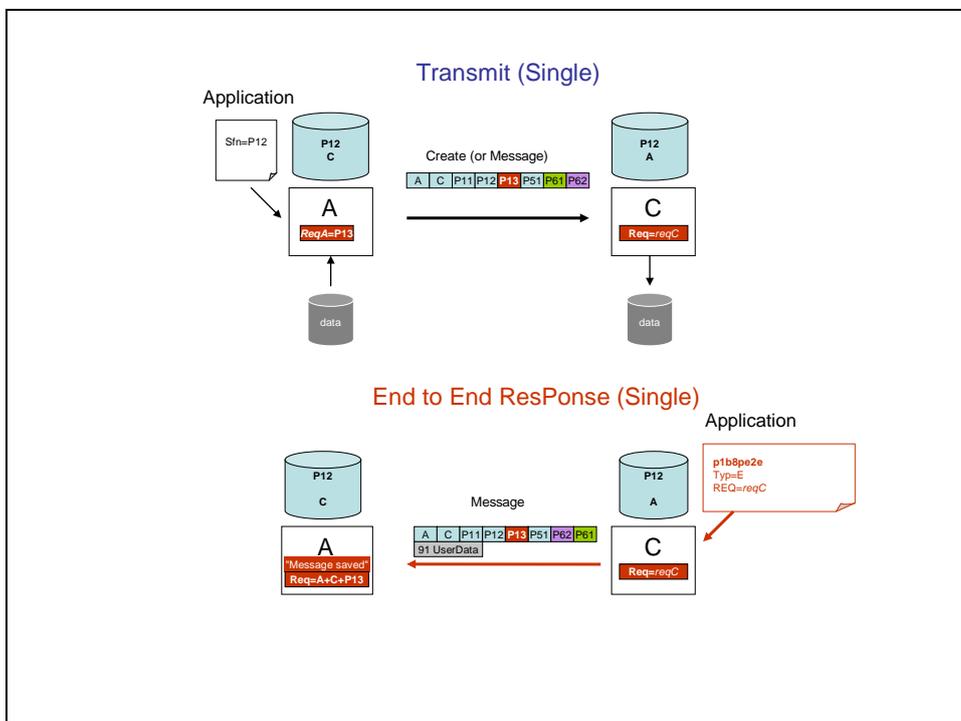
Les 254 premiers caractères du compte rendu s'affichent dans les écrans de suivi de STERM.

Les opérations liées au Store and Forward

La figure ci dessous montre comment l'acquiescement de bout en bout fonctionne: A envoie un fichier à C, et C envoie en retour un acquiescement pour confirmer la réception.

Chez A le numéro de requête, *ReqA*, est envoyé dans le pi13 qui représente l'identification du transfert. C reçoit le fichier avec le numéro de requête *ReqC*. L'enregistrement *ReqC* est sauvé chez C dans le fichier RENC. L'application acquiesce le fichier à partir des informations de la requête *ReqC*, parmi lesquelles se trouve le pi13 initial. L'EERP est construit à partir de *ReqC* et envoyé à A par le mode message. A reçoit l'EERP et recherche dans son fichier RENC la requête initiale, à partir des informations pi13+A+C.

A la fin du transfert réussi de l'EERP, C change l'état de la requête *ReqC* de 'E' à 'X' et A change l'état de la requête *ReqA* de 'E' à 'X'.



Selon que l'acquiescement de bout en bout circule dans un mécanisme de store and forward ou non, l'EERP peut être lui-même l'objet d'une redirection.

Généralités

Il est possible de mettre en œuvre le store and forward grâce à l'utilitaire p1b8pe2e. Les commandes de fin de transfert permettent de sauvegarder les paramètres pour usage ultérieur, ou d'activer une redirection ou un acquiescement automatique. La fonctionnalité de store and forward est fournie par Connect:Express Unix. Plusieurs commandes utilisateur sont disponibles : elles sont décrites au paragraphe suivant.

Utilisation du routage automatique

Lorsque le champ DPCSID ALIAS de la définition de partenaire est égal à la chaîne ****xxxx**** - où 'xxxx' contient les caractères A-Z, 0-9, a-z - la commande UEXxxxx est lancée en fin de réception. Ce mécanisme permet de rediriger un fichier, un message ou un EERP vers son destinataire. Les commandes de store and forward indiquées ci-dessous sont livrées.

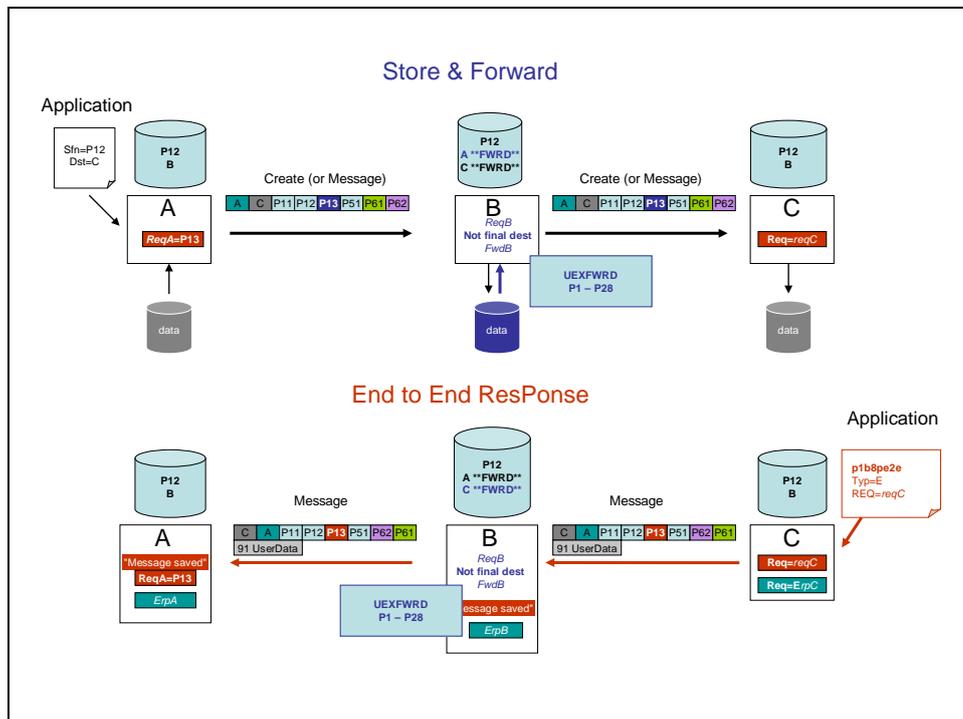
- exit/UEXFWRD utilise le programme p1b8pe2e pour rediriger le fichier courant.
- exit/UEXEERP utilise le programme p1b8pe2e pour acquiescer le fichier courant par un EERP.
- exit/UEXROUT utilise le programme p1b8req pour rediriger le fichier courant selon une règle basée sur les valeurs origin/destination (pi3bis/pi4bis).

La figure suivante montre le cas où A envoie un fichier vers C, via B, et C envoie en retour un acquiescement vers A, via B.

Les partenaires A et C sont configurés dans B avec DPCSID ALIAS = ****FWRD****. B active UEXFWRD en fin de réception d'un fichier ou d'un message quelconque en provenance de A, par l'utilisation du programme p1b8pe2e, avec FUN=F, type de requête N ou M. Tous les paramètres de bout en bout sont réutilisés dans le nouveau transfert. Le numéro de requête chez A est *ReqA* : il est envoyé dans le pi13. Les numéros de requêtes chez B sont *ReqB* en réception et *FwdB* en émission. Le numéro de requête chez C est *ReqC*.

Après traitement des données, l'application soumet à C une requête de transfert d'EERP par p1b8pe2e, pour ReqC, afin d'acquitter le fichier reçu. C'est une nouvelle requête, numéro ErpC. Connect:Express prépare le message PeSIT à partir des informations initiales du fichier RENC pour la requête ReqC.

B reçoit le message EERP, sous le numéro de requête ErpB. C change l'état de ReqC de 'E' à 'X'. B retrouve la requête en émission FwdB à partir des informations (Pi13+A+C+envoyé de A à C) et passe son état à 'X'. Les partenaires A et C sont configurés dans B avec DPCSID ALIAS = **FWRD**. B lance UEXFWRD en fin de réception, et active le programme p1b8pe2e avec FUN=F, type de requête = E. Tous les paramètres de bout en bout sont réutilisés dans le nouveau transfert. A reçoit le message EERP, sous le numéro de requête ErpA. A retrouve la requête en émission ReqA à partir des informations (Pi13+A+C+envoyé de A à C) et change son état en 'X'. B retrouve la requête en réception ReqB à partir des informations (Pi13+A+C+reçu de A à C) et change son état en 'X'.



Note : la différence entre UEXROUT est UEXFWRD est que UEXROUT ne redirige pas tous les paramètres initiaux. La requête de redirection est un nouveau transfert, avec un identifiant pi13 distinct.

Control des DN

Cette section décrit le mécanisme de contrôle des certificats, comment l'utiliser et le configurer dans Connect:Express Unix.

Généralités

Quand Connect:Express reçoit le certificat d'un partenaire distant, l'interface SSL (openssl) contrôle l'authenticité du certificat et sa présence dans la base des certificats du moniteur. Pour pouvoir vérifier que le nom contenu dans le certificat correspond à celui qu'on attend, il faut utiliser la fonction de contrôle des certificats.

On peut définir le nom de domaine, le DN, attendu pour une session entrante (DN Client) et une session sortante (DN serveur). En mode serveur, le contrôle est lié au paramètre OPTIONS DE VERIFICATION des paramètres SSL. Le DN du Client ne peut être contrôlé que si le paramètre OPTIONS DE VERIFICATION n'est pas égal à zéro. En mode client, le contrôle de DN est fait même si le paramètre OPTIONS DE VERIFICATION n'est pas égal à 2.

Mise en œuvre

Les définitions de DN sont gérées par STERM. Il faut définir des profils de contrôle de DN, puis associer à un partenaire le profil qui convient.

L'option 4 du menu SSL permet d'accéder au CONTROL DES CERTIFICATS

```
C:E/UNIX 146-1 ----- MENU PRINCIPAL (GLOBAL) ----- tom1
OPTION ===> LABS LINUX

                C O N N E C T   :   E x p r e s s
                    F o r   U n i x           (C) STERLING COMMERCE 2005

_ 1  REPERTOIRES      _ 2  MONITEUR      _ 3  TABLES          _ 4  REQUETE
    PARTENAIRES        SUIVI              SESSION          _ 5  SSL
    FICHIERS           LOG                PRESENTATION

                    PURGE DES ENCOURS

X  EXIT                                                    -F3- FIN
```

```

C:E/UNIX 146-1 -----SSL----- tom1
OPTION ==>

1 PARAMETRES DE SESSIONS SSL
2 IMPORTATION DE CERTIFICATS
3 PROPRIETES D'UN CERTIFICAT
4 CONTROLE DES CERTIFICATS

X EXIT -F3- FIN

```

Le contrôle des certificats se fait au travers de définitions de DN (Domain Name) . Pour un certificat, on peut contrôler le 'DN objet' (l'identité du partenaire) et le 'DN Racine ' (l'identité de l'autorité qui a délivré le certificat).

Les figures ci dessous représentent les définitions identifiées par le nom symbolique TESTDN.

```

C:E/UNIX 146-1 -----CONTROLE DES CERTIFICATS----- tom1
OPTION ==> V

C CREER
L LISTER
M MODIFIER
S SUPPRIMER
V VISUALISER

ID ==> TESTDN

X EXIT -F3- FIN

```

On peut afficher la liste des définitions, ajouter, modifier, supprimer, visualiser une définition.

Configurer le contrôle des certificats

Une définition de contrôle de certificat peut contenir à la fois les règles en mode serveur et en mode client. En session entrante, mode serveur, le certificat du client peut être contrôlé. En session sortante, mode client, le certificat du serveur peut être contrôlé. Les caractères '*' et '?' peuvent être utilisés, de la même façon que pour les noms de fichiers.

Le champ ETAT indique si le contrôle est actif ou non.

```

C:E/UNIX 146-1 -----CONTROLE DES CERTIFICATS----- tom1
OPTION ==>

ID   ==>          TESTDN01          ETAT   ==> E

DN CLIENT  DISTANT
DN OBJET   : CN=Test*
DN RACINE  : CN=CA*

DN SERVEUR DISTANT
DN OBJET   : CN=Test*
DN RACINE  : CN=CA*

OPTION : MODIFIER                      MAJ : 10/03/09 17:22 gcz
-ENTER- CHAMP SUIVANT                  -F3- ANNULLATION                    -F8- VALIDATION

```

La liste montre les définitions en deux parties.

```

C:E/UNIX 146-1 -----CONTROLE DES CERTIFICATS----- tom1
OPTION ==>
DN CLIENT  DISTANT

ID          ST DN OBJET   :          DN RACINE  :
TESTDN     E  CN=Test*   :          CN=CA*
TESTDN02   E  CN=test*   :

```

<- -F10- -F3- FIN -F7- ECRAN PRECEDENT -F8- ECRAN SUIVANT -F11- ->

F10 or F11 affiche la seconde partie.

```

C:E/UNIX 146-1 -----CONTROLE DES CERTIFICATS----- tom1
OPTION ==>
DN SERVER  DISTANT

ID          ST DN OBJET   :          DN RACINE  :
TESTDN     E  CN=Test*   :          CN=CA*
TESTDN02   E  CN=test*   :

```

<- -F10- -F3- FIN -F7- ECRAN PRECEDENT -F8- ECRAN SUIVANT -F11- ->

Contrôle des certificats d'un partenaire

On peut associer à un partenaire les règles de contrôles des certificats en mode client et serveur, au travers du paramètre CONTROLE DES CERTIFICATS.

```

C:E/UNIX 146-1 ----- REPERTOIRE DES PARTENAIRES ----- toml
OPTION ==>

NOM SYMBOLIQUE ..... : SSLBCL
MOT DE PASSE ..... : PSW
ETAT INITIALISATION ... : E
NATURE ..... : 0
NUMERO PROTOCOLE ..... : 3
TABLE DE SESSION ..... : 1
PORT X25 ..... :
NOMBRE DE SESSIONS .... : 20/10/10
TYPE DE LIAISON ..... : T
NUMERO X25 ..... :
(SOUS)ADRESSE LOCALE .. :
COMPLEMENT D'APPEL .... :
FACILITES ..... :
HOST TCPIP ..... : localhost
ADRESSE TCPIP ..... :
DPCSID ALIAS ..... : SSLBCL
DPCPSW ALIAS ..... : PSW
NOMBRE DE REPRISES .... : 00

INTERNE AU MONITEUR
E:EN-SERVICE H:HORS-SERVICE
T/O
1:ETEBAC 3, 2:FTP, 3:PESIT
1->9 TABLES DE SESSION
NOM DEVICE X25
01->64 TOT/IN/OUT
X, P, T OU M
1-15 CAR. (NO TRANSPAC)
1-15 CAR. (NO TRANSPAC)
'USER-DATA-FIELD'

PORT . : 05090
DEF FICHER FTP :
SSLPARM ID .... : SSLCLI
CONTROLE DES CERTIFICATS : TESTDN
INTERV.SESS ,TRF : 00, 00 MINUTES

OPTION : MODIFIER MAJ : 10/04/09 09:15 gcz

```

Codes retour:

Mode serveur : prc=302 partenaire client non autorisé
 Mode client : prc=304 server non autorisé

trc=2057 DN objet invalide
 trc=2058 DN racine invalide
 trc=2059 DN définition inexistante

Identification FTP étendue

L'identification FTP étendue permet d'utiliser, dans les transferts FTP, des noms et des mots de passe longs, au lieu des valeurs symboliques limitées à 8 caractères. Ce système permet, par exemple, d'établir des connexions au travers de pare feux.

Par exemple, pour se connecter à un serveur, via un pare feux, on utilise la syntaxe suivante:

```

USER my_username_any_length@server_host_name_any_length
PASSWORD my_password_any_length

```

Le fichier des identifiants FTP étendus

Vous pouvez définir les identifications FTP étendues dans le fichier *apmftpe* placé dans le répertoire /config.

```

$TOM_DIR/config/apmftpe

```

Utiliser un éditeur pour gérer ce fichier, et respecter la syntaxe montrée ci dessous: le mot clé ID= est obligatoire, la virgule est utilisée comme séparateur. Une définition peut contenir un identifiant utilisateur et un mot de passe, ou seulement l'un des deux.

```

ID=Aliasid,Ftp_Extended_Id,Ftp_Extended_Password
ou
ID=Aliasid,Ftp_Extended_Id (pas de mot de passe)

```

ou

ID=**Aliasid**,,Ftp_Extended_Password

(pas d'identifiant)

Le champ DPCSID ALIAS avec FTP

La définition d'un partenaire propose le champ DPCSID ALIAS. Si ce champ contient le mot clé **\$\$FTPE\$\$**, Connect:Express recherche un alias long dans le fichier apmfte, identifié par le nom symbolique de partenaire. Si le champ DPCSID ALIAS contient une chaîne de caractère commençant par **\$\$**, par exemple **\$\$alias**, Connect:Express recherche un alias long dans le fichier apmfte, identifié par la chaîne **'alias'**.

```
E/UNIX 146-1 ----- REPERTOIRE DES PARTENAIRES ----- tom1
OPTION ==>

NOM SYMBOLIQUE ..... : PFTP01
MOT DE PASSE ..... : PASSWD
ETAT INITIALISATION ... : E
NATURE ..... : 0
NUMERO PROTOCOLE ..... : 2
TABLE DE SESSION ..... : 1
PORT X25 ..... :
NOMBRE DE SESSIONS ... : 20/10/10
TYPE DE LIAISON ..... : T
NUMERO X25 ..... :
(SOUS)ADRESSE LOCALE .. :
COMPLEMENT D'APPEL .... :
FACILITES ..... :
HOST TCPIP ..... : localhost
ADRESSE TCPIP ..... :
DPCSID ALIAS ..... : $$FTPE$$
DPCPSW ALIAS ..... : PASSWD
NOMBRE DE REPRISES .... :

INTERNE AU MONITEUR
E:EN-SERVICE H:HORS-SERVICE
T/O
1:ETEBAC 3, 2:FTP, 3:PESIT
1->9 TABLES DE SESSION
NOM DEVICE X25
01->64 TOT/IN/OUT
X, P, T OU M
1-15 CAR. (NO TRANSPAC)
1-15 CAR. (NO TRANSPAC)
'USER-DATA-FIELD'

PORT . : 05050
DEF FICHER FTP : FFTP01
SSLPARM ID .... :
CONTROLE DES CERTIFICATS :
INTERV.SESS ,TRF : , MINUTES

OPTION : MODIFIER MAJ : 10/04/01 15:33 gcz
-ENTER- CHAMP SUIVANT -F3- ANNULATION -F8- VALIDATION
```

Pour le partenaire PFTP01, le fichier apmfte contient la définition ID=PFTP01.

```
ID=PFTP01,Ftp_Extended_pftp01,Ftp_pftp01_Extended_Password
ID=alias,my_alias,my_alias_Password
ID=SPECID,my_user@address_of_server_destination.com
```

Le paramètre DPCSID ALIAS peut être modifié au moment de soumettre la requête de transert, depuis STERM ou ITOM.

```
$TOM_DIR/itom/plb8preq "/SFN=FILE01/DIR=T/SPN=PFTP01/SID=$$SPECID"
"/DSN=\$TOM_DIR/config/sysin"
```

Pour la requête ci dessus, le fichier apmfte contient la définition ID=SPECID.

Si un champ est omis pour la définition courante, dans le fichier apmfte, la valeur pour ce champ est prise à partir de la définition du partenaire au répertoire, ou du fichier sysin. Dans l'exemple précédent, la définition SPECID ne fournit pas de mot de passe : le mot de passe sera donc PASSWD pris au répertoire. De la même façon, si l'identifiant n'est pas fourni par le fichier apmfte, c'est le champ DPCSID ALIAS qui est pris, ou, par défaut, le champ DPCSID du fichier sysin.

L'utilitaire P1b8pe2e

L'utilitaire de bout en bout, appelé p1b8pe2e, vous permet de rediriger et d'acquitter des fichiers et des messages.

Acquitter un transfert

Si une requête est présente dans le fichier RENC, on peut acquitter le transfert à partir de son numéro.

```
$TOM_DIR/itom/plb8pe2e "/FUN=E/REQ=10400065/SPN=adjacent" "/ACK='feedback message'"
```

Le paramètre SPN est nécessaire si le nœud initial n'est pas adjacent. Si la requête a été purgée, il faut fournir tous les paramètres du transfert initial.

```
$TOM_DIR/itom/plb8pe2e "/FUN=E /SPN=adjacent"  
"/P12=file/P11=XX/P03=oo/P04=dd/P13=id/p51=dh/p61=cc/p62=bb"  
"/ACK='feedback message'"
```

Rediriger un transfert

Si une requête est présente dans le fichier RENC, on peut rediriger le transfert à partir de son numéro.

```
$TOM_DIR/itom/plb8pe2e "/FUN=F/REQ=10400065/SPN=adjacent"
```

Le paramètre SPN est nécessaire.

Si la requête a été purgée, il faut fournir tous les paramètres du transfert initial.

```
$TOM_DIR/itom/plb8pe2e "/FUN=F,TYP=N/SPN=adjacent"  
"/P12=file/P11=XX/P03=oo/P04=dd/P13=id/p51=dh/p61=cc/p62=bb"
```

Description de P1b8pe2e

Ce paragraphe fournit les règles de syntaxe et les paramètres supportés par l'utilitaire p1b8pe2e.

P1b8pe2e peut recevoir de 1 à 5 paramètres, selon le type de fonction utilisée et la manière dont le transfert est défini. Le paramètre N°1 peut fournir des paramètres généraux de requête de transfert comme la priorité, l'option de notification, le type de lien, les dates et heures d'exécution etc. ...

Le tableau ci dessous fournit la liste des paramètres et sous-paramètres, leur description et les règles de syntaxe à respecter.

EERP - Requête

Cette requête fait référence à la requête initiale, par le sous-paramètre /REQ=. Maximum 2 paramètres.

Paramètre	Sous-paramètre	Description	Obligatoire/Défaut
#1	/FUN	Fonction E=EERP	Oui
	/REQ	Numéro de requête, 8 caractères numériques. Exemple : /REQ=09800005	Oui
	/SPN	Nom du partenaire distant (adjacent)	Oui
	/SID	Nom local (alias)	RPAR/sysin
	/PSW	Mot de passe local	RPAR/sysin
	/NTF	Option de notification	RFIC
	/PRT	Priorité	RFIC
	/LNK	Type de lien	RPAR
	/DAT	Date d'exécution	Immédiat
	/FAG	Option File Agent	'N'
#2		Message d'acquiescement (message ou fichier) défaut pris dans la définition \$\$EERP\$\$. Compte rendu (message) Compte rendu (fichier)	\$\$EERP\$\$
/ACK			
/DSN			

EERP - Définition d'un transfert

Cette requête fournit les informations du transfert initial. Le sous-paramètre /REQ= est absent, le paramètre N°2 fournit la définition du transfert. Le paramètre N°3 fournit le compte rendu éventuel. Maximum 3 paramètres.

Paramètre	Sous-paramètre	Description	Obligatoire/Défaut
#1	/FUN	Fonction E=EERP	Oui
	/SPN	Nom du partenaire distant (adjacent)	Oui
	/SID	Nom local (alias)	RPAR/sysin
	/PSW	Mot de passe local	RPAR/sysin
	/NTF	Option de notification	RFIC
	/PRT	Priorité	RFIC
	/LNK	Type de lien	RPAR
	/DAT	Date d'exécution	Immédiat
	/FAG	Option File Agent	'N'
#2		Définition du transfert	Oui
	/ORG	Origine du transfert. 1 à 8 caractères alphanumériques. (pi3) Exemple: /ORG=Orgtrf01	Oui
	/DST	Destination du transfert. 1 à 8 caractères alphanumériques. (pi4) Exemple: /DST=DSTtrf01	Oui
	/P11	Type de fichier. 4 caractères hexadécimaux. (pi11) Exemple: /P11=01FA	Oui
	/P12	File name. 1 à 8 caractères alphanumériques. (pi12) – RFIC définition. Exemple: /P12=Ftest01	Oui
	/P13	Transfer identification. 1 à 8 caractères alphanumériques. (pi13) Exemple /P13=18	Oui
	/P51	File creation date: 12 caractères numériques. Exemple: /P51=040110092503	Oui
	/P61	Transfer sender: 0 à 24 caractères. (pi61) Exemple: /P61=Nom du client	Oui
	/P62	Transfer receiver: 0 à 24 caractères. (pi62) Exemple: /P62=nom du service	Oui
#3		Message d'acquiescement (message ou fichier) défaut pris dans la définition \$\$EERP\$\$. Compte rendu (message) Compte rendu (fichier)	\$\$EERP\$\$
/ACK			
/DSN			

Rediriger une requête

Cette requête fait référence à la requête initiale, par le sous-paramètre /REQ=. Seul le paramètre N°1 est fourni. Les paramètres /DSN, /P99, /LAB sont invalides, car ces informations sont récupérées dans l'enregistrement de la requête dans le fichier RENC.

Paramètre	Sous-paramètre	Description	Obligatoire/Défaut
#1	/FUN	Fonction F=Forward	Oui
	/REQ	Numéro de requête, 8 caractères numériques. Exemple: /REQ=09800005	Oui
	/SPN	Nom du partenaire distant (adjacent)	oui
	/SID	Nom local (alias)	RPAR/sysin
	/PSW	Mot de passe local	RPAR/sysin
	/NTF	Option de notification	RFIC
	/PRT	Priorité	RFIC
	/LNK	Type de lien	RPAR
	/DAT	Date d'exécution	Immédiat
	/FAG	Option File Agent	'N'

Rediriger une définition de transfert

Cette requête fournit les informations du transfert initial. Le sous-paramètre /REQ= est absent, le paramètre N°2 fournit la définition du transfert. Les paramètres N°3, 4 et 5 fournissent les éventuels paramètres complémentaires, en particulier le nom physique de fichier.

Paramètre	Sous-paramètre	Description	Obligatoire/Défaut
#1	/FUN	Fonction F=Forward	Oui
	/SPN	Nom du partenaire distant (adjacent)	Oui
	/SID	Nom local (alias)	RPAR/sysin
	/PSW	Mot de passe local	RPAR/sysin
	/NTF	Option de notification	RFIC
	/PRT	Priorité	RFIC
	/LNK	Type de lien	RPAR
	/DAT	Date d'exécution	Immédiat
	/FAG	Option File Agent	'N'
#2		Définition du transfert	Oui
	/ORG	Origine du transfert. 1 à 8 caractères alphanumériques. (pi3) Exemple: /ORG=Orgtrf01	Oui
	/DST	Destination du transfert. 1 à 8 caractères alphanumériques. (pi4) Exemple: /DST=DSTtrf01	Oui
	/P11	Type de fichier. 4 caractères hexadécimaux. (pi11) Exemple: /P11=01FA	Oui
	/P12	File name. 1 à 8 caractères alphanumériques. (pi12) – RFIC definition. Exemple: /P12=Ftest01	Oui
	/P13	Transfer identification. 1 à 8 caractères alphanumériques. (pi13) Exemple /P13=18	Oui
	/P51	File creation date: 12 caractères numériques. Exemple: /P51=040110092503	Oui
	/P61	Transfer sender: 0 à 24 caractères. (pi61) Exemple: /P61=Nom du client	Oui
	/P62	Transfer receiver: 0 à 24 caractères. (pi62) Exemple: /P62=nom du service	Oui
#3 #4 #5	/DSN /P99 /LAB	Nom physique de fichier Données utilisateur Label du fichier	RFIC RFIC

Codes Erreur

Ce paragraphe fournit la liste des codes rendus par l'utilitaire p1b8pe2e. Le code retour est un nombre de 4 chiffres, structuré de la façon suivante :

Champ	Définition
1	1 caractère numérique, numéro de paramètre, de 1 à 5
2	2 caractères numériques, numéro de sous-paramètre. 00 Autre (inconnu) 01 Priorité 02 Direction 03 Lien 04 Partenaire 05 Fichier 06 Nom physique 07 UDF 08 Date 09 Moniteur 10 Numéro de requête 11 Alias Name 12 Alias Password 13 Format d'enregistrement 14 Longueur d'enregistrement 15 Champ Api 16 Etat 17 Type de requête 18 Type/Struct/Mode FTP 19 Store/Unique FTP 20 Flag File agent Y/N 21 Label 22 Pi99 254 23 User Origine 24 User Destination 25 Pi61 26 Pi62 27 Quantième 28 Notification 29 Eerp/snf pi11 30 Eerp/snf pi12 31 Eerp/snf pi13 32 Eerp/snf pi51 33 Eerp ACK 34 Eerp ou FWD
3	1 caractère numérique: code erreur: 1 Champ invalide 2 Champ dupliqué 3 Longueur de champ invalide 4 Champ obligatoire absent

Exemple : 2331 signifie que, dans le paramètre N°2, le sous-paramètre pi13 a une longueur invalide.

Nouvelles variables, nouveaux paramètres et codes retour

Les dernières modifications et nouvelles fonctionnalités ont abouti à l'introduction de nouveaux champs.

Variable Symbolique

La variable &IDT est remplacée par le pi13 , l'identification de transfert, qui est véhiculée par le mécanisme de store and forward et l'acquittement de bout en bout. Cette variable est traitée dans le nom physique de fichier en réception, dans le label et le Pi99 dans les deux sens.

Paramètres des commandes utilisateur

Tous les paramètres nécessaires aux échanges de bout en bout sont passés en paramètres de la commande, ce qui permet la mise en place d'automatismes. Le type de requête est ajouté afin de différencier les transferts de fichier, de message et d'EERP.

1. Numéro de requête de transfert
2. Nom symbolique de fichier
3. Nom symbolique de partenaire
4. Nom physique de fichier (chemin complet)
5. Sens de transfert (R,T)
6. Code retour système (SRC)
7. Code retour Connect:Express (TRC)
8. Code retour Protocole (PRC)
9. Pi99 reçu
10. Pi99 émis
11. Origine du transfert
12. destination du transfert
13. Nom local
14. Label
15. Client (Pi61)
16. Service (Pi62)
17. Date de début
18. Heure de début
19. Etat du transfert
20. Quantième
21. Nombre d'enregistrements
22. Nombre d'octets
23. Date de fin
24. Heure de fin
25. Type de requête (N,I,H,M,E)
26. Type de fichier (4 caractères hexadécimaux)
27. Identification du transfert (8 caractères décimaux)
28. Date-heure de création du fichier (12 caractères)

Si le paramètre **5.** (Sens de transfert) est 'R' et le paramètre **25.** (Type de requête) est 'N' ou 'M', le fichier ou message peut être acquitté à partir des paramètres **2. , 3. , 11., 12. , 15. , 16. , 26. , 27. , 28. .**

Si le paramètre **5.** (Sens de transfert) est 'R' et le paramètre **25.** (Type de requête) est 'N' ou 'M' ou 'E', le fichier, le message, ou l'EERP peut être redirigé, avec le même type de requête **20. ,** à partir des paramètres **2. , 3. , 11., 12. , 15. , 16. , 26. , 27. , 28..**

Codes TRC

- 2050 Une requête de transfert d'EERP ou de redirection est soumise pour une requête précédente non terminée.
- 2051 Une requête de transfert d'EERP ou de redirection est soumise pour une requête en émission, ou reçue pour une requête en réception.
- 2053 Une requête de transfert d'EERP est soumise ou reçue pour un transfert d'EERP précédent
- 2055 Une requête de transfert d'EERP ou de redirection est soumise pour une requête précédente déjà acquittée.

- 2057 Le DN objet du partenaire est invalide.
- 2058 Le DN racine du partenaire est invalide.
- 2059 L'identifiant de contrôle du certificat n'est pas défini.

- 2219 La définition d'identifiant FTP étendu n'existe pas.

