



Connect:Express

Interface Applicative Java

Version 1.3.6

Sterling Commerce
An IBM Company

Connect:Express Interface Applicative Java

Version 1.3.6 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 Etats-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 CONCÉDÉE, Y COMPRIS LES GARANTIES DE QUALITÉ MARCHANDE ET DE CONVENANCE À 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 Etats-Unis.

Copyright © 2007. 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.

Contenu

CONTENU	2
PRÉFACE	1
INTRODUCTION	1
GÉNÉRALITÉS	2
EXEMPLE D'UTILISATION	2
CLASSES DE L'API	3
TYPE D'OS	4
EXCEPTIONS	5
<i>CxConnectionException</i>	6
<i>CxLogonException</i>	6
<i>CxInvalidArgumentException</i>	6
<i>CxServerException</i>	6
LISTES	7
STATISTIQUES	8
LIMITATION DU NOMBRE D'ENREGISTREMENTS RETOURNÉS	9
TRACES	10
SUBSTITUTION DES VARIABLES D'ENVIRONNEMENT UNIX	10
REQUÊTE DE TRANSFERT	10
AFFICHAGE DES CERTIFICATS X509	11
EXEMPLES	15
EXEMPLE 1: PARTENAIRES, FICHIERS SYMBOLIQUES, TABLES	15
EXEMPLE 2: STATISTIQUES	20
EXEMPLE 3: CONFIGURATION STATIQUE DU SERVEUR	23
EXEMPLE 4: JOURNAL	23
EXEMPLE 5: SOUMISSION D'UNE REQUÊTE DE TRANSFERT	23
EXEMPLE 6: VARIABLES D'ENVIRONNEMENT D'UN SERVEUR UNIX	23
EXEMPLE 7: PARAMÈTRES DE CLIENTS OU DE SERVEURS SSL	23
EXEMPLE 8: CERTIFICATS X509	23
CONFIGURATION SERVEUR CONNECT:EXPRESS	25
CONNECT:EXPRESS UNIX	25
CONNECT:EXPRESS WINDOWS	25
CONNECT:EXPRESS OS/390	25
ANNEXE. DONNÉES DE CONNECT:EXPRESS	26
DONNÉES D'UN PARTENAIRE SYMBOLIQUE	26
DONNÉES D'UN FICHIER SYMBOLIQUE	27
DONNÉES D'UNE TABLE DE SESSION	28
DONNÉES D'UNE TABLE DE PRÉSENTATION	29
PARAMÈTRES DE SOUMISSION DE REQUÊTE	29
DONNÉES DU JOURNAL	31
DONNÉES D'UN TRANSFERT EN COURS	33
DONNÉES DE CONFIGURATION DU MONITEUR	34
PARAMÈTRES DE CLIENTS SSL	36
PARAMÈTRES DE SERVEURS SSL	37

Préface

Ce document décrit l'utilisation de l'API java de Connect:Express. Celle-ci permet d'intégrer à des applications clientes l'accès distant aux diverses fonctionnalités d'un moniteur.

Le Chapitre 1 donne une description générale de l'API.

Le Chapitre 2 fournit des exemples de programmes utilisant l'API.

L'Annexe 1 décrit les données des moniteurs Connect:Express.

La présente version de l'API s'applique pour le moment uniquement à des moniteurs Connect:Express Unix et Windows. Des versions ultérieures permettront de s'adresser à des moniteurs Connect:Express OS/390. On trouvera, néanmoins, ci-dessous des références au comportement de l'API pour chaque type de moniteur.

L'API java de Connect:Express nécessite une version du JDK supérieure ou égale à JDK 1.4.2.

Introduction

L'API java de Connect:Express (CXJAI) permet de contrôler par programme des moniteurs Connect:Express Unix, Windows et OS/390 distants.

Chaque moniteur distant (serveur) est représenté par une classe CxServer fournissant des méthodes permettant d'interroger et d'agir sur les diverses composantes du moniteur:

- Interrogation, création, modification et suppression de partenaires symboliques et fichiers symboliques.
- Interrogation et modification des tables de session et de présentation.
- Interrogation des éléments de configuration statique du moniteur.
- Interrogation des transferts actifs, du journal des transferts et des statistiques.
- Soumission de requêtes de transfert.
- Suspension, reprise et purge de transferts.

La connexion aux moniteurs Connect:Express distants se fait par TCP/IP.

Les différentes fonctions sont abordées une à une dans la suite après une présentation générale des principes d'utilisation de l'API. La description détaillée de l'API se trouve dans le fichier javadoc CXJAIDOC.jar.

Les éléments fournis sont les suivants:

- CXJAI.jar Classes de l'api
- CXJAIDOC.jar Documentation javadoc de l'api
- CXJAI_guide.pdf Le présent document
- CXJAI_examples.jar Exemples de programmes utilisant l'API

Ce chapitre décrit d'une manière générale l'utilisation de l'API.

Généralités

Pour accéder à un moniteur, l'application effectue les opérations suivantes:

- Connexion au moniteur
- Interrogations, actions
- Déconnexion

La connexion de l'application à un moniteur s'effectue par l'instanciation d'une classe `CxServer`, en fournissant en paramètre du constructeur, les éléments nécessaires aux connexions réseaux et protocolaires.

Les diverses opérations sur le moniteur se font par appel des méthodes appropriées de la classe `CxServer`. Chaque composante du moniteur est définie par une classe permettant l'accès à ses divers éléments. Ces classes interviennent dans les valeurs fournies (requêtes) ou retournées (réponses) lors de l'utilisation des méthodes de `CxServer`.

Une fois les opérations effectuées, l'application se déconnecte du moniteur par appel de la méthode `disconnect()` de `CxServer`.

Exemple d'utilisation

L'exemple ci-dessous illustre les différents appels de l'API décrits ci-dessus:

```
...
import com.sterlingcommerce.cx.sdk.*;
...

...
//Connect to server
Char[] pwd = {'A','D','M','I','N'};
String c_host = "192.168.0.18";
int c_port = 9000;
CxServer = new CxServer(c_host,c_port,"ADMIN",pwd,"TCP/IP",0,true);

// Get symbolic partner BOUCLE
CxPartner partner = srv.getPartner("BOUCLE");
System.out.println("Protocol = "+partner.getProtocol());

// Get symbolic file FILE01
CxFile file = srv.getFile("FILE01");
System.out.println("LocalPhysicalName = "+file.getLocalPhysicalName());
```

```
// Disconnect from server
srv.disconnect();
...
```

Classes de l'API

Le tableau suivant donne la liste des différentes classes et interfaces de l'API:

Classe / interface	Description
CxActiveTransfer	Décrit un transfert actif de Connect:Express
CxActivityFilter	Définit les critères de sélection à utiliser pour demander une liste des transferts actifs
CxActivityListElement	Décrit un élément d'une liste de transferts actifs retournée par le serveur
CxAPKey	Décrit les éléments de la clé de protection logicielle du serveur Connect:Express
CxConnectionInformation	Fournit des indications sur la connexion actuelle de l'API avec le serveur
CxEtebac3Presentation	Décrit une table de présentation Etebac3 d'un moniteur Connect:Express Windows
CxFile	Décrit un fichier symbolique
CxJournal	Décrit un enregistrement journal d'une requête de transfert
CxJournalFilter	Définit les critères de sélection à utiliser pour demander une liste des transferts du journal
CxJournalListElement	Décrit un élément d'une liste de transferts du journal retournée par le serveur
CxPartner	Décrit un partenaire symbolique
CxPresentation	Décrit une table de présentation
CxPurgeFilter	Définit les critères de sélection à utiliser pour demander une purge de transferts
CxServer	Classe principale identifiant la connexion au moniteur distant
CxServerConfig	Décrit les éléments de configuration du moniteur
CxSession	Décrit une table de session
CxStatistics	Décrit un enregistrement de statistiques
CxStatisticsFilter	Définit les critères de sélection à utiliser pour demander un ensemble d'enregistrements de statistiques
CxTransferRequest	Définit les paramètres à utiliser pour effectuer une requête de transfert
CxCertificateFilter	Définit les critères de sélection pour demander une liste de certificats X509 ou obtenir la description d'un certificat
CxCertificateListElement	Décrit un élément d'une liste de certificats X509
CxCertificate	Décrit un certificat X509
CxSslClientParameter	Décrit l'ensemble des paramètres d'un client SSL

Classe / interface	Description
CxSslServerParameter	Décrit l'ensemble des paramètres d'un serveur SSL
CxEnumeration	Enumération d'un ensemble d'enregistrements de statistiques retourné par le serveur
CxException	Classe racine des exceptions levées par l'API
CxConnectionException	Exception d'erreur de connexion
CxLogonException	Exception d'erreur de logon
CxServerException	Exception levée lorsque l'API a reçue une indication d'erreur du serveur (hors logon)

Type d'OS

Bien que fortement similaires dans leur ensemble, les différentes données des moniteurs Connect:Express ne présentent pas dans le détail toutes les mêmes caractéristiques suivant le type de système d'exploitation (Unix, Windows ou OS/390).

Par exemple la longueur maximum d'un nom physique de fichier (PhysicalName) est de 44 caractères pour Connect:Express OS/390 et Unix alors qu'elle est de 127 caractères pour Connect:Express Windows.

Par ailleurs, les moniteurs ne présentent pas tous les mêmes fonctionnalités suivant la plateforme.

Par exemple le protocole FTP est disponible avec Connect:Express OS/390 et Unix mais pas avec Connect:Express Windows. De même le protocole réseau LU6.2 est disponible avec Connect:Express OS/390 et Windows mais pas avec Connect:Express Unix.

L'API doit nécessairement tenir compte de ces différences.

Lorsque l'API se connecte au serveur, elle obtient de celui-ci une chaîne indiquant le type d'OS du serveur: "WINDOWS", "UNIX" ou "OS390".

```
...
    String ostype = srv..getOSType();
...
```

La documentation javadoc de l'API (fournie dans CXJAIDOC.jar) indique, pour chaque méthode, le type d'OS pour lequel elle est valide.

Par ailleurs, la documentation de chaque méthode "setter" indique la longueur maximum du paramètre en fonction du type d'OS quand c'est nécessaire.

Par exemple, la méthode `setEndTransmitCommand` de la classe `CxFile` est documentée de la manière suivante:

setEndTransmitCommand

```
public void setEndTransmitCommand(java.lang.String endTransmitCommand)
    throws CxInvalidArgumentException
```

Sets the user command called at end of transmission.

OS: Windows, Unix, OS390.

Windows: Maximum length: 127 characters.

Unix: Maximum length: 12 characters.

OS390: Maximum length: 32 characters.

Parameters:

`endTransmitCommand` - the name of the command.

Throws:

`CxInvalidArgumentException` - if the length of the parameter is invalid.

L'API contrôle la validité de l'appel de méthode "setter" ainsi que la longueur du paramètre en fonction du type d'OS du serveur. En cas d'erreur, une exception `CxInvalidArgumentException` est levée.

En ce qui concerne les méthodes "getter", aucune exception n'est levée si la donnée correspondante n'est pas définie pour la plateforme et une valeur nulle est retournée (String "", int 0, char ' ').

Ceci permet d'écrire une application de consultation en partie transparente du point de vue des types d'OS des serveurs, puisque, par exemple, si une donnée String n'est pas définie, elle s'affichera comme une chaîne vide.

L'annexe 1 présente sous forme de tableau la validité et la longueur de chaque donnée de Connect:Express en fonction du système d'exploitation.

Exceptions

Les exceptions de l'API dérivent de la classe racine `CxException`, elle-même dérivée de la classe `java.lang.Exception`. Ces exceptions sont les suivantes:

CxConnectionException

Cette exception est levée en cas d'échec de la connexion réseau au serveur, lors d'un incident réseau en cours de connexion, ou en cas d'erreur protocolaire dans le dialogue entre l'API et le serveur.

Si cette exception survient, il est nécessaire de rétablir la connexion ultérieurement, par l'instanciation d'une nouvelle classe CxServer.

CxLogonException

Cette exception est levée en cas d'échec lors de l'identification de l'utilisateur auprès du serveur. Si cette exception survient, il est nécessaire de se reconnecter par une nouvelle instance de CxServer avec un nom d'utilisateur et un mot de passe corrects.

Note: Actuellement, le contrôle d'identification n'est effectif que lorsqu'on s'adresse à un moniteur Connect:Express Windows.

CxInvalidArgumentException

Cette exception est levée en cas d'erreur de paramètre d'une des méthodes de l'API. Cette exception résulte d'un contrôle local par l'API des paramètres fournis par le programme appelant.

Si cette exception survient, il est nécessaire de corriger le programme appelant.

CxServerException

Cette exception est levée lorsque le moniteur retourne une erreur sur une connexion valide. Cette erreur correspond à une erreur "logique". Par exemple:

- Création d'un fichier symbolique existant
- Erreur dans le dépôt d'une requête de transfert

Dans ce cas l'exception contient, dans son champ mid (message id), la valeur du code d'erreur indiqué par le moniteur (Il s'agit en fait d'un code "TRC" Connect:Express) ainsi qu'un message explicatif dans la langue locale par défaut.

Le message id peut être obtenu par la méthode CxServerException.getMid().

Le message peut être obtenu par la méthode CxServerException.getMessageText().

Les guides utilisateurs de Connect:Express sur les différentes plates formes indiquent la signification des TRC pouvant être retournés par le moniteur.

Note: En principe les méthodes de la classe CxServer capturent les autres exceptions java afin de ne lever que les exceptions décrites ci-dessus. Il est néanmoins préférable, afin de se garder de toute mauvaise éventualité, que l'application appelante utilise en dernier recours un bloc catch (java.lang.Exception) trasant la pile.

La capture des exceptions par le programme appelant pourra se faire de la manière suivante:

```
CxServer srv = null;
try {
    ...
    //Connect to server
```

```

Char[] pwd = {'A','D','M','I','N'};
String c_host = "192.168.0.18";
int c_port = 9000;
srv = new CxServer(c_host,c_port,"ADMIN",pwd,"TCP/IP",0,true);

// Get symbolic partner BOUCLE
CxPartner partner = srv.getPartner("BOUCLE");
System.out.println("Protocol = "+partner.getProtocol());

// Get symbolic file FILE01
CxFile file = srv.getFile("FILE01");
System.out.println("LocalPhysicalName = "+file.getLocalPhysicalName());

// Disconnect from server
srv.disconnect();
srv = null;
...
} catch (CxServerException e) {
// Depending on the server message id, processing can continue or not
// See documentation on the Trc codes that can be returned by Connect:Express
System.out.println("Message id = "+e.getMid());
System.out.println("Message text = "+e.getMessageText());
...
} catch (CxLogonException e) {
// Instantiate a new CxServer with different user/password ?
...
} catch (CxConnectionException e) {
// Reconnect ?
...
} catch (CxInvalidArgumentException e) {
// Fatal error. Correct the program
...
} catch (Exception e) {
// Fatal error
e.printStackTrace();
} finally {
if (srv!=null) {
Try {
Srv.disconnect();
} catch (Exception) {
}
}
}
...

```

Listes

La classe CxServer permet d'obtenir des listes pour les éléments suivants:

- Liste des partenaires symboliques (listPartners)
- Liste des fichiers symboliques (listFiles)
- Liste des tables de sessions (listSessions)
- Liste des tables de présentation (listPresentations)
- Liste des tables de présentation Etebac3 Connect:Express Windows (listEtebac3Presentations)
- Liste des transferts en cours (selectActivity)
- Liste du journal (selectJournal)

- Liste de certificats (selectCertificates)
- Liste de paramètres de clients SSL (listSslClientParameters)
- Liste de paramètres de serveurs SSL (listSslServerParameters)

Les différentes méthodes renvoient des objets de type array qui peuvent contenir soit simplement une liste d'identifiants (par exemple liste des nom de partenaires), soit un ensemble d'éléments plus complets (éléments de liste de transfert en cours ou éléments de liste du journal).

Une fois obtenue une liste d'éléments il est possible d'obtenir du serveur le détail d'un élément donné.

Les demandes de listes sont illustrées ci-dessous:

```

...
// PARTNERS
// Get the list of the symbolic partners names
String[] = srv.listPartners();

// For each partner name, get the detail of the partner
for (i=0;i<partnerlist.length;i++) {
    // Get the detail of each partner
    CxPartner partner = srv.getPartner(partnerlist[i]);
    ...
}

// JOURNAL
// Get the list of all elements of the journal
// Selection criteria
CxJournalFilter filter = new CxJournalFilter(CxServer.UNIX);
filter.setFileName("");
filter.setPartnerName("");
filter.setDirection("");
filter.setMinimumDate("");
filter.setMaximumDate("");
filter.setStatus("");

CxJournalListElement[] journallist = srv.selectJournal(filter);
// Each journal element already contains basic information
// such as requestNumber,...

for (i=0;i<journallist.length;i++) {
    CxJournalListElement elem = journallist[i];
    // Get the full detail of the journal for this element
    // The key is the request number
    CxJournal jnl = srv.getJournal(elem.getRequestNumber());
    ...
}
...

```

Statistiques

Les statistiques sont obtenues en fournissant, soit des critères de sélection (CxStatisticsFilter), soit un nombre de

secondes, à la méthode `selectStatistics` de la classe `CxServer`.

Le filtre permet d'effectuer la sélection en fonction d'une plage de dates et éventuellement d'un n° de requête. Le nombre de secondes permet de ne sélectionner que les dernières statistiques dans la plage du nombre de secondes indiqué.

Les résultats sont obtenus sous forme d'une énumération, chaque élément de l'énumération étant une ligne de statistique `CxStatistics`.

Une ligne de statistiques est constituée d'un ensemble d'éléments présentés sous la forme mot-clé=valeur séparés par des virgules.

Des méthodes appropriées de la classe `CxStatistics` permettent d'extraire sous la forme de tableaux les différents éléments de chaque ligne.

L'exemple suivant décrit l'interrogation des statistiques d'un serveur:

```

...
// Define minimum and maximum date
Date minimumDate = ...;
Date maximumDate = ...;

// Define a statistics filter for the selection
CxStatisticsFilter filter = new CxStatisticsFilter(CxServer.UNIX);
filter.setMinimumDate(minimumDate);
filter.setMaximumDate(maximumDate);

CxStatistics stat;
CxEnumeration senum = srv.selectStatistics(filter);
while (senum.hasMoreElements()==true) {
    Stat = (CxStatistics)senum.nextElement();
    System.out.println("statline = "+stat.toString());
    String[][] lvp = stat.getLabelValuePair();
    for (int ii=0;ii<keys.length;ii++) {
        System.out.println(lvp[ii][0]+" = "+lvp[ii][1]);
    }
}
...

```

Note:

A l'intérieur de la boucle `while` ci-dessus, la liaison TCP/IP avec le serveur est occupée. En effet tant que `hasMoreElements()` retourne `true`, l'API est susceptible de recevoir des lignes de statistiques du serveur.

Il n'est donc pas possible, à l'intérieur de cette boucle d'utiliser d'autres méthodes de l'API que `nextElement()`.

Limitation du nombre d'enregistrements retournés

L'obtention d'un trop grand nombre d'éléments de journal ou de statistiques peut être pénalisante en terme de ressources et de temps de réponse pour l'application cliente et pour le serveur distant. Il est possible de fixer une limite au nombre d'éléments retournés à l'aide de la méthode `setLimit` de `CxServer`.

Deux méthodes sont impactées par `setLimit`:

- `CxServer.selectJournal`: Nombre maximum d'éléments pouvant être retournés dans la liste.
- `CxServer.selectStatistics`: Nombre maximum de lignes statistiques pouvant être retournées dans l'énumération.

Traces

Il est possible d'activer ou de désactiver les traces de l'api en positionnant un drapeau de trace:

- Dans le constructeur de la classe CxServer
- Dynamiquement par l'appel de la méthode CxServer.setTrace

La trace est dirigée vers un fichier cxjai.<adress-ip-du-serveur>.<port-du-serveur>.trc dans le répertoire courant d'exécution de l'application ou vers la sortie utilisateur suivant qu'un fichier témoin cxjaitrc.flag est présent dans le répertoire ou pas.

La trace indique les données réseau échangées avec le serveur.

Substitution des variables d'environnement Unix.

Nom physique local :

Lors de l'interrogation d'un élément du journal (classe CXJournal), un serveur Unix renseigne le nom physique dans 2 propriétés équivalentes (LocalPhysicalName et XlocalPhysicalName). La première propriété contient le nom physique du fichier transféré sans substitution des variables d'environnement, la 2^{ème} avec substitution.

La taille maximale de la chaîne XlocalPhysicalName est de 512 caractères. En cas de dépassement la propriété XlocalPhysicalNameOvf est valorisée à true.

getEnvironmentValue et replaceEnvironment :

La classe CXServer contient 2 méthodes permettant d'accéder aux variables d'environnement du serveur.

getEnvironmentValue retourne la valeur de la variable d'environnement dont le nom est passé en paramètre.

replaceEnvironment prends comme paramètre une chaîne contenant des variables d'environnement et retourne une chaîne équivalente où les variables d'environnement ont été substituées.

Requête de transfert

Le dépôt d'une requête de transfert se fait en définissant l'ensemble des paramètres du transfert dans un objet de la classe CxTransferRequest, puis en appelant la méthode submitTransfer avec cet objet en paramètre.

Les requêtes de transfert Connect:Express ont des ensembles de paramètres spécifiques pour chaque système d'exploitation. L'api contrôle de manière unitaire la validité de chaque paramètre en fonction du type de système d'exploitation du serveur. Pour avoir plus de détails sur les paramètres des requêtes de transfert, consulter le guide utilisateur de Connect:Express pour le système d'exploitation concerné.

L'exemple ci-dessous illustre le dépôt d'une requête de transfert pour Connect:Express Unix:

```

...
// Transfer request
...
// Set the parameters
CxTransferRequest treq = new CxTransferRequest(CxServer.UNIX);
treq.setFileName("FILE01");
treq.setTransferDirection('T');
treq.setPartnerName("BOUCLE");
treq.setLocalPassword("PSW");

```

```

    treq.setPhysicalName("$TOM_DIR/out/file.txt");

    // Submit the transfer
    String[] requestNumber = srv.submitTransfer(treq);

    // Resulting request number
    System.out.println("request number="+requestNumber[0]);

    ...

```

Cet exemple effectue la soumission d'un transfert du fichier physique file.txt, vers le partenaire BOUCLE conformément au fichier symbolique FILE01.

Le numéro de requête résultant est obtenu dans requestNumber[0].

Notes:

- Avec Connect:Express Windows, plusieurs transferts peuvent résulter d'une seule requête (Emission d'un ensemble de fichier). C'est pourquoi le résultat de submitTransfer() est un tableau et non une simple chaîne. Dans le cas de Connect:Express Unix, un seul numéro de requête est obtenu.
- Les erreurs « logiques » dans la soumission de la requête génèrent des exception CxServerException.

Affichage des certificats X509

L'accès aux certificats se fait en lecture uniquement à l'aide des 2 méthodes :

- selectCertificates(CxCertificateFilter filter)
- getCertificate(CxCertificateFilter filter)

de la classe CxServer.

Le paramètre CxCertificateFilter définit les critères de sélection du/des certificats. Ce paramètre prend des formes différentes selon que l'on s'adresse à un serveur Connect:Express Unix ou Windows.

Connect:Express Unix

Les certificats sont importés dans une base de donnée privée, à l'aide de l'utilitaire \$stern. Chaque certificat est alors référencé par un identifiant unique (CertificateId) d'au plus 8 caractères. Les certificats peuvent être des certificats personnels ou des certificats d'autorité racine.

Les paramètres de sélection de CxCertificateFilter sont dans ce cas :

Pour selectCertificates (sélection d'un ensemble de certificats) :

- CertificateId
- CertificateType
- Subject
- Issuer

Ces éléments peuvent être des motifs simples.

Pour getCertificate (détail d'un certificat) :

- CertificateId

Exemples:

```

...
//Get a list of certificates
CxCertificateFilter filter = new CxCertificateFilter(CxServer.UNIX);
filter.setCertificateId("");
filter.setSubject("*.www.caexmpl.com*");
CxCertificatesListElement[] elm = srv.selectCertificates(filter);
...

```

```

...
//Get a list of certificates
CxCertificateFilter filter = new CxCertificateFilter(CxServer.UNIX);
filter.setCertificateId("CAEXMP*");
CxCertificatesListElement[] elm = srv.selectCertificates(filter);
...

```

```

...
//Get a certificate
CxCertificateFilter filter = new CxCertificateFilter(CxServer.UNIX);
filter.setCertificateId("CAEXMP1");
CxCertificates cert = srv.getCertificate(filter);

//Display the certificate as a PEM certificate (Base 64 encoded)
System.out.println(cert.getPEMCertificate());

//Display the certificate characteristics
System.out.println(cert.getCertificateCharacteristics());
...

```

Connect:Express Windows

Les certificats sont importés dans les magasins de certificats du système Windows à l'aide de mmc (Microsoft Management Console).

Les paramètres de sélection de CxCertificateFilter sont dans ce cas :

Pour selectCertificates (sélection d'un ensemble de certificats) :

- StoreLocation
- StoreName
- Subject (chaîne de caractères à rechercher dans le distinguished name du sujet)
- Issuer (chaîne de caractères à rechercher dans le distinguished name de l'émetteur)

StoreName, StoreLocation doivent être obligatoirement renseignés.

Pour getCertificate (détail d'un certificat) :

- StoreLocation
- StoreName
- Subject (Distinguished name complet du sujet ou valeur du common name de ce DN)
- Issuer (Un chaîne facultative devant être reconnue dans le distinguished name de l'émetteur)

StoreName et StoreLocation et Subject doivent être obligatoirement renseignés.

StoreLocation peut prendre l'une des valeurs : « SYSTEM_STORE_LOCAL_MACHINE », « SYSTEM_STORE_CURRENT_USER » ou « SYSTEM_STORE_SERVICES ».

StoreName peut prendre l'une des valeurs « My » (Personal), « Root » (Trusted root authorities), « CA » (Intermediate certification authorities) ou Trust (Enterprise trust).

Exemples :

```

...
//Get a list of certificates
CxCertificateFilter filter = new CxCertificateFilter(CxServer.WINDOWS);
filter.setStoreLocation("SYSTEM_STORE_LOCAL_MACHINE");
filter.setStoreName("Root");
filter.setSubject("Microsoft");
filter.setIssuer("O=Microsoft Trust Network");
CxCertificateListElement[] elm = srv.selectCertificates(filter);
...

```

```

...
//Get a certificate
CxCertificateFilter filter = new CxCertificateFilter(CxServer.WINDOWS);
filter.setStoreLocation("SYSTEM_STORE_LOCAL_MACHINE");
filter.setStoreName("Root");

//Value of the common name of the subject dn
filter.setSubject("Microsoft Root Certificate Authority");

CxCertificates cert = srv.getCertificates(filter);

//Display the certificate as a PEM certificate (Base 64 encoded)
System.out.println(cert.getPEMCertificate());

//Display the certificate characteristics
System.out.println(cert.getCertificateCharacteristics());
...

```

```

...
//Get a certificate
CxCertificateFilter filter = new CxCertificateFilter(CxServer.WINDOWS);
filter.setStoreLocation("SYSTEM_STORE_LOCAL_MACHINE");
filter.setStoreName("Root");

//Complete distinguished name of the subject in reverse order
filter.setSubject(
    "CN=Microsoft Root Certificate Authority, DC=microsoft, DC=com");

CxCertificates cert = srv.getCertificate(filter);

//Display the certificate as a PEM certificate (Base 64 encoded)

```

```
System.out.println(cert.getPEMCertificate() );  
  
//Display the certificate characteristics  
System.out.println(cert.getCertificateCharacteristics() );  
...
```

Exemples

Ce chapitre donne des exemples complets de programmes utilisant l'API pour un serveur Connect:Express Unix. Chaque exemple admet comme 2 premiers paramètres: l'adresse et le port du serveur.

Les exemples 1 à 4 et 6 à 8 sont utilisables tel quels avec n'importe quel serveur Connect:Express Unix. L'exemple 5 dépend de l'existence préalable d'un partenaire BOUCLE et d'un fichier symbolique FILE01.

Les sources ci-dessous sont fournis dans le fichier CXJAI_exemples.jar.

Pour exécuter les exemples, extraire CXJAI_exemples.jar et recopier CXJAI.jar dans le répertoire bin obtenu. Puis lancer les exemples par les scripts examplex.sh (bat) en fournissant les paramètres adéquats.

Par exemple:

```
./example1.sh 10.87.15.106 9000 (Unix)
example1.bat 10.87.15.106 9000 (Windows)
```

où 10.87.15.105 et 9000 correspondent à l'adresse et au port d'API du moniteur Connect:Express.

Exemple 1: Partenaires, fichiers symboliques, tables.

Ce programme affiche les partenaires, fichiers symboliques et tables de session et présentation.

```

                                                                    /*Example1 (1)*/
/*
 * Example1.java
 * This program displays
 * - the symbolic partners
 * - the symbolic files
 * - the session tables
 * - the presentation tables
 * of a Connect:Express Unix or Windows server
 *
 * The parameters are the IP address/host name and port of the server
 */
import java.util.Date;

                                                                    /*Example1 (2)*/
import com.sterlingcommerce.cx.sdk.*;

public class Example1 {
    public Example1() {
    }
}

/**
```

```

* @param args
* Lists partners, files, tables ...
* param1 = server address
* param2 = server port
*
*/
public static void main(String[] args) {
    CxServer srv = null;
    String OStype = null;

    Date startDate = new Date();
    if (args.length!=2) {
        usage();
        return;
    }
    int uu = 0;
    try {
        uu = Integer.parseInt(args[1].trim());
    } catch (NumberFormatException e) {
        System.out.println("Invalid port");
        return;
    }
    try {
        srv = connectToServer(args[0],uu);
        displayConnectionInfo(srv);
        displayPartners(srv);
        displayFiles(srv);
        displaySessions(srv);
        displayPresentations(srv);
        Date endDate = new Date();
        System.out.println("Elapsed time = "
            +(endDate.getTime()-startDate.getTime())+" ms");

    } catch (CxServerException e) {
        //Depending on the server message id, processing can continue or not
        //See documentation on the MIDs (TRC) that can be returned by
Connect:Express
        System.out.println("Message id = "+e.getMid());
        System.out.println("Message Text = "+e.getMessageText());
        displayErrors(e);
    } catch (CxConnectionException e) {
        //Reconnect ?
        displayErrors(e);
    } catch (CxLogonException e) {
        //Enter userid/password again ?
        displayErrors(e);
    } catch (CxInvalidArgumentException e) {
        //Fatal error
        displayErrors(e);
    } catch (Exception e) {
        //Fatal error
        e.printStackTrace();
    } finally {
    }
        try {
            srv.disconnect();
        } catch (Exception e) {
        }
    }

        /*Example1 (3)*/

    }
} //End main

public static void displayErrors(CxException e) {
    System.out.println("Errors:");
    System.out.println("-----");
    Throwable t = e;
    while (t!=null) {

```



```

        System.out.println(t.getMessage());
        t = t.getCause();
    }
} //End displayErrors

public static CxServer connectToServer(String c_host,int c_port)
                                   throws CxException {
    char[] pwd = {'A','D','M','I','N'};

    CxServer srv = new CxServer(c_host,c_port,"ADMIN",pwd,"TCP/IP",0,false);
    return srv;
} //End connectToServer

public static void displayConnectionInfo(CxServer srv)
                                   throws CxException {
    CxConnectionInformation cInfo = srv.getConnectionInfo();
    System.out.println("Address           = "+cInfo.getAddress());
    System.out.println("Port             = "+cInfo.getPort());
    System.out.println("Userid          = "+cInfo.getUserid());
    System.out.println("NodeName        = "+cInfo.getNodeName());
    System.out.println("OSType          = "+cInfo.getOSType());
    System.out.println("NodeVersion     = "+cInfo.getNodeVersion());
    System.out.println("APKeySupport    = "+cInfo.getAPKeySupport());
} //End displayConnectionInfo

public static void displayPartners(CxServer srv) throws CxException {
    String[] list = null;

    list = srv.listPartners();
    System.out.println("-----");
    System.out.println("SYMBOLIC PARTNERS");
    System.out.println("-----");
    for (int i=0;i<list.length;i++) {
        CxPartner partner = srv.getPartner((String)list[i]);
        displayPartnerDetail(srv,partner);
        System.out.println("-----");
    }
} //End displayPartners

public static void displayPartnerDetail(CxServer srv,CxPartner partner)
                                   throws CxException {
    System.out.println("PartnerName       = "+partner.getPartnerName());
    System.out.println("PartnerPassword   = "+partner.getPartnerPassword());
    System.out.println("LocalName         = "+partner.getLocalName());
    System.out.println("LocalPassword     = "+partner.getLocalPassword());
    System.out.println("PartnerState      = "+partner.getPartnerState());
    System.out.println("TypeOfPartner     = "+partner.getTypeOfPartner());
    System.out.println("Protocol          = "+partner.getProtocol());
    System.out.println("MaxSession        = "+partner.getMaxSession());
    System.out.println("MaxSessionIn     = "+partner.getMaxSessionIn());
    System.out.println("MaxSessionOut    = "+partner.getMaxSessionOut());
    System.out.println("SessionTableId   = "+partner.getSessionTableId());
    System.out.println("TypeOfLink       = "+partner.getTypeOfLink());
    System.out.println("TcpipAddress     = "+partner.getTcpipAddress());
    System.out.println("TcpipPort        = "+partner.getTcpipPort());
    System.out.println("TcpipHostName    = "+partner.getTcpipHostName());
    System.out.println("X25LocalAddress  = "+partner.getX25LocalAddress());

                                   /*Example1 (4)*/

    System.out.println("X25RemoteAddress = "+partner.getX25RemoteAddress());
    System.out.println("X25LocalPort     = "+partner.getX25LocalPort());
    System.out.println("X25UserDataField = "+partner.getX25UserDataField());

    System.out.println("X25Facilities    = "+partner.getX25Facilities());
    String OSType = srv.getConnectionInfo().getOSType();
    if (OSType.equals(CxServer.UNIX)==true) {
        System.out.println("FtpDefaultFile   = "+partner.getFtpDefaultFile());
    }
}

```

```

        System.out.println("RetryNumber          = "+partner.getRetryNumber());
        System.out.println("SessionTimer       = "+partner.getSessionTimer());
        System.out.println("TransferTimer    = "+partner.getTransferTimer());
    } else if (OSType.equals(CxServer.WINDOWS)==true) {
        System.out.println("LocalNameType     = "+partner.getLocalNameType());
        System.out.println("PartnerComment    = "+partner.getPartnerComment());
        System.out.println("RestartUsed       = "+partner.isRestartUsed());
        System.out.println("SnaLuName         = "+partner.getSnaLuName());
        System.out.println("AppcModeName      = "+partner.getAppcModeName());
        System.out.println("AppcTpName        = "+partner.getAppcTpName());
    }
}
} //End displayPartnerDetail

public static void displayFiles(CxServer srv) throws CxException {
    String[] list = null;

    list = srv.listFiles();
    System.out.println("-----");
    System.out.println("SYMBOLIC FILES");
    System.out.println("-----");
    for (int i=0;i<list.length;i++) {
        CxFile file = srv.getFile((String)list[i]);
        displayFileDetail(srv, file);
        System.out.println("-----");
    }
} //End displayFiles

public static void displayFileDetail(CxServer srv,CxFile file)
                                     throws CxException {
    System.out.println("FileName          = "+file.getFileName());
    System.out.println("FileState         = "+file.getFileState());
    System.out.println("TypeOfAllocation  = "+file.getTypeOfAllocation());
    System.out.println("FileDirection    = "+file.getFileDirection());
    System.out.println("TypeOfFile        = "+file.getTypeOfFile());
    System.out.println("FileOpenOption    = "+file.getFileOpenOption());
    System.out.println("FileSender        = "+file.getFileSender());
    System.out.println("FileReceiver      = "+file.getFileReceiver());
    System.out.println("PresentationTableId= "
        +file.getPresentationTableId());
    System.out.println("LocalPhysicalName = "+file.getLocalPhysicalName());
    System.out.println("FileRecordLength  = "+file.getFileRecordLength());
    System.out.println("StartTransmitExit = "+file.getStartTransmitExit());
    System.out.println("EndTransmitExit   = "+file.getEndTransmitExit());
    System.out.println("StartReceiveExit  = "+file.getStartReceiveExit());
    System.out.println("EndReceiveExit    = "+file.getEndReceiveExit());
    System.out.println("StartTransmitCommand= "
        +file.getStartTransmitCommand());
    System.out.println("EndTransmitCommand = "
        +file.getEndTransmitCommand());
    System.out.println("StartReceiveCommand= "+file.getStartReceiveCommand());
    System.out.println("EndReceiveCommand  = "+file.getEndReceiveCommand());
    String OSType = srv.getConnectionInfo().getOSType();
    if (OSType.equals(CxServer.UNIX)==true) {
        System.out.println("Priority          = "+file.getPriority());
        System.out.println("RemotePhysicalName = "+file.getRemotePhysicalName());
        System.out.println("FtpOptions        = "+file.getFtpOptions());
        System.out.println("ParamFileUsed     = "+file.isParamFileUsed());

        /*Example1 (5)*/

        System.out.println("SpaceAllocationUsed= "+file.isSpaceAllocationUsed());
        System.out.println("FtpStoreUniqueUsed = "+file.isFtpStoreUniqueUsed());
        System.out.println("FileAgentUsed      = "+file.isFileAgentUsed());
        System.out.println("TypeOfNotification = "+file.getTypeOfNotification());
    } else if (OSType.equals(CxServer.WINDOWS)==true) {
        System.out.println("FileComment       = "+file.getFileComment());
        System.out.println("ErrorCommand      = "+file.getErrorCommand());
    }
}

```

```

        System.out.println("NotifyUsed           = "+file.isNotifyUsed());
        System.out.println("ClientToNotify       = "+file.getClientToNotify());
        System.out.println("Pi990offsetT   = "+file.getPi990offsetT());
        System.out.println("Pi99LengthT   = "+file.getPi99LengthT());
        System.out.println("Pi99ValueT    = "+file.getPi99ValueT());
        System.out.println("Pi990offsetR   = "+file.getPi990offsetR());
        System.out.println("Pi99LengthR   = "+file.getPi99LengthR());
        System.out.println("Pi99ValueR    = "+file.getPi99ValueR());
        System.out.println("FileLabel     = "+file.getFileLabel());
    }
} //End displayFileDetail

public static void displaySessions(CxServer srv) throws CxException {
    String[] list = null;

    list = srv.listSessions();
    System.out.println("-----");
    System.out.println("SESSION TABLES");
    System.out.println("-----");
    for (int i=0;i<list.length;i++) {
        CxSession session = srv.getSession((String)list[i]);
        displaySessionDetail(srv,session);
        System.out.println("-----");
    }
} //End displaySessions

public static void displaySessionDetail(CxServer srv,CxSession ses)
                                     throws CxException {
    System.out.println("SessionTableId      = "+ses.getSessionTableId());
    System.out.println("BaseMessageSize     = "+ses.getBaseMessageSize());
    System.out.println("BaseSynchronizationSize = "
                                     +sess.getBaseSynchronizationSize());
    System.out.println("CrcUsed              = "+ses.isCrcUsed());
    String OSType = srv.getConnectionInfo().getOSType();
    if (OSType.equals(CxServer.UNIX)==true) {
        System.out.println("BaseWindowSize      = "+ses.getBaseWindowSize());
        System.out.println("ProtocolVersion     = "+ses.getProtocolVersion());
        System.out.println("RetryNumber         = "+ses.getRetryNumber());
    } else if (OSType.equals(CxServer.WINDOWS)==true) {
        System.out.println("SessionDirection   = "+ses.getSessionDirection());
        System.out.println("ResynchronizationNumber = "
                                     +ses.getResynchronizationNumber()
    );
    }
} //End displaySessionDetail

public static void displayPresentations(CxServer srv) throws CxException {
    String[] list = null;

    list = srv.listPresentations();
    System.out.println("-----");
    System.out.println("PRESENTATION TABLES");
    System.out.println("-----");
    for (int i=0;i<list.length;i++) {
        CxPresentation presentation = srv.getPresentation((String)list[i]);
        displayPresentationDetail(srv,presentation);
        System.out.println("-----");
    }

                                     /*Example1 (6)*/

} //End displayPresentations

public static void displayPresentationDetail(CxServer srv,CxPresentation pres)
                                     throws CxException
{
    System.out.println("PresentationTableId =

```

```

"+pres.getPresentationTableId());

    System.out.println("TypeOfCompression      = "+pres.getTypeOfCompression());
    System.out.println("MultiArticleUsed      = "+pres.isMultiArticleUsed());
        System.out.println("TranslationToEbcDic          =
"+pres.getTranslationToEbcDic());
    String OSType = srv.getConnectionInfo().getOSType();
    if (OSType.equals(CxServer.WINDOWS)==true) {
        System.out.println("ConcatenationUsed          =
"+pres.isConcatenationUsed());
        System.out.println("SegmentationUsed      = "+pres.isSegmentationUsed());
        System.out.println("TranslationUsed      = "+pres.isTranslationUsed());
        System.out.println("TranslationToAscii    =
"+pres.getTranslationToAscii());
    }
} //End displayPresentationDetail

static void usage() {
    System.out.println("Usage: java Example1 <ip-address/host-name> <port>\n");
} //End usage

private CxServer srv = null;
} //End class Example1

```

Exemple 2: Statistiques.

Ce programme affiche les statistiques des x dernières secondes, x étant passé en 3^{ème} paramètre de lancement.

```

                                                                    /*Example2 (1)*/

/*
 * Example2.java
 * This program displays the statistics of a Connect:Express Unix or Windows
server
 * for the last seconds indicated as 3rd parameter
 *
 * The parameters are
 * - the IP address/host name of the server
 * - the port of the server
 * - a number of seconds
 *
 */

import java.util.Locale;
import com.sterlingcommerce.cx.sdk.*;

public class Example2 {

                                                                    /*Example2 (2)*/

    public Example2() {

```

```

}

/**
 * @param args
 * Displays statistics
 * param1 = address
 *
 * param2 = port
 * param3 = number of seconds
 */
public static void main(String[] args) {
    CxServer srv = null;

    if (args.length!=3) {
        System.out.println("Invalid number of parameters");
        usage();
    }

    int uu = 0;
    try {
        uu = Integer.parseInt(args[1].trim());
    } catch (NumberFormatException e) {
        System.out.println("Invalid port");
        return;
    }
    int vv = 0;
    try {
        vv = Integer.parseInt(args[2]);
    } catch (NumberFormatException e) {
        System.out.println("The 3rd argument must be a number of seconds");
        return;
    }

    try {
        srv = connectToServer(args[0],uu);
        //srv.setLimit(1000);
        displayStatistics(srv,vv);
    } catch (CxServerException e) {
        //Depending on the server message id, processing can continue or not
        //See documentation on the MIDS (TRC) that can be returned by
Connect:Express
        System.out.println("Message id = "+e.getMid());
        System.out.println("Message Text = "+e.getMessageText());
        displayErrors(e);
    } catch (CxConnectionException e) {
        //Reconnect ?
        displayErrors(e);
    } catch (CxLogonException e) {
        //Enter userid/password again ?
        displayErrors(e);
    } catch (CxInvalidArgumentException e) {
        //Fatal error
        displayErrors(e);
    } catch (Exception e) {
        //Fatal error
        e.printStackTrace();
    } finally {
        try {
            //Disconnect
            srv.disconnect();
        } catch (Exception e) {
        }
    }
}

/*Example2 (3)*/

```

```

} //End main

public static void displayErrors(CxException e) {
    System.out.println("Errors:");
    System.out.println("-----");
    Throwable t = e;
    while (t!=null) {
        System.out.println(t.getMessage());
        t = t.getCause();
    }
} //End displayErrors

public static CxServer connectToServer(String c_host,int c_port)
                                                                    throws
CxException {
    char[] pwd = {'A','D','M','I','N'};

    CxServer srv = new CxServer(c_host,c_port,"ADMIN",pwd,"TCPIP",0,false);
    return srv;
} //End connectToServer

    public static void displayStatistics(CxServer srv,int seconds) throws
CxException {
    CxStatistics stat;

    CxEnumeration senum = srv.selectStatistics(seconds);
    System.out.println("-----");
    System.out.println("STATISTICS");
    System.out.println("-----");
    while (senum.hasMoreElements()==true) {
        stat = (CxStatistics)senum.nextElement();
        String[][] lvp = stat.getLabelValuePairs(Locale.FRANCE);
        System.out.println("-----");
        for (int i=0;i<lvp.length;i++) {
            System.out.println(lvp[i][0]+" = "+lvp[i][1]);
        }
        String trc = stat.getTrc();
        if (trc.equals("")==false) {
            if (trc.equals("0000")==false) {
                System.out.println("TRC explanation = "+srv.getTrcMessage(trc));
            }
        }
        String prc = stat.getPrc();
        if (prc.equals("")==false) {
            if (prc.substring(1).equals("0000")==false) {
                System.out.println("PRC explanation = "+srv.getPrcMessage(prc));
            }
        }
    } //End while
    System.out.println("Returning from displayStatistics");
} //End displayStatistics

static void usage() {
    System.out.println(
        "Display statistics\n"+
        "Usage: java Example2 <ip-address> <port> <number-of-seconds>\n");
    return;
} //End usage

private CxServer srv = null;
} //End class Example2

```

Exemple 3: Configuration statique du serveur.

Ce programme affiche la configuration statique du serveur ainsi que les éléments de la clé logicielle (composantes, dates d'expiration, ...).

Le source Example3.java est fourni dans le fichier CXJAI_examples.jar.

Exemple 4: Journal.

Ce programme affiche le contenu du journal pour les transferts ayant eu lieu pendant les x dernières secondes, x étant passé en 3^{ème} paramètre de lancement.

Le source Example4.java est fourni dans le fichier CXJAI_examples.jar.

Exemple 5: Soumission d'une requête de transfert.

Ce programme soumet une requête de transfert en boucle à un Connect:Express Unix. Son exécution suppose que l'on ait au préalable créé un partenaire BOUCLE et un fichier symbolique FILE01 dans le moniteur.

Le source Example5.java est fourni dans le fichier CXJAI_examples.jar.

Exemple 6: Variables d'environnement d'un serveur Unix.

Ce programme obtient du serveur la valeur de ses variables d'environnement \$TOM_DIR et \$PATH.

Il fait également effectuer par le serveur la substitution de variables d'environnement dans une chaîne.

Le source Example6.java est fourni dans le fichier CXJAI_examples.jar.

Exemple 7: Paramètres de clients ou de serveurs SSL.

Ce programme affiche les paramètres des clients et des serveurs SSL.

Chaque ensemble de paramètres SSL est identifié lors de sa création dans Connect:Express par un identifiant d'au plus 8 caractères.

A la différence des paramètres de clients SSL, les paramètres des serveurs SSL ne peuvent qu'être consultés ,c.a.d qu'ils ne peuvent être ni créés, modifiés ou supprimés à l'aide de l'API. En effet, ces dernières opérations nécessitent un arrêt/redémarrage du moniteur, qui ne peut se faire à distance.

Le source Example7.java est fourni dans le fichier CXJAI_examples.jar.

Exemple 8: Certificats X509.

Ce programme affiche les certificats X509 pouvant être utilisés par Connect:Express pour effectuer l'authentification lors des transferts SSL.

Le source Example8.java est fourni dans le fichier CXJAI_examples.jar.

Configuration serveur Connect:Express

Le paragraphe suivant décrit les éléments de configuration serveur d'API des moniteurs Connect:Express.

Connect:Express Unix.

Ajouter une ligne `APPORTE=<port-d'écoute>` ou `APPORTE=<adresse-ip-locale>:<port-d'écoute>` dans le fichier `$TOM_DIR/config/sysin`.

Par exemple:

```
APPORTE=9000
```

Par ailleurs, la mise en œuvre des statistiques est activée pour le moniteur avec le paramètre

```
ISSTAT=1
```

dans le même fichier `sysin`.

Connect:Express Windows.

Le port d'écoute de l'API est déjà fixé à la valeur 7000 à l'installation du moniteur. Ce port est utilisé par toute application client, notamment l'interface graphique locale.

Pour le modifier, lancer l'interface graphique et ouvrir la boîte de dialogue Paramétrage / Réseaux / TCPIP.

Pour activer les statistiques, utiliser la boîte de dialogue Paramétrage / Fichiers.

Connect:Express OS/390.

Annexe. Données de Connect:Express

Les tableaux ci-dessous indiquent en fonction du système d'exploitation la validité et la taille des différents champs de données des serveurs Connect:Express.

La colonne 'Key' indique le mot clé permettant d'identifier la donnée dans les lignes de statistiques.

Données d'un partenaire symbolique

Key	Field	Lg max	Description	Win	Unix	OS / 390
APPD	AppcDisconnectUsed	1	Disconnect Option for LU6.2 is used, Y or N			1
APPM	AppcModeName	8	Remote LU6.2 mode name	8		8
APPT	AppcTpName	64	Remote LU6.2 transaction program	64		8
FTAC	FtpAccessRight	8	Access rights definition name			8
FTDF	FtpDefaultFile	8	Default file name for FTP		8	8
FTPV	FtpPasvUsed	1	Pasv is active, Y or N			1
LNKL	LinkList	3	If link type = M, list of available links			3
LNAM	LocalName	8	Alias name of the local Connect:Express	8	8	8
LNTP	LocalNameType	1	Dynamic local ID is used, Y or N	1		
LPSW	LocalPassword	8	Alias password of the local Connect:Express	8	8	8
MSES	MaxSession	3	Maximum simultaneous sessions	3	2	3
MSIN	MaxSessionIn	2	Maximum simultaneous sessions Inbound	3	2	3
MSOU	MaxSessionOut	2	Maximum simultaneous sessions Outbound	3	2	3
ODNM	OdetteName	25	Odette identification			25
PCLA	PartnerClass	1	Class to use for inbound transfers			1
PTXT	PartnerComment	80	Description of the symbolic partner definition	80		70
PNAM	PartnerName	8	Symbolic Partner name	8	8	8
PPSW	PartnerPassword	8	Symbolic Partner password	8	8	8
PSTA	PartnerState	1	Symbolic Partner status, E=Enable, H=Disabled	1	1	1
PROT	Protocol	1	Transfer protocol: Windows: D=PeSITD, E=PeSIT, O=Oftp, 3=Etebac3, F=ftp Unix: 0: PeSIT, 1: FTP, 2: Etebac3 (Voir ProtocolVersion dans la table de session pour le niveau de version de PeSIT, D ou E)	1	1	1
RACG	RacfGroup	8	Security racf group			8
RACU	RacfUser	8	Racf user			8
RCSD	RemoteClientSubjectDn	256	Criteria for remote client subject DN control	256		
RCRD	RemoteClientRootDn	256	Criteria for remote client root DN control	256		
RETO	RestartUsed	1	Automatic restart is used, Y or N	1		1
RETN	RetryNumber	2	Maximum number of retries for this partner		2	
RSSD	RemoteServerSubjectDn	256	Criteria for remote server subject DN control	256		
RSRD	RemoteServerRootDn	256	Criteria for remote server root DN control	256		
STAB	SessionTableId	50	Name of the session table or identification number	50	1	1
STMR	SessionTimer	2	Session timer		2	
SLID	SldEntryId	1	SLD entry identification			1
SNAL	SnaLuName	8	Remote SNA address	8		8
TCPA	TcpIpAddress	15	Remote TCP/IP address	15	15	15
TCPH	TcpIpHostName	127	Remote TCP/IP host name	127	32	32
TCPP	TcpIpPort	5	Remote listening TCP/IP port	5	5	5
TTMR	TransferTimer	2	Transfer timer		2	

TYPL	TypeOfLink	1	Type of link, 0 = LU 6.2, 1 = X25, 2 = TCP/IP, M=mixed	1	1	1
TYPP	TypeOfPartner	1	Type of Partner, Other or Tom	1	1	1
X25F	X25Facilities	32	Remote X25 address, facilities	32	16	12
X25L	X25LocalAddress	15	Local X25 address	15	15	15
X25P	X25Localport	2	Local device or MCH identification	2	1	1
X25A	X25RemoteAddress	15	Remote X25 address	15	15	15
X25T	X25Taxation	1	Remote X25 address, Tax rule			1
X25U	X25UserDataField	16	Remote X25 address, user data field	8	8	16
X25G	X25UserGroup	2	X25 Remote X25 address, user Group			2

Données d'un fichier symbolique

Key	Field	Lg max	Description	Win	Unix	OS / 390
CLIN	ClientToNotify	8	Name of the client to notify	8		
DIRB	DirectoryBlock	3	Number of directory blocks (file = P or PU) (MVS dcb)			3
DISP	Disposition	3	Allocation disposition (SHR, NEW, OLD) (MVS dcb)			3
ERCD	EndReceiveCommand	127	User command called at end of reception	127	12	32
EREX	EndReceiveExit	127	User exit called at end of reception	127	12	8
ETCD	EndTransmitCommand	127	User command called at end of transmission	127	12	32
ETEX	EndTransmitExit	127	User exit called at end of transmission	127	12	8
ERRC	ErrorCommand	127	User command called when an error occurs	127		
FLAO	FileAgentUsed	1	Interconnected File Agent is active, Y or N		1	
FBLK	FileBlockSize	5	File Physical block size (MVS dcb)			5
FTXT	FileComment	80	Description of the symbolic file definition	80		79
FDIR	FileDirection	1	Transfer direction authorized, T = transmit, R = receive, * = both	1	1	1
FLAB	FileLabel	80	File user identification	80		
FNAM	FileName	8	Symbolic file name	8	8	8
FOPO	FileOpenOption	1	Allocation rule, N = New file, R = Replace, O = Append	1	1	2
FRCV	FileReceiver	8	Partner, or list of partners, authorized to receive the file	8	8	8
FRFM	FileRecordFormat	3	Local record format (MVS DCB)			3
FRLG	FileRecordLength	5	Local record length	5	5	5
FRET	FileRetention	8	Local expiration or retention date (MVS dcb)			8
FSND	FileSender	8	Partner, or list of partners, authorized to send the file	8	8	8
FSTA	FileState	1	Symbolic file status, E=Enable, H=Disabled	1	1	1
FUNM	FileUnitName	8	Local unit name for allocation (MVS dcb)			8
FTOP	FtpOptions	4	FTP file transfer options (type/structure/mode)		4	4
FTSU	FtpStoreUniqueUsed	1	FTP store Unique is used, Y or N		1	1
GDBG	GdgNumber	3	Gdg file generation number (+xx or -xx)			3
MEMB	JobMember	8	Unload/Reload selection member (file = PU, SU, UU)			8
LPHN	LocalPhysicalName	127	Local file physical name	127	44	44
NFYO	NotifyUsed	1	Notification is used, Y or N	1		
OPHN	OriginPhysicalName	44	File name proposed to remote as their remote data set name (or Pi99)			44
PARM	ParamFileUsed	1	Parameter card file is used, Y or N		1	
RP99	Pi99ValueR	254	Value for Pi99 (reception)	254		
R99O	Pi99OffsetR	3	Offset of the preceding value in the Pi99 (reception)	3		
R99L	Pi99LengthR	3	Length of the preceding value in the Pi99 (reception)	3		
SP99	Pi99ValueT	254	Value for Pi99 (transmission)	254		
S99O	Pi99OffsetT	3	Offset of the preceding value in the Pi99 (transmission)	3		
S99L	Pi99LengthT	3	Length of the preceding value in the Pi99 (transmission)	3		
PTAB	PresentationTableId	50	Name or identification number of the presentation table used	50	1	2
PRIO	Priority	1	Transfer priority, 0 = Urgent, 1 = Normal, 2 = slow		1	1
RPHN	RemotePhysicalName	44	Remote file physical name		44	
SECU	Security	2	Security table identification number			2
SPAO	SpaceAllocationUsed	1	Space reservation Y/N		1	
SPA1	SpacePrimary	4	Allocation primary space (MVS dcb)			4

SPA2	SpaceSecondary	4	Allocation secondary space (MVS dcb)			4
SPAT	SpaceType	3	Allocation space type (CYL, TRK, ...) (MVS dcb)			3
SRCD	StartReceiveCommand	127	User command called at beginning of reception	127	12	32
SREX	StartReceiveExit	127	User exit called at beginning of reception	127	12	8
STCD	StartTransmitCommand	127	User command called at beginning of transmission	127	12	32
STEX	StartTransmitExit	127	User exit called at beginning of transmission	127	12	8
TYPA	TypeOfAllocation	1	Type of allocation, F = Fixed, D=Dynamic	1	1	1
TYPF	TypeOfFile	2	Type of file, TF = Text fixed, TV = text variable, XF = Unix text fixed on Windows, XV = Unix text variable on Windows, UF = Unix fixed, UV = Unix Variable, BF = binary fixed, BU = Binary undefined, S = Sequential, V = VSAM, P = PDS, PE = PDSE, PU = PDS unload, VU = VSAM unload, SU = SYSOUT unload, UU = User unload	2	2	2
TYPN	TypeOfNotification	1	Type of Notification: 1 character ('0' to '7'). '0': No notification. '1': Notification at the beginning of the transfer. '2': Notification at the end of the transfer. '4': Notification if transfer error. Other possibilities are combinations with inclusive « OR » of these values. For example: '6' = '2' OR '4' for a notification at the end of transfer or in case of transfer error. Windows: This flag is used for HTTP notification only. Unix: This flag is used for HTTP notification or standard notification depending on the values of the keywords HTTPNF and NOTIFY in the sysin configuration file.	1	1	
VOLN	VolumeName	30	List of 1 to 5 (6 characters) Volume(s) name(s)			30

Données d'une table de session

Key	Field	Lg max	Description	Win	Unix	OS / 390
BMSG	BaseMessageSize	5	Network message size (negociation)	4	5	5
BSNC	BaseSynchronizationSize	5	Synchronization Kbytes size (negociation)	2	2	5
BWIN	BaseWindowSize	2	Synchronization window size (negociation)		2	2
CMPO	CompressionUsed	1	Odette Compression used, Y or N			1
CRCO	CrcUsed	1	CRC used , Y or N	1	1	1
PVER	ProtocolVersion	2	Protocol version		1	2
RSYN	ResynchronizationNumber	2	Number of resynchronization for the request	2		
RSYO	ResynchronizationUsed	1	Resynchronization is used, Y or N	1		1
RETN	RetryNumber	2	Number of retries for the request		2	
SDIR	SessionDirection	1	Transfer direction authorized, T = transmit, R = receive, * = both	1		1
STAB	SessionTableId	50	Name of the session table or identification number	50	1	1
SMSG	SnaMessageSize	5	Specific SNA message size (negociation)			5
SSNC	SnaSynchronizationSize	5	Specific SNA Synchronization Kbytes size (negociation)			5
SWIN	SnaWindowSize	2	Specific SNA Synchronization window size (negociation)			2
TMSG	TcpMessageSize	5	Specific TCP/IP message size (negociation)			5
TSNC	TcpSynchronizationSize	5	Specific TCP/IP Synchronization Kbytes size (negociation)			5
TWIN	TcpWindowSize	2	Specific TCP/IP Synchronization window size (negociation)			2
XMSG	X25MessageSize	5	Specific X25 message size (negociation)			5
XSNC	X25SynchronizationSize	5	Specific X25 Synchronization Kbytes size (negociation)			5
XWIN	X25WindowSize	2	Specific X25 Synchronization window size (negociation)			2

Données d'une table de présentation

Key	Field	Lg max	Description	Win	Unix	Mvs
CMPO	CompressionUsed	1	Compression used, Y or N			1
CONC	ConcatenationUsed	1	PeSIT Fpdu Data option	1		
HPFO	HighPerformanceUsed	1	Bulk transfer flag			3
IBUF	InternalBuffer	3	Size of internal buffer			3
IOEX	IoUserExit	8	Name of the I/O user exit			8
MULT	MultiArticleUsed	1	Multiarticle is used in PeSIT Fpdu Data, Y or N	1	1	
PTAB	PresentationTableId	50	Name or identification number of the presentation table used	50	1	3
SEGM	SegmentationUsed	1	Segmentation is used in PeSIT Fpdu Data, Y or N	1		1
TREA	TranslationToAscii	127	EbcDic to Ascii Translation table identification number or name	127		
TRAE	TranslationToEbcDic	127	Ascii to EbcDic translation table identification number or name	127	1	
TRAO	TranslationUsed	1	Translation is used, Y or N	1		
TYPC	TypeOfCompression	2	Compression, Horizontal,Vertical,Mixed or pres.table identification number	1	1	1
TYPD	TypeOfData	1	Type of data, A = Ascii, E = EbcDic, B = Binary			1
UEX1	UserExitOne	8	Name of the first user exit			8
UEX3	UserExitThree	8	Name of the third user exit			8
UEX2	UserExitTwo	8	Name of the second user exit			8

Paramètres de soumission de requête

Key	Field	Lg max	Description	Win	Unix	Mvs
AHGP	AdHocGroup	8	AdHoc remote user racf group			8
AHN2	AdHocNewConfirm	8	AdHoc remote user New Password confirmation			8
AHN1	AdHocNewPassword	8	AdHoc remote user New Password			8
AHPW	AdHocPassword	8	AdHoc remote user Password	8		8
AHUS	AdHocUser	8	AdHoc remote user ID	8		8
CLIN	ClientToNotify	256	Name of the client to notify	256		
DATE	DateOfExecution	18	Date Time when the request must be scheduled	18	18	18
FLAO	FileAgentUsed	1	Interconnected File Agent is active, Y or N		1	
FAPI	FileApi	88	User description of the transfer – Etebac3 card – Odette transfer	80	88	82
FLAB	FileLabel	80	File user identification	80	80	
FNAM	FileName	8	Symbolic file name	8	8	8
FRFM	FileRecordFormat	3	AdHocLocal record format (MVS DCB)			3
FRLG	FileRecordLength	5	Local record length		5	5
FTOP	FtpOptions	4	FTP file transfer options (type/structure/mode)		3	4
FTSU	FtpStoreUniqueUsed	1	FTP store Unique is used, Y or N		1	1
MEMB	JobMember	8	Unload/Reload selection member (file = PU, SU, UU)			8
JOBN	JobName	8	Job name of the sysout to transfer (SYSOUT)			8
LSP1	LoaclSpacePrimary	4	AdHoc local allocation primary space type (MVS dcb)			4
LBLK	LocalBlockSize	5	AdHoc local physical block size (MVS dcb)			5
LDIR	LocalDirectoryBlock	3	AdHoc local number of directory blocks (file = P or PU)			3
LDS1	LocalDisposition1	3	AdHoc local allocation disposition (SHR, NEW, OLD)			3
LDS2	LocalDisposition2	3	AdHoc local allocation disposition (KEEP,CTLG)			3
LDS3	LocalDisposition3	3	AdHoc local Allocation disposition (KEEP,CTLG)			3
LNAM	LocalName	8	Alias name of the local Connect:Express	8	8	8
LPSW	LocalPassword	8	Alias password of the local Connect:Express	8	8	8
LPHN	LocalPhysicalName	127	Local file physical name	127	44	44

LRET	LocalRetentionDate	8	AdHoc local expiration or retention date (MVS dcb)			8
LSP2	LocalSpaceSecondary	4	AdHoc local allocation secondary space type (MVS dcb)			4
LSPT	LocalSpaceType	3	AdHoc local allocation space type (CYL, TRK, ...) (MVS dcb)			3
LTAP	LocalTapeDefinition	7	AdHoc local sequence number (4) and Tape label (3)			7
LUNT	LocalUnitName	8	AdHoc local unit name for allocation (MVS dcb)			8
LVOL	LocalVolumeName	30	AdHoc list of 1 to 5 (6 characters) local volume(s) name(s)			30
MEML	MemberList	256	List of 1 to 32 (8 characters) members (SELECTION)			256
NFYO	NotifyUsed	1	Notification is used, Y or N	1		
OPHN	OriginPhysicalName	44	File name proposed to remote as their remote data set name (or Pi99)	44		44
PNAM	PartnerName	8	Symbolic Partner name	8	8	8
PI99	Pi99Value	254	Pi99 to send	254	254	
P99O	Pi99Offset	3	Offset in the Pi99		3	
P99L	Pi99Length	3	Length in the Pi99		3	
PRIO	Priority	1	Transfer priority, 0 = Urgent, 1 = Normal, 2 = slow	1	1	1
RACG	RacGroup	8	Security racf group			8
RSP2	RemoteSpaceSecondary	4	AdHoc remote allocation secondary space (MVS dcb)			4
RBLK	RemoteBlockSize	5	AdHoc remote physical block size (MVS dcb)			5
RDIR	RemoteDirectoryBlock	2	AdHoc remote number of directory blocks (file = P or PU) (MVS dcb)			2
RDS3	RemoteDisposition3	3	AdHoc remote allocation disposition (KEEP,CTLG) (MVS dcb)			3
RDS1	RemoteDisposition1	3	AdHoc remote allocation disposition (SHR, NEW, OLD) (MVS dcb)			3
RDS2	RemoteDisposition2	3	AdHoc remote allocation disposition (KEEP,CTLG) (MVS dcb)			3
RPHN	RemotePhysicalName	44	Adhoc remote file physical name	44	44	44
RRFM	RemoteRecordFormat	3	AdHoc remote Record format (MVS dcb)			3
RREC	RemoteRecordLength	5	AdHoc remote record length (MVS dcb)			5
RRET	RemoteRetentionDate	8	AdHoc remote expiration or retention date (MVS dcb)			8
RSP1	RemoteSpacePrimary	4	AdHoc remote allocation primary space (MVS dcb)			4
RSPT	RemoteSpaceType	3	AdHoc allocation space type (CYL, TRK, ...) (MVS dcb)			3
RTAP	RemoteTapeDefinition	7	AdHoc remote sequence number (4) and tape label (3)			7
RUNT	RemoteUnitName	8	AdHoc remote unit name for remote allocation (MVS dcb)			8
RVOL	RemoteVolumeName	40	AdHoc list of 1 to 5 (6 characters) remote volume name(s) (MVS dcb)			40
RCLA	RequestClass	1	APM class where to execute the request			1
RMOD	RequestMode	1	Request scheduling mode, I = Immediat, D = Differed			1
REQR	Requestor	8	The name of the entity (user, job ..) that submitted the request	8		
SYSN	SysoutNumber	8	Sysout identification number to transfer (SYSOUT transfer)			8
PRMF	SysprmFile	44	Unload/Reload selection member directory (SELECTION)			44
TDST	TransferDestination	8	The entity that is processing the transfer request	8	8	8
TDIR	TransferDirection	1	Transmission or Reception	1	1	1
TORG	TransferOrigin	8	The entity that is requesting the transfer	8	8	8
TRCV	TransferReceiver	24	The entity that is processing the file after receiving it	8	24	
TSND	TransferSender	24	The entity that is processing the file before sending it	8	24	
TYPF	TypeOfFile	2	Type of file, TF = Text fixed, TV = text variable, UF = Unix fixed, UV = Unix Variable, BF = binary fixed, BU = Binary undefined, S = Sequential, V = VSAM, P = PDS, PE = PDSE, PU = PDS unload, VU = VSAM unload, SU = SYSOUT unload, UU = User unload		2	
TYPL	TypeOfLink	1	Type of link, 0 = LU 6.2, 1 = X25, 2 = TCP/IP, M=mixed	1	1	1
TYPN	TypeOfNotification	1	Type of Notification: 1 character ('0' to '7'). '0': No notification. '1': Notification at the beginning of the transfer. '2': Notification at the end of the transfer. '4': Notification if transfer error. Other possibilities are combinations with inclusive « OR » of these values. For example: '6' = '2' OR '4' for a notification at the end of transfer or in case of transfer error. Windows: This flag is used for HTTP notification only. Unix: This flag is used for HTTP notification or standard notification depending on the values of the keywords HTTPNF and NOTIFY in the sysin configuration file.	1	1	
TYPR	TypeOfRequest	1	Type of request, N = Normal, I = Inquiry, H = Hold	1	1	1
REQU	UserRequestId	16	Identification of the request given by the user	16		

Données du journal

Key	Field	Lg max	Description	Win	Unix	OS / 390
APPM	AppcModeName	8	Remote LU6.2 mode name	8		8
APPT	AppcTpName	64	Remote LU6.2 transaction program	64		8
BDEB	BitDebit	8	Number of bits per second			8
CDEB	CharacterDebit	8	Number of characters per second			8
CLIN	ClientToNotify	8	Name of the client to notify			x
CRCO	CrcOption	1	CRC used , Y or N	1	1	
ERC	CtreeReturnCode	4	Return code from ctree access	4		
DATE	DateOfExecution	18	Date when the request is accepted by Connect:Express	18	18	18
REQX	ExternalRequestNumber	8	Request number on the remote side	8	8	8
FLAO	FileAgentUsed	1	Interconnected File Agent is active, Y or N		1	
FAPI	FileApi	88	User description of the transfer – Etebac3 card – Odette transfer		88	82
FBYT	FileBytes	12	Number of bytes of the file	12		8
FLAB	FileLabel	80	File user identification	80	80	80
FNAM	FileName	8	Symbolic file name	8	8	8
FNRD	FileNumberOfRecords	12	Number of records sent/received	12	12	8
FOPO	FileOpenOption	1	Allocation rule, N = New file, R = Replace, O = Append	1	1	1
FORG	FileOrganization	1	File organization, S = Sequential, I = Indexed, R = Relative	1	1	1
FRLG	FileRecordLength	5	Local record length	5	5	5
FSIZ	FileSize	8	Size announced by the sender			8
FTOP	FtpOptions	3	FTP file transfer options (type/structure/mode)		3	
FTSU	FtpStoreUniqueUsed	1	FTP store Unique is used, Y or N		1	
JNDA	JournalRecordDate	18	Date when the journal record is written by Connect:Express	18		18
LNAM	LocalName	8	Alias name of the local Connect:Express	8	8	8
LPHN	LocalPhysicalName	127	Local file physical name	127	44	44
MRET	MaxRetries	2	Maximum number of retries for this partner		2	
MULT	MultiArticleUsed	1	Multiarticle is used in PeSIT Fpdu Data, Y or N		1	1
NBYT	NetworkBytes	12	Number of bytes transferred	12	12	8
NMGS	NetworkMessageSize	5	Network message size	4	5	5
NRC	Nrc	4	Network Return code	4	4	6
OPHN	OriginPhysicalName	44	File name proposed to remote as their remote data set name (or P199)		44	
PNAM	PartnerName	8	Symbolic Partner name	8	8	8
PPSW	PartnerPassword	8	Symbolic Partner password		8	
PRC	Prc	3	Protocol return code	4	4	4
PTAB	PresentationTableId	2	Identification number of the presentation table used			2
PRI0	Priority	1	Transfer priority, 0 = Urgent, 1 = Normal , 2 = slow	1	1	1
PRID	ProcessId	12	Identification of the process that executed the request (PID, APMEFF ..)		12	5
PROT	Protocol	1	Transfer protocol, D=PesitD, E=PesitE, O=Ofpt, 3=Etebac3, F=ftp	1	1	1
REQP	PurgedByMonitor	1	Request is purged, Y or N	1		
COMP	RealCompression	4	Compression performed	2		4
RPHN	RemotePhysicalName	44	Remote file physical name	44	44	
RCLA	RequestClass	1	APM class where the request has been executed			1
RELA	RequestElapse	8	Transfer elapse			8
REQN	RequestNumber	12	Request number given by Connect:Express	12	8	8
REQR	Requestor	8	The name of the entity (user, job ..) that submitted the request	8	8	8
RSYN	ResynchronizationNumber	3	Number of resynchronization for the request	3		3
RETN	RetryNumber	2	Number of retries for the request	2	2	2
SRCT	SendReceiveCount	12	Number of network send receive	12		8
SERV	ServiceType	1	Type of transfer request, AdHoc or Normal			1
STAB	SessionTableId	2	Identification number of the session table used			2
STMR	SessionTimer	2	Session timer		2	
SNAL	SnaLuName	8	Remote SNA address	8		8
SNA1	SnaRc1	4	Primary SNA return code	4		

SNA2	SnaRc2	8	Secondary SNA return code	8		
SPAO	SpaceAllocationUsed	1	Space reservation Y/N		1	
SRC	Src	8	System Return code	8	8	4
SRC2	Src2	4	Complementary System Return code			
RSTA	Status	1	Transfer status, E = Ended / S = Started / I = Interrupted / W = Waiting	1	1	1
SYNC	SynchronizationInterval	2	Size of synchronization interval Kbytes	2		2
WIND	SynchronizationWindow	2	Number of synchronization intervals	2		2
TCPA	TcpIpAddress	15	Remote TCP/IP address	15	15	15
TCPH	TcpIpHostName	127	Remote TCP/IP host name	127	32	15
TCPP	TcpIpPort	5	Remote listening TCP/IP port	5	5	5
TCPC	TcpIpRc	4	TCP/IP return code	4	4	
TBDA	TransferBeginningDate	18	Beginning of transfer date and time	18	18	18
TDST	TransferDestination	8	The entity that is processing the transfer request	8	8	8
TDIR	TransferDirection	1	Transmission or Reception	1	1	1
TEDA	TransferEndDate	18	End of transfer date and time	18	18	18
TIDT	TransferIdent	8	Transfer ident exchanged with the partner	6	6	8
TORG	TransferOrigin	8	The entity that is requesting the transfer	8	8	8
TRCV	TransferReceiver	24	The entity that is processing the file after receiving it	8	24	24
TSND	TransferSender	24	The entity that is processing the file before sending it	8	24	24
TTMR	TransferTimer	2	Transfer timer		2	
TRTN	TranslationTableNumber	1	Translation table identification number		1	2
TRC	Trc	4	Connect:Express Return code	4	4	4
TYPA	TypeOfAllocation	1	Type of allocation, F = Fixed, D=Dynamic	1		
TYPC	TypeOfCompression	1	Compression, Horizontal,Vertical,Mixed or presentation table identification	1	1	1
TYPD	TypeOfData	1	Type of data, A = Ascii, E = Ebcdic, B = Binary	1	1	1
TYPF	TypeOfFile	2	Type of file, TF = Text fixed, TV = text variable, UF = Unix fixed, UV = Unix Variable, BF = binary fixed, BU = Binary undefined, S = Sequential, V = VSAM, P = PDS, PE = PDSE, PU = PDS unload, VU = VSAM unload, SU = SYSOUT unload, UU = User unload	2	2	2
TYPL	TypeOfLink	1	Type of link, 0 = LU 6.2, 1 = X25, 2 = TCP/IP, M=mixed	1	1	1
TYPN	TypeOfNotification	1	Type of Notification: 1 character ('0' to '7'). '0': No notification. '1': Notification at the beginning of the transfer. '2': Notification at the end of the transfer. '4': Notification if transfer error. Other possibilities are combinations with inclusive « OR » of these values. For example: '6' = '2' OR '4' for a notification at the end of transfer or in case of transfer error. Windows: This flag is used for HTTP notification only. Unix: This flag is used for HTTP notification or standard notification depending on the values of the keywords HTTPNF and NOTIFY in the sysin configuration file.	1	1	
TYPP	TypeOfPartner	1	Type of Partner, Other or Tom	1	1	1
TYPR	TypeOfRequest	1	Type of request, N = Normal, I = Inquiry, H = Hold	1	1	1
TYPU	TypeOfUser	1	Type of user, I = Internal, E = External	1	1	1
USDR	UserDataReceived	254	User information received with the file	254	254	44
USDS	UserDataSent	254	User information sent with the file	254	254	44
REQU	UserRequestID	16	Identification of the request given by the user	16		
X25C	X25Cause	2	X25 Cause	2	2	2
X25D	X25Diagnostic	2	X25 Diagnostic	2	2	
X25F	X25Facilities	32	Remote X25 address, facilities	32	32	12
X25L	X25LocalAddress	15	Local X25 address	15	15	15
X25P	X25Localport	1	Local device or MCH identification	2	1	1
X25R	X25Rc	4	X25 Return code	4	4	
X25A	X25RemoteAddress	15	Remote X25 address	15	15	15
X25U	X25UserDataField	16	Remote X25 address, user data field	8	8	16
XLOV	XLocalPhysicalNameOvf	1	Set to true if overflow while setting XlocalPhysicalName value		8	
XLPH	XLocalPhysicalName	16	Equivalent to LocalPhysicalName with environment variables replaced.		512	

Données d'un transfert en cours

Key	Field	Lg max	Description	Win	Unix	OS / 390
APPM	AppcModeName	8	Remote LU6.2 mode name			8
APPT	AppcTpName	64	Remote LU6.2 transaction program			8
BDEB	BitDebit	12	Number of bits per second			12
CDEB	CharacterDebit	12	Number of characters per second			12
CLIN	ClientToNotify	8	Name of the client to notify			
CRCO	CRCOption	1	CRC used , Y or N			1
ERC	CtreeReturnCode	4	Return code from ctree access	4		
REQX	ExternalRequestNumber	8	Request number on the remote side			8
FLAO	FileAgentUsed	1	Interconnected File Agent is active, Y or N			
FAPI	FileApi	82	User description of the transfer – Etebac3 card – Odette transfer			82
FBYT	FileBytes	12	Number of bytes of the file	12		12
FLAB	FileLabel	80	File user identification			80
FNAM	FileName	8	Symbolic file name	8	8	8
FNRD	FileNumberOfRecords	12	Number of records sent/received			12
FOPO	FileOpenOption	1	Allocation rule, N = New file, R = Replace, O = Append			1
FORG	FileOrganization	1	File organization, S = Sequential, I = Indexed, R = Relative			1
FRLG	FileRecordLength	5	Local record length			5
FSIZ	FileSize	12	Size announced by the sender			12
JNDA	JournalRecordDate	18	Date when the journal record is written by Connect:Express			18
LNAM	LocalName	8	Alias name of the local Connect:Express			8
LPHN	LocalPhysicalName	127	Local file physical name	127	44	44
MULT	MultiArticleUsed	1	Multiarticle is used in PeSIT Fpdu Data, Y or N			1
NBYT	NetworkBytes	12	Number of bytes transferred	12	12	12
NMGS	NetworkMessageSize	4	Network message size			4
NRC	Nrc	4	Network Return code	4	4	6
PNAM	PartnerName	8	Symbolic Partner name	8	8	8
PRC	Prc	4	Protocol return code	4	4	4
PTAB	PresentationTableId	2	Identification number of the presentation table used			2
PRIO	Priority	1	Transfer priority, 0 = Urgent, 1 = Normal , 2 = slow			1
PRID	ProcessID	12	Identification of the process that executed the request (PID, APMEFF ..)			12
PROT	Protocol	1	Transfer protocol, D=PesitD, E=PesitE, O=Ofpt, 3=Etebac3, F=ftp			1
REQP	PurgedByMonitor	1	Request is purged, Y or N			
COMP	RealCompression	4	Compression performed			
RCLA	RequestClass	1	APM class where the request has been executed			1
REQD	RequestDate	18	Date when the request is accepted by Connect:Express			18
RELA	RequestElapse	8	Transfer elapse			8
REQN	RequestNumber	12	Request number given by Connect:Express	12	8	8
REQR	Requestor	8	The name of the entity (user, job ..) that submitted the request	8	8	8
RSYN	ResynchronizationNumber	2	Number of resynchronization for the request			2
RETN	RetryNumber	2	Number of retries for the request			2
SRCT	SendReceiveCount	12	Number of network send receive			12
SERV	ServiceType	1	Type of transfer request, AdHoc or Normal			1
STAB	SessionTableId	2	Identification number of the session table used			2
SNAL	SnaLuName	8	Remote SNA address			8
SNA1	SnaRc1	4	Primary SNA return code			
SNA2	SnaRc2	8	Secondary SNA return code			
SRC	Src	8	System Return code	8	8	4
SRC2	Src2	4	Complementary System Return code			
RSTA	Status	1	Transfer status, E = Ended / S = Started / I = Interrupted / W = Waiting	1	1	1
SYNC	SynchronizationInterval	4	Size of synchronization interval Kbytes			4
WIND	SynchronizationWindow	4	Number of synchronization intervals			4
TCPA	TcpIpAddress	15	Remote TCP/IP address			15
TCPH	TcpIpHostName	127	Remote TCP/IP host name			32
TCPP	TcpIpPort	5	Remote listening TCP/IP port			5
TCPC	TcpipPrc	4	TCP/IP return code			

TBDA	TransferBeginningDate	18	Beginning of transfer date and time	18	18	18
TDST	TransferDestination	8	The entity that is processing the transfer request		8	8
TDIR	TransferDirection	1	Transmission or Reception	1	1	1
TEDA	TransferEndDate	18	End of transfer date and time			18
TIDT	TransferIdent	8	Transfer ident exchanged with the partner			8
TORG	TransferOrigin	8	The entity that is requesting the transfer		8	8
TRCV	TransferReceiver	24	The entity that is processing the file after receiving it			24
TSND	TransferSender	24	The entity that is processing the file before sending it			24
TRTN	TranslationTableNumber	1	Translation table identification number			2
TRC	Trc	4	Connect:Express Return code	4	4	4
TYPA	TypeOfAllocation	1	Type of allocation, F = Fixed, D=Dynamic			1
TYPC	TypeOfCompression	2	Compression, Horizontal,Vertical,Mixed or pres.table identification number			1
TYPD	TypeOfData	1	Type of data, A = Ascii, E = Ebcdic, B = Binary			1
TYPF	TypeOfFile	2	Type of file, TF = Text fixed, TV = text variable, UF = Unix fixed, UV = Unix Variable, BF = binary fixed, BU = Binary undefined, S = Sequential, V = VSAM, P = PDS, PE = PDSE, PU = PDS unload, VU = VSAM unload, SU = SYSOUT unload, UU = User unload			2
TYPL	TypeOfLink	1	Type of link, 0 = LU 6.2, 1 = X25, 2 = TCP/IP, M=mixed			1
TYPP	TypeOfPartner	1	Type of Partner, Other or Tom	1	1	1
TYPR	TypeOfRequest	1	Type of request, N = Normal, I = Inquiry, H = Hold			1
TYPU	TypeOfUser	1	Type of user, I = Internal, E = External	1	1	1
USDR	UserDataReceived	254	User information received with the file		254	254
USDS	UserDataSent	254	User information sent with the file		254	254
REQU	UserRequestID	16	Identification of the request given by the user	16		8
X25C	X25Cause	2	X25 Cause			
X25D	X25Diagnostic	2	X25 Diagnostic			
X25F	X25Facilities	32	Remote X25 address, facilities			12
X25L	X25LocalAddress	15	Local X25 address			15
X25P	X25Localport	1	Local device or MCH identification			1
X25R	X25rc	4	X25 Return code			
X25A	X25RemoteAddress	15	Remote X25 address			15
X25U	X25UserDataField	16	Remote X25 address, user data field			16

Données de configuration du moniteur

Key	Field	Lg max	Description	Win	Unix	OS / 390
AHSO	AdHocSecurityUsed	1	AdHoc Racf option: Yes or Unsafe			1
ALIAS	Alias name	80	Alias name from asset protection key	80	80	
APIA	ApiAddress	15	API listening address	x		15
APIP	ApiPort	5	API listening port	X	5	5
APIV	ApiVersion	3	API version Number	X		
AUXX	APLinexx	100	Asset protection line numer xx (n lines)			
APPV	AppcVersion	3	Appc api version number	3		
AUTF	AuthFile	44	AUTHDS data set name			44
BLDO	BuidDataBaseUsed	1	Automatic build Data base during termination		1	
CLDN	ClientDefaultToNotify	8	Name of the default client for notifications	8		
CLTM	ClientTimer	4	Timer for Client sessions	4		
CPUI	CpuId	20	JES2 Interface (ISF, SAM, SYSV)			20
CSVV	CsvVersion	3	Csv api version number	3		
DNOT	DefaultNotificationUsed	1	default client to notify is defined, Y or N	1		
SPSW	Dpcpsw	8	Partner password of the Connect:Express server		8	8
DPCI	Dpcsid	8	Partner identification of the Connect:Express server		8	8

FILF	FilesFile	44	Path and file name for the Files directory			44
FILT	FilesTotalDefinitions	5	Number of symbolic files defined			5
FTDF	FtpDefaultFile	8	Default file name for FTP		8	8
FTAL	FtpListAllUsed	1	\$ALL\$ files are included into the FTP list, Y or N		1	1
FTLA	FtpListenAddress	15	FTP listening address		15	15
FTLP	FtpListenPort	5	FTP listening port		5	5
FTPN	FtpTransferNumber	3	Maximum number of FTP file transfers			3
IJNO	InitJournalUsed	1	Automatic journal initialization	1		
ILOG	InitLogFileUsed	1	Automatic logfile initialization	1		
ISTA	StatisticsUsed	1	Statistics active, Y or N		1	
JES2	Jes2Interface	4	JES2 Interface (ISF, SAM, SYSV)			4
JNLS	JournalSize	5	Number of records of the journal	5		5
LOAD	LoadLib	44	Loadlib data set name			44
LODB	LogDebugUsed	1	Syslog debugging activation		1	
LOGS	LogSize	5	Number of records of the log	5	4	5
MSGU	MessageUser	80	Message sent to operator at initialization			80
MSTA	MonitorStatus	1	Monitor status, Active or Inactive			1
NPPN	NamedPipeName	127	Name of the named pipe	127		
NPPO	NamePipeUsed	1	Named pipe resource status	1		
NRES	Network ressources	10	T/S/X/L... for TCP/IP, SNA, X25, APPC			10
NOTI	NotificationsUsed	1	Notifications active, Y or N			
NOTS	NotificationSize	5	Number of records of the notification file	5		5
X25C	NumberCVC	2	Number of X25 virtual circuit			2
ODPT	OdetteListenPort	5	OFTP (Odette) listening port			5
PARF	PartnersFile	44	Path and file name for the partners directory			44
PART	PartnersTotalDefinitions	5	Number of symbolic partners defined			5
PROD	ProductInfo	80	Connect:Express information (version)	x	14	80
RACF	RacfProfile	1	Transfers are under security control, Y or N			1
RQTS	RequestTableSize	5	Request table size			5
RETN	RetryNumber	4	Number of retries for the request	4		4
RUTY	RunType	1	Run type, Hot or Cold	1	1	1
SVCO	ServiceUsed	1	Service configuration	1		1
SMSO	SmsUsed	1	Sms is used, Y or N			1
SDL1	SnaAppcWindowsDll	15	Name of the APPC dll	15		
SNAA	SnaApplication	8	SNA application prefix or SNA LUName	8		6
SNAB	SnaControlUsed	1	SNA Incoming call control is bypassed, Y or N	1		
SDL2	SnaCsvWindowsDll	15	Name of the CSV dll	15		
SNAO	SnaUsed	1	SNA resource is active, Y or N	1		
SSNM	Subsystem name	4	Name of connect:Express subsystem interface			4
STEV	SessionTimer	4	Session timer in minutes		2	
EVTF	SysevtFile	44	SYSEVT data set name			44
SYCF	SysinConfigurationFile	44	SYSIN data set name			44
JCLF	SysjclFile	44	SYSJCL data set name			44
JNLF	SysjnlFile	44	SYSJNL data set name			44
LOGF	SyslogFile	44	SYSLOG data set name			44
PRMF	SysprmFile	44	SYSPRM data set name			44
SYSF	SyssnaFile	44	SYSSNA data set name			44
SYTF	SystcpFile	44	SYSTCP data set name			44
SYXF	Sysx25File	44	SYSX25 data set name			44
SINF	SytemInfo	255	Operating system information (version)	x	255	80
TCPB	TcpipBypassUsed	1	TCP/IP Incoming call control is bypassed, Y or N	1		
TDLL	TcpipDll	15	Name of the windows socket	15		
TCPO	TcpipUsed	1	TCP/IP resource is active, Y or N	1		
TPLA	TcpListenAddress	15	General listening address	15	15	
TPPT	TcpListenPort	5	General listening port	5	5	5
TWUN	TemporaryWorkUnit	6	Work Unit used for temporary files			6
TRAC	TraceUsed	1	Trace is active, Y or N	1	1	
TRFM	TransferMax	3	Maximum number of simultaneous file transfers (outbound for Unix)		4	
TTEV	TransferTimer	2	Timer for file transfer sessions in minutes		2	

TTMR	TransferTimer	4	Timer for file transfer sessions	4		
X25B	X25BypassUsed	1	X25 Incoming call control option	1		
X25N	X25LineNumber	2	Number of X25 lines			2
X25L	X25LocalAddress	15	Local X25 address ou DLL Windows	15		15
X25P	X25Localport	2	Local X25 address, port number	2		
X25O	X25Used	1	X25 resource is active, Y or N	1		
XDLL	X25WindowsDll	15	Name of the Eicon X25 dll	15		

Paramètres de clients SSL

Les paramètres de clients SSL peuvent être créés, mis à jour ou supprimés.

Key	Field	Lg max	Description	Win	Unix	OS / 390
SNAM	ParameterName	8	Name of this parameter definition	8	8	
SSTA	Status	1	Status, E=Enabled, H=Disabled	1	1	
STOL	StoreLocation	64	Store location SYSTEM_STORE_LOCAL_MACHINE, SYSTEM_STORE_CURRENT_USER or SYSTEM_STORE_SERVICES	64		
STNM	StoreName	128	Store name (My)	128		
SUBJ	Subject	256	Certificate subject distinguished name	256		
ISSU	Issuer	256	Certificate issuer distinguished name	256		
PROT	Protocol	1	Protocol – '1':TLSV1, '3':SSLV3, '2':SSLV2	1		
CISU	CipherSuites	45	List of cipher suites numbers 00: TLS_RSA_WITH_RC4_128_MD5 01: TLS_RSA_WITH_RC4_128_SHA 02: TLS_RSA_WITH_3DES_EDE_CBC_SHA 03: TLS_RSA_WITH_DES_CBC_SHA 04: TLS_RSA_WITH_NULL_MD5 05: TLS_RSA_WITH_NULL_SHA 06: SSL CK_RC4_128_WITH_MD5 07: SSL CK_DES_64_CBC_WITH_MD5 08: SSL CK_RC2_128_CBC_WITH_MD5	45		
TRLV	TraceLevel	1	'0':None, '1':Medium, '2': Full	1		
SBHD	SslBytesHeaderUsed	1	A 2 bytes length header is inserted before each PeSIT data sent '1': Yes, '0': No	1	1	
VOPT	VerificationOption	1	Authentication '0':VERIFY_NONE, '1': VERIFY_PEER		1	
CERT	CertificateId	8	Connect:Express Identifier of a certificate associated to this definition. This identifier references a name chosen when importing the certificate into the Connect:Express certificates database		8	
CILI	CipherList	16	Name of a file located in \$TOM_DIR/config/ssl/ciphlist and containing a list of ciphers that can be used		16	
TLS1	Tlsv1	1	'1':TLSV1 available, '0':TLSV1 not available		1	
SSL3	Sslv3	1	'1':SSLV3 available, '0':SSLV3 not available		1	
SSL2	Sslv2	1	'1':SSLV2 available, '0':SSLV2 not available		1	

Paramètres de serveurs SSL

Les paramètres de serveurs SSL ne peuvent être créés, mis à jour ou supprimés que sur le système du serveur, par \$stern (Unix) ou l'interface graphique en mode configuration (Windows).

Key	Field	Lg max	Description	Win	Unix	OS / 390
SNAM	ParameterName	8	Name of this parameter definition	8	8	
SSTA	Status	1	Status, E=Enable, H=Disabled	1	1	
STOL	StoreLocation	64	Store location	64		
STNM	StoreName	128	Store name	128		
SUBJ	Subject	256	Certificate subject distinguished name	256		
ISSU	Issuer	256	Certificate issuer distinguished name	256		
PROT	Protocol	1	Protocol – '1':TLSV1, '3':SSLV3, '2':SSLV2	1		
CISU	CipherSuites	45	List of cipher suites numbers 00: TLS_RSA_WITH_RC4_128_MD5 01: TLS_RSA_WITH_RC4_128_SHA 02: TLS_RSA_WITH_3DES_EDE_CBC_SHA 03: TLS_RSA_WITH_DES_CBC_SHA 04: TLS_RSA_WITH_NULL_MD5 05: TLS_RSA_WITH_NULL_SHA 06: SSL CK_RC4_128_WITH_MD5 07: SSL CK_DES_64_CBC_WITH_MD5 08: SSL CK_RC2_128_CBC_WITH_MD5	45		
TRLV	TraceLevel	1	'0':None, '1':Medium, '2': Full	1		
SBHD	SslBytesHeaderUsed	1	A 2 bytes length header is inserted before each PeSIT data sent '1': Yes, '0': No	1	1	
VOPT	VerificationOption	1	Authentication '0':VERIFY_NONE, '1':VERIFY_PEER, '2':VERIFY_FAIL_IF_NO_PEER_CERT		1	
CERT	CertificateId	8	Connect:Express Identifier of a certificate associated to this definition. This identifier references a name chosen when importing the certificate into the Connect:Express certificates database		8	
CILI	CipherList	16	Name of a file located in \$TOM_DIR/config/ssl/ciphlist and containing a list of ciphers that can be used		16	
CLIA	ClientAuthenticationUsed	1	'0': None, '1': Server requests clients to authenticate	1		
TLS1	Tlsv1	1	'1':TLSV1 available, '0':TLSV1 not available		1	
SSL3	Sslv3	1	'1':SSLV3 available, '0':SSLV3 not available		1	
SSL2	Sslv2	1	'1':SSLV2 available, '0':SSLV2 not available		1	
TCPP	TcpipPort	5	SSL server listen port number	5	5	
TCPA	TcpipAddress	15	Server IP address		15	
CALI	CaList	8	Certification authority certificate identifier (as imported into Connect:Express) or name of a file containing a list of such identifiers (#LIST).		8	
DHPF	DHParamFile	16	Name of a file containing Diffie-Hellman parameters		16	