Connect:Express

Batch Administration Utility

Version 1.3.6

Sterling Commerce
An IBM Company

Connect:Express
Batch Administration Utility

Version 1.3.6
First edition

Contents

CONTENTS **2**

Preface 1
Introduction 2

CHAPTER 1 .... **1**

Overview 1
Installation *1*
Activating the API Server *3*
Passing Parameters Instream to Cxcmd *4*
Help *6*
Using Text Files for Input and Ouput *7*
Using Results as Input Script File *10*
Return Codes of Cxcmd *11*
Connecting to a Remote Server *11*
Specifying the Format of Information *13*
Short and Long Parameter Names *14*
Wildcards *14*

CHAPTER 2 .... **17**

Reference Guide 17
Activity *18*
APKey *20*
Configuration *21*
Etb3Presentation *22*
File *24*
Journal *28*
NodeVersion *31*
Partner *33*
Presentation *37*
Server *40*
Session *42*
Statistics *44*
Transfer *46*
Version *51*
SslClientParameter *52*
SslServerParameter *56*
Certificate *58*

APPENDIX **63**

Symbolic Partner Data 63
Symbolic File Data 64
Session Table Data 65
Presentation Table Data 66
Request Submission Parameters 66
Journal Data 68
Active Transfer Data 70
Monitor’s Configuration Data 72

SSL CLIENT PARAMETERS 74
SSL SERVER PARAMETERS 75

Preface

This document describes how to use the batch java administration utility of Connect:Express. This utility enables you to manage Connect:Express monitors that are running on Unix (from version 143-109) and Windows (from version 303.002), both locally and remotely.

Chapter 1 describes the installation procedure and provides an overview of the utility.

Chapter 2 provides a reference guide for the parameters of the utility.

The batch administration utility of Connect:Express requires JRE version 1.4.2 or higher.

Introduction

The administration utility of Connect:Express (cxcmd) enables you to administrate both Connect:Express Unix and Windows monitors, locally and remotely.
In the following, Connect:Express is called « server » or « monitor », and the utility is called « client ».

The administration utility enables you to interact with the various components of the monitor, as listed below :

- View, create, update and delete symbolic partners and files.
- View and update session and presentation tables.
- View general configuration parameters of the monitor.
- View active transfers, journal records and statistics records.
- Submit transfer request.
- Interrupt, restart and purge transfer requests.

The client establishes a TCP/IP session with the monitor, locally or remotely. The utility can execute on both Unix and Windows platforms, indifferently.

Chapter 2 provides detailed information about the utility functionality.

Chapter 1

This chapter provides an overview about installation and use of the administration utility.

Overview

If you need to administrate one or several Unix or Windows servers, you have to install the utility on either a Unix or a Windows client. You must activate an API server that listens on an IP port, one for each monitor that you want to administrate. On the client side, session parameters can be defined in a property file, or specified in the parameters of the utility.

The utility enables you to manage all components of the monitor (partner, file …). A help is available online. Various modes are provided for use (create, replace, …), and each mode requires several parameters. You can pass these parameters to the utility in stream or prepare them in an input script file. The input script file can be the output file returned by the utility during a previous execution.

Long names and short names are provided for parameters. You can specify the format of the output.

Installation

You can install cxcmd utility on a Unix or a Windows platform. First, install a Java Runtime Environment on the client platform. The JRE version must be 1.4.2 or higher.

Use the following command to check the version of the current JRE:

```
java -version
```

Specify the pathname of the java program in the PATH variable of cxcmd utility environment.

The administration utility is delivered as a zip file CXCMD1.3.6.zip (Windows) or a tar file CXCMD1.3.6.tar (Unix).

Installing the Client on a Unix Platform

Perform the following operations :

1. Copy CXCMD1.3.6.tar file in the directory of the user account where you plan to execute the utility, for example : /u2/CXCMD.
2. From a terminal, extract the archive as shown in the following box :

```
tar xvf CXCMD1.3.6.tar
```

3. Extracted files are listed below :

   - cxcmd (shell to execute the utility)
   - cxcmd.bat
   - cxcmd.class
   - cxcmd.jar
   - CXJAI.jar

4. Edit the cxcmd execution shell script :

```
# !/bin/ksh
#========================================================================================
# 1. Replace /opt/jdk1.5.0_06/jre with the path of your java runtime directory
# 2. Replace /u2/CXCMD with the path of your cxcmd installation directory
#========================================================================================
export CLASSPATH=/u2/CXCMD:/u2/CXCMD/cxcmd.jar:/u2/CXCMD/CXJAI.jar:/opt/jdk1.5.0_06/jre:
java -DCXCMD_HOME=/u2/CXCMD cxcmd $*
if [ "$?" != "0" ]; then
  echo === ERROR ===
fi
```

5. Replace /opt/jdk1.5.0_06/jre by the current path name to the JRE of your system.
6. Replace /u2/CXCMD by the path name of the files that you extracted.
7. Change cxcmd permissions:

```
chmod u+x cxcmd
```

8. If needed, add the pathname of cxcmd script to your environment variable $PATH.

Intalling the Client on a Windows Platform

Perform the following operations :

1. Extract CXCMD1.3.6.zip in a folder where you want to install the utility, for example c:\u2\CXCMD.

2. The following files are extracted :

   - cxcmd
   - cxcmd.bat (.bat file for executing the utility)
   - cxcmd.class
   - cxcmd.jar
   - CXJAI.jar

3. Edit the execution file cxcmd.bat :

```
echo OFF
REM ========================================================
REM 1. Replace c:\Program Files\java\jre1.5.0_06 with the path
REM    of your java runtime directory
REM 2. Replace c:\u2\CXCMD with the path of your cxcmd installation
REM ========================================================
set CLASSPATH=c:\u2\CXCMD;c:\u2\CXCMD\cxcmd.jar;c:\u2\CXCMD\CXJAI.jar;c:\Program Files\java\jre1.5.0_06
java -DCXCMD_HOME=c:\u2\CXCMD cxcmd %*
if errorlevel 1 echo === ERROR =====
```

4. Replace c:\Program Files\java\jre1.5.0_06 by the current path of the JRE of your system.
5. Replace c:\u2\CXCMD by the path name of the files that you extracted.
6. If needed, add the pathname of cxcmd script to your environment variable PATH. Open the dialog box « Control Panel - System - Advanced – Environment Variables ».

Activating the API Server

The way you can activate the API server depends on the platform on which the monitor is executing.

Activating the Server on Windows

To accept cxcmd utility requests Connect:Express Windows must be licensed for « Activity Manager » option.

Connect:Express Windows listens on a special port to accept calls from remote client requests. The default value of this port is 7000. You can view the actual value of the API port through the Connect:Express GUI , in the dialog box « Administration \ Parameters \ Monitor \ Network \ TCP/IP ».

Activating the Server on Unix

To accept cxcmd utility requests Connect:Express Unix must be licensed for « Activity Manager » option.

To accept cxcmd utility requests, Connect:Express Unix must be specified an APPORT parameter in its sysin file, that is placed in $TOM_DIR/config directory.

Edit the sysin file and insert the following line :

```
APPORT=<port-number>
```

You must stop / start the monitor for the update to be effective. When the process tom_api is active, Connect:Express is ready to accept client sessions.

Passing Parameters Instream to Cxcmd

The syntax of the command for launching the utility is shown below :

```
cxcmd <component> parameter1=<value1>,…, parameterN=<valueN>,mode=<mode>
```

Parameters can be passed in any order. « Component » specifies the type of object to which the command applies. « Mode » indicates the type of operation to perform.

The table below lists the component that are available :

Component	**Description**
Server	Use this component to define session parameters on the client platform. Network address and logging information for establishing connection to the remote API server are defined in this object.
Partner	Symbolic partners table of one monitor.
File	Symbolic files table of one monitor.
Session	Session tables - PeSIT and Etebac3 – of one monitor.
Presentation	Presentation tables - PeSIT and Etebac3 – of one monitor on Unix. Presentation tables - PeSIT – of one monitor on Windows.
Etb3Presentation	Presentation table – Etebac3 – of one monitor on Windows.
Transfer	Transfer requests table of one monitor.
Journal	Journal file of one monitor.
Activity	Active transfers table of one monitor.
Configuration	Configuration file of one monitor.
APKey	License information of one monitor (Asset Protection Key).
Statistics	Statistics file of one monitor.
NodeVersion	Version number of one Connect:Express monitor.
Version	Version number of the cxcmd utility and the Connect :Express JAI.

The table below lists the modes that are available :

Mode	**Description**
CREATE	A new entry is added to the component. An error is issued if the entry already exists.
REPLACE	A new entry is added to the component if it does not already exist. The entry is replaced if it already exists.
UPDATE	The entry of the component is updated from the fields provided.
GET	View one entry or a list a entries of the component. Parameters of the entry are returned to the user output.
DELETE	Delete one entry from the component.
LIST	Display a list of entries of the component. The list is returned to the user output
CONNECT	Establish session with a remote server.
DISCONNECT	Close session with a remote server.
SUBMIT	Submit a transfer request to the monitor.
INTERRUPT	Interrupt a transfer request.
RESTART	Restart a transfer request.
PURGE	Purge one or several transfer requests.

A mode may not apply to all components. For example mode=REPLACE is not valid with component=Journal.

You can pass commands from a Unix command line or a Windows command line.

For example, the following command submits a transfer request to a monitor :

```
$ cxcmd transfer ServerId=ADMIN@LIN01,FileName=FILE01,TransferDirection=T,\
> PartnerName=BOUCLE,LocalName=BOUCLE,LocalPassword=PSW,\
> PhysicalName=$TOM_DIR/config/sysin.txt,  Mode=submit
```

Continuation Character.

Some commands, such as CREATE or REPLACE, can require a large number of parameters. You can use input scripts as described in « *Using Text Files for Input* » , but you can also use the command line and you will probably need several lines and use the continuation character.

On Unix the continuation character is « \ », on Windows the continuation character is « ^ ».

Case.

Parameters are not case sensitive. For example, « RequestNumber » is equivalent to « requestnumber ».
Values of parameters are generally changed to uppercase, except if the parameter is case sensitive for the monitor to which the command is sent. For example, « TransfertDirection=t » is equivalent to « TransferDirection=T », but in « LocalPhysicalName=/User1/Files/file01.txt » the value is unchanged.

Blanks in parameters.

When the value of a parameter includes blanks, the parameter must be specified between brackets (example « parameter='value   with a lot of   blanks' »).

Help

You can display the help using the following syntax :

```
cxcmd help
or
cxcmd help <component>
```

Using Text Files for Input and Ouput

The utility supports text files for input and output, called script files and result files, in which you can group several operations to perform.

Use the syntax shown below :

```
cxcmd in=<script-file-name>[,out=<result-file-name>]
```

The format of an input script file is :

```
[control directive or comment]
<component> parameter1=<value1>,
             …,
             parameterN=<valueN>,
             mode=<mode>

…
[control directive or comment]
<component> parameter1=<value1>,
             …,
             parameterN=<valueN>,
             mode=<mode>

…
```

A comment either starts with « # » character, or is placed between « /* » and « */ ». For example :

```
…
# Comment
/* Comment
*/
…
```

Control directives are listed below :

onerror continue : specifies that, if an error is detected during execution of a command, next commands are executed. This is the default.
onerror return : specifies that, if an error is detected during execution of a command, next commands are ignored.
return : The execution of the script is interrupted unconditionally. Next commands are ignored.
comment <*text*> : displays « text » to the standard output.

The following script, script1.txt, displays the entries of the partners and the files of a monitor :

script1.txt

```
onerror continue
/*------------------------------*/
/* SYMBOLIC PARTNERS            */
/*------------------------------*/
Comment =========== PARTNERS OF LIN01 =============
partner
     PartnerName        = *,
     Serverid      = ADMIN@LIN01,
     Mode               = get
/*------------------------------*/
/* SYMBOLIC FILES               */
/*------------------------------*/
Comment =========== FILES OF LIN01 =============
file
     FileName      = *,
     Serverid      = ADMIN@LIN01,
     Mode               = get
return
...
```

The following is displayed :

```
comment ===========PARTNERSOFLINE01=================
_Correct Connected to server ADMIN@LIN01
_Correct partner PartnerName=*,Serverid=ADMIN@LIN01,Mode=get
_Correct Count = 12
Partner
     PartnerName             = BCLDSA,
     ServerId                = ADMIN@LIN01,
     PartnerPassword         = PSW,
     LocalName               = BCLDSA,
     LocalPassword           = PSW,
     PartnerState            = E,
     TypeOfPartner           = O,
     Protocol                = 3,
     MaxSession              = 20,
     MaxSessionIn            = 10,
     MaxSessionOut           = 10,
     SessionTableId          = 1,
     TypeOfLink              = T,
     TcpipAddress            = 127.0.0.1,
     TcpipPort               = 6681,
     TcpipHostName           = ,
     X25LocalAddress         = ,
     X25RemoteAddress        = ,
     X25LocalPort            = ,
     X25UserDataField        = ,
     X25Facilities           = ,
     FtpDefaultFile          = ,
     RetryNumber             = 0,
     SessionTimer            = 0,
     TransferTimer           = 0,
     SslparmId               = CLIDSA,
     Mode                    = replace
...
comment ===========FILESOFLIN01=================
_Correct file FileName=*,ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 4
File
     FileName                = ETB01,
     ServerId                = ADMIN@LIN01,
     FileState               = E,
     TypeOfAllocation        = D,
     FileDirection           = *,
     TypeOfFile              = TF,
     FileOpenOption          = R,
     FileSender              = $$ALL$$,
     FileReceiver            = $$ALL$$,
     PresentationTableId     = 1,
     LocalPhysicalName       = $TOM_DIR/in/&REQNUMB.tmp,
     FileRecordLength        = 80,
     StartTransmitExit       = ,
     EndTransmitExit         = ,
     StartReceiveExit        = ,
     EndReceiveExit          = ,
     StartTransmitCommand    = ,
     EndTransmitCommand      = ,
     StartReceiveCommand     = ,
     EndReceiveCommand       = ,
     Priority                = 1,
     RemotePhysicalName      = ,
     FtpOptions              = ***,
     ParamFileUsed           = false,
     SpaceAllocationUsed     = false,
     FtpStoreUniqueUsed      = false,
     FileAgentUsed           = false,
     TypeOfNotification      = 0,
     Mode                    = replace
...
```

The reporting of the execution of the command begins with « _Correct » or « _Error », then the results of the command are displayed to the user-out.

Using Results as Input Script File

You can use the result out of certain command or script as input script for another command.
Use the following syntax, with an output file to receive results:

```
cxcmd in=<script-file-name>,out=<result-file-name>
```

For example, the following command receives the script1.txt file shown before as input :

```
cxcmd in=script1.txt,out=script2.txt
```

The reporting of the execution is displayed on the user-out :

```
_Correct Connected to server ADMIN@LIN01
_Correct partner PartnerName=*,Serverid=ADMIN@LIN01,Mode=get
_Correct Count = 12
_Correct file FileName=*,ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 4
```

The result displayed in script2.txt is shown in the following :

script2.txt

```
comment ===========PARTENAIRESDELIN01=================
Partner
     PartnerName             = BCLDSA,
     ServerId                = ADMIN@LIN01,
     PartnerPassword         = PSW,
     LocalName               = BCLDSA,
       ...
     SslparmId               = CLIDSA,
     Mode                    = replace
...
comment ===========FICHIERSDELIN01=================
File
     FileName                = ETB01,
     ServerId                = ADMIN@LIN01,
     FileState               = E,
     TypeOfAllocation        = D,
       ...
     TypeOfNotification      = 0,
     Mode                    = replace
...
```

script2.txt contains all the partners and the files entries of the monitor : you can use it as input script file to reload the partners and files in the monitor tables.

Return Codes of Cxcmd

Cxcmd returns the following status : « 0 » if there is no error, « 1 » in case of error. If the script contains the directive « onerror continue », status is « 1 » if at least one error has been detected during execution.

You can test the status in a Unix shell script as shown below :

```
cxcmd ...
if [ "$?" != "0" ] ; then
  echo ====== ERROR ====
  ...
 exit 1
fi
...
```

You can test the status in a .bat Windows file as shown below :

```
cxcmd ...
if errorlevel 1 goto error
...
:error
echo ====== ERROR ====
...
```

Connecting to a Remote Server

Each command must be specified a « ServerId » parameter that defines session information used to logon to the server.

Session information are listed below :

- TCP/IP Address or host name of the server
- Server listening port
- Operating system of the server (X or W)
- User name
- User password
- Trace option (optionnal)
- Maximum number of statistics record expected (optionnal)
- Session Timeout (optionnal)
- Local session port range (optionnal)

You can specify session information in the parameters of the command or in a local properties text file, call cxcmd_servers.txt, that you place in the directory where the utility is installed.

Passing session information on the command line:

```
<component> ServerId=
"<addr/host>:port:<OS>:<user>:<password>[:<FALSE|TRUE>][:<limit>][:<timeout>][:<
port-range>]"
```

(OS= X : Unix, W : Windows)

For example :

```
<component>
      ServerId="10.87.15.42:9000:X:ADMIN:ADMIN"
```

Defining a server in a local property file cxcmd_servers.txt :

Use the component « server » to create, update, view or delete the local definitions of the servers, as shown in the following :

```
server
     ServerId      = ADMIN@WIN01,
     Address       = 10.87.15.92,
     Port          = 7000,
     OSType        = WINDOWS,
     UserName      = ADMIN,
     UserPassword  = ADMIN,
     Mode               = replace
```

You can use « `ServerId=ADMIN@WIN01` » in the command line to specify session information.

For example :

```
<component>
      ServerId=ADMIN@WIN01
```

Note : you can test the link to a server as shown below :

```
cxcmd server ServerId="10.87.15.42:9000:X:ADMIN:ADMIN",mode=connect
```

Specifying the Format of Information

By default, the information is returned by the utility as shown below :

```
<component>
      Parameter            =  value,
      …
```

Use parameter « Ofmt » to specify different formats.

Ofmt = XML

XML format is available for all components with GET and LIST modes.

For example :

```
cxcmd journal RequestNumber=*,ServerId=ADMIN@LIN01,Ofmt=xml,Mode=get
```

displays the journal information as the following :

```
<Journal>
<Transfer>
     <RequestNumber>07500001</RequestNumber>
     <TransferIdent>4B0001</TransferIdent>
      ...
     <MaxRetries>0</MaxRetries>
     <Mode>get</Mode>
</Transfer>
...
<Transfer>
     <RequestNumber>07800016</RequestNumber>
     <TransferIdent>4E000F</TransferIdent>
      ...
     <MaxRetries>0</MaxRetries>
     <Mode>get</Mode>
</Transfer>
</Journal>
```

Ofmt = COL:HDR

This format is available for the list of journal record or the list of active transfers. The information is displayed in a table (COL) with (HDR) or without a header.

Example :

```
cxcmd journal RequestNumber=*,ServerId=ADMIN@LIN01,Ofmt=COL:HDR,Mode=list
```

The result is shown below :

```
-----------------------------------------------------------------------------
Request    Requestor  FileName  PartnerName  Dir  Status Date
-----------------------------------------------------------------------------
07500001   REMAPI     FILE01    BOUCLE       T    E      2006/09/15 10:07:47
07800016   pga        FILE01    BOUCLE       R    E      2006/09/18 14:45:27
...
```

Ofmt = RAW

This option is available for statistics. Statistics must be active in the configuration of the monitor.

Short and Long Parameter Names

You can use short names for parameters input, instead of long names, but long names are only used for output. Short names correspond to long names and are 5 characters long maximum.

Chapter 2 provides the corresponding long and short names in the description of operations.

In the following, the first example uses long names and the second example uses the corresponding short names:

```
$ cxcmd transfer ServerId=ADMIN@LIN01,FileName=FILE01,TransferDirection=T,\
> PartnerName=BOUCLE,LocalName=BOUCLE,LocalPassword=PSW,\
> PhysicalName=$TOM_DIR/config/sysin.txt,Mode=submit
```

This is equivalent to :

```
$ cxcmd transfer srvid=ADMIN@LIN01,fnam=FILE01,tdir=T,\
> pnam=BOUCLE,lnam=BOUCLE,lpsw=PSW,\
> phn=$TOM_DIR/config/sysin.txt,mode=submit
```

Wildcards

For modes GET, LIST and PURGE, some of the components support « * » and « ? » wildcards.

« * » : any number of characters, including no character.
« ? »: any character at the current position.

Example :
RequestNumber=071*, corresponds to the list of all transfer requests for day 071 on Connect:Express Unix.
PartnerName=P?A, corresponds to all partners which name is 3 characters long, with « P » in the first position, any character in the second position, and « A » in the third position.

Chapter 2

This chapter provides detailed description of the command for each component.

Reference Guide

The following provides the format of the command, the parameters and examples, for each component.

Data types are specified in the tables as follows :

- S : Alphanumeric string
- s : Alphanumeric string, case sensitive
- C : Character
- i : Integer
- L : Long
- B : Boolean (true or false)
- D : Date YYYY/MM/DDHH:MM:SS

Default parameter values for modes CREATE and REPLACE are : *blank* for types S and s, *blank* for type c, *zero* for types i and L, and *false* for type boolean. If the default value is different from the default indicated above, it is specified.

Columns **Unix** and **Win** specify if a parameter is defined for the platform, and the maximum length supported for this parameter.

Activity

Use « activity » to display information of an active transfer or a list of transfers. Active transfers have the following status :

Unix :

- A : Waiting selection
- C : Executing
- D : Differed
- J : Restarting
- K : Waiting restart

Windows :

- W : Waiting selection
- T : Selected
- C : Executing
- I : Interrupted
- S : Selection error
- R : Restarting

Format

```
activity
      ServerId=id,
      RequestNumber=(request-number|wildcard),
      TypeOfUser=(E|I|*),
      Requestor=(requestor-name|*),
      UserRequestId=(id|*),
      FileName=(name|*),
      TransferDirection=(T|R|*),
      PartnerName=(name|*),
      Ofmt=(std|xml|col|col:hdr),
      Mode=(get|list)
```

Parameters ServerId and Mode are required.

Modes Get and List - Parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
RequestNumber	REQN	Request number of the transfer – wild card supported	12	8	S
TypeOfUser	TYPU	Type of user (E :External, I : Internal, * : both)	1		C
Requestor	REQR	Name of the transfer requestor (Username, process name, …), or ‘*’	8	8	S
UserRequestId	REQU	User identification of the transfer request , or ‘*’	16		s
FileName	FNAM	Symbolique file name , or ‘*’	8	8	S

TransferDirection	TDIR	Transfer direction (T : Transmission, R : Reception, * : Both)	1	1	C
PartnerName	PNAM	Symbolic partner name , or ‘*’	8	8	S
Ofmt	OFMT	Format of the result returned by cxcmd Modes get and list : std (Standard), xml Mode list : col (column), hdr (header)	64	64	S
Mode	MODE	get or list	4	4	S

Example1

```
activity
      ServerId          = ADMIN@WIN01,
      Mode              = get
```

Display :

```
_Correct activity ServerId=ADMIN@LIN01,Mode=get
_Correct Connected to server ADMIN@LIN01
_Correct activity ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 2
Activity
Transfer
      RequestNumber           = 08800181,
      Requestor               = pga,
      FileName                = FILE02,
      TransferDirection       = T,
      PartnerName             = BOUCLE,
      PhysicalName            = $TOM_DIR/config/sysin.txt,
      Status                  = C,
      StatusMessage           = In progress,
      TypeOfUser              = I,
      Trc                     = 0000,
      Prc                     = 0000,
      Src                     = 0000,
      Nrc                     = 0000,
      NetworkBytes            = 0,
      TransferBeginningDate   = "2006/09/28 15:03:37",
      TransferOrigin          = BOUCLE,
      TransferDestination     = BOUCLE,
      TypeOfPartner           = O,
      UserDataReceived        = ,
      UserDataSent            = ,
      Mode                    = get
...
Transfer
      RequestNumber           = 08800184,
        ...
      Mode                    = get
```

Example2

```
activity
      ServerId          = ADMIN@WIN01,
      TransferDirection       = T,
      Ofmt              = col:hdr,
      Mode              = list
```

Display :

```
_Correct activity ServerId=ADMIN@LIN01,TransferDirection=T,Ofmt=COL:HDR,Mode=list
_Correct Connected to server ADMIN@LIN01
_Correct Count = 2
-------------------------------------------------------------------------------
Request       Requestor  FileName  PartnerName  Dir  Status
-------------------------------------------------------------------------------
08800181      pga        FILE02    BOUCLE       T    C
08800184      pga        FILE02    BOUCLE       T    C
```

APKey

Use « apkey » to display information from the asset protection key of the monitor. With this command you can check the expiration date of the key and the options included in the license.

Format

```
apkey
      ServerId=id,
      Ofmt=(std|xml),
      Mode=get
```

Parameters ServerId and Mode are required.

Parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard) or xml	64	64	S
Mode	MODE	get	3	3	S

Example1

```
apkey
      ServerId          = ADMIN@LIN01,
      Mode              = get
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct apkey ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 1
Apkey
      ServerId                = ADMIN@LIN01,
      ExpirationDate          = NO-LIMIT,
      ApkeyLines
      ApkeyLine               = "PRODUCT EXPRESS",
      ApkeyLine               = "PRODUCT NUMBER 082",
      ApkeyLine               = "EXPIRATION DATE NO-LIMIT",
      ApkeyLine               = "OPERATING SYSTEM UNIX",
      ApkeyLine               = "ALIAS $SSL-LABS X001",
      ApkeyLine               = "PESIT NO-LIMIT",
      ApkeyLine               = "TCPIP NO-LIMIT",
      ApkeyLine               = "FTP NO-LIMIT",
      ApkeyLine               = "ETEBAC3 NO-LIMIT",
      Mode                    = get
```

Configuration

Use « configuration » to display the configuration information of a monitor.

Format

```
configuration
      ServerId=id,
      Ofmt=(std|xml),
      Mode=get
```

Parameters ServerId and Mode are required.

Parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard) or xml	64	64	S
Mode	MODE	get	3	3	S

Example1

```
configuration
      ServerId          = ADMIN@LIN01,
      Mode              = get
```

Display :

```
Configuration
      ServerId                = ADMIN@LIN01,
      OSType                  = UNIX,
      GMTOffset               = 7200,
      AliasName               = UNIX,
      ApiPort                 = 9000,
      ApiAddress              = ,
      BuildDatabaseUsed       = false,
      Dpcsid                  = FEDORA,
      FtpListAllUsed          = true,
      FtpDefaultFile          = FTPFILE,
      FtpListenAddress        = ,
      FtpListenPort           = 0,
      StatisticsUsed          = false,
      SyslogUsed              = false,
      LogSize                 = 10000,
      NotificationsUsed       = false,
      RunType                 = C,
      Dpcpsw                  = PSW,
      SessionTimer            = 1,
      TcpipListenAddress      = ,
      TcpipListenPort         = 6677,
      TraceUsed               = false,
      MaxTransfer             = 6,
      TransferTimer           = 1,
      X25LocalPort            = ,
      X25LocalAddress         = ,
      ProductInfo             = "C:E/UNIX 142-1",
      SystemInfo              = "Sysname:Linux.Release:2.6.17-1.2157_FC5.Version:#1 Tue Jul 11
       22:55:46 EDT 2006.Machine:i686.Nodename:localhost.localdomain",
      Mode                    = get
```

Etb3Presentation

Use « Etb3Presentation » to manage Etebac3 presentation tables of Connect:Express Windows. The access key is the name of the table « PresentationTableId ».

Note : for Connect :Express Unix, use « Presentation » to manage PeSIT and Etebac3 presentation tables .

Format

```
etb3presentation
      PresentationTableId=id,
      ServerId=id,
      TranslationUsed=(true|false),
      TranslationToEbcdic=name,
      TranslationToAscii=name,
      Ofmt=(std|xml),
      Mode=(create|replace|update|delete|get|list)
```

Modes Create, Replace and Update - Parameters

Long name	Short name	Description	Win	Unix	Data Type
PresentationTableId	PTAB	Name of the Etebac3 presentation table	50		s
ServerId	SRVID	Identification of the server	128		S
TranslationUsed	TRAO	Transcodage option (true/false)	5		
TranslationToEbcdic	TRAE	Name or identification number of the ASCII - EBCDIC translation table	127		
TranslationToAscii	TREA	Name or identification number of the EBCDIC - ASCII translation table	127		
Mode	MODE	create, replace or update	7		S

Parameters PresentationTableId ,ServerId and Mode are required.

Defaults :
With modes **create** and **replace**, the following non-standard defaults apply, if the corresponding parameter is omitted : TypeOfCompression=N, TranslationToEbcdic=TOMNT.ASC, TranslationToAscii=TOMNT.ASC.

Modes Delete - Parameters

Long name	Short name	Description	Win	Unix	Data Type
PresentationTableId	PTAB	Name of the Etebac3 presentation table	50		s
ServerId	SRVID	Identification of the server	128		S
Mode	MODE	delete	6		S

Parameters PresentationTableId , ServerId and Mode are required.

Modes Get and List - parameters

Long name	Short name	Description	Win	Unix	Data Type
PresentationTableId	PTAB	Name of the Etebac3 presentation table or wildcard	50		s
ServerId	SRVID	Identification of the server	128		S
Ofmt	OFMT	Format of the information by cxcmd - std (standard) or xml	64		S
Mode	MODE	get or list	4		S

Parameters PresentationTableId , ServerId and Mode are required.

Example1

```
etb3Presentation
      PresentationTableId     = "ETB3PRESB",
      ServerId                = ADMIN@WIN01,
      TranslationUsed         = true,
      TranslationToEbcdic     = TOM.ASC,
      TranslationToAscii      = TOM.EBC,
      Mode                    = replace
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct Etb3Presentation PresentationTableId="ETB3PRESB",ServerId=ADMIN@WIN01,
TranslationUsed=true,TranslationToEbcdic=TOMNT.ASC,TranslationToAscii=TOMNT.EBC,Mode=replace
_Correct etb3Presentation ETB3PRESB replaced
```

Example2

```
etb3Presentation
      PresentationTableId     = *,
      Mode                    = list
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct Etb3Presentation PresentationTableId=*,ServerId=ADMIN@WIN01,Mode=list
_Correct Count = 3
Etb3PresentationList
      PresentationTableId     = "ETEBAC3 presentation with translation",
      PresentationTableId     = "ETEBAC3 presentation without translation",
      PresentationTableId     = ETB3PRESB
```

File

Use « file » to manage the symbolic files of a monitor. The access key to the symbolic files table is the symbolic file name, « FileName ».

Format

```
file
      FileName=name,
      ServerId=id,
      FileComment=text,
      FileState=(E|H),
      TypeOfAllocation=(D|F),
      FileDirection=(T|R|*),
      TypeOfFile=(TF|TV|BF|BU),
      FileOpenOption=(N|R|O),
      FileSender=name,
      FileReceiver=name,
      PresentationTableId=id,
      LocalPhysicalName=file-name,
      FileRecordLength=(0|length),
      StartTransmitExit=file-name,
      EndTransmitExit=file-name,
      StartReceiveExit=file-name,
      EndReceiveExit=file-name,
      StartTransmitCommand=file-name,
      EndTransmitCommand=file-name,
      StartReceiveCommand= file-name,
      EndReceiveCommand=file-name,
      ErrorCommand=file-name,
      NotifyUsed=(true|false),
      ClientToNotify=name,
      Pi99OffsetT=offset,
      Pi99LengthT=length,
      Pi99ValueT=string,
      Pi99OffsetR=offset,
      Pi99LengthR=length,
      Pi99ValueR=string,
      FileLabel=string,
      Priority=(0|1|2),
      RemotePhysicalName=file-name,
      FtpOptions=type-structure-mode,
      ParamFileUsed=(true|false),
      SpaceAllocationUsed=(true|false),
      FtpStoreUniqueUsed=(true|false),
      FileAgentUsed=(true|false),
      TypeOfNotification=(type),
      Ofmt=(std|xml),
      Mode=(create|replace|update|delete|get|list)
```

Modes Create, replace and Update - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
FileName	FNAM	Symbolic name of the file Characters supported : A-Z,0-9,ESPACE	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
FileComment	FTXT	Description of the symbolic file	80		s
FileState	FSTA	Status of the symbolic file (E : Enabled, H : Hold)	1	1	C
TypeOfAllocation	TYPA	Type of allocation (F : Fixed, D :Dynamic)	1	1	C
FileDirection	FDIR	Transfer direction supported (t : Transmission, R : Reception, * : Both)	1	1	C
TypeOfFile	TYPF	Type of the file (TF : Text fixed, TV : Text variable, BF : Binary fixed, BU : Binary undefined), XF : Unix text fixed on Windows, XV : Unix text variable on Windows Unix: T*, B*, ** can be used	2	2	S
FileOpenOption	FOPO	File open option (N : New, R : Replace, O : Append)	1	1	C
FileSender	FSND	Partner or list of partners allowed to send the file Character range : A-Z,0-9,Blank,$,* (Windows) A-Z,0-9,Blank,$,* and # (Unix)	8	8	S
FileReceiver	FRCV	Partner or list of partners allowed to receive the file Characters range: A-Z,0-9,Blank,$,* (Windows) A-Z,0-9,Blank,$,* and # (Unix)	8	8	S
PresentationTableId	PTAB	Name or identification number of the presentation table	50	1	S
LocalPhysicalName	LPHN	Local physical name	127	44	s
FileRecordLength	FRLG	Record length	5	5	i

StartTransmitExit	STEX	Transmission, beginning of transfer user exit	127	12	s(*)
EndTransmitExit	ETEX	Transmission, end of transfer user exit	127	12	s(*)
StartReceiveExit	SREX	Reception, beginning of transfer user exit	127	12	s(*)
EndReceiveExit	EREX	Reception, end of transfer user exit	127	12	s(*)
StartTransmitCommand	STCD	Transmission, beginning of transfer user command	127	12	s(*)
EndTransmitCommand	ETCD	Transmission, end of transfer user command	127	12	s(*)
StartReceiveCommand	ERCD	Reception, beginning of transfer user command	127	12	s(*)
EndReceiveCommand	ERCD	Reception, end of transfer user command	127	12	s(*)
ErrorCommand	ERRC	Transfer error user command	127		s
NotifyUsed	NFYO	Notification option (true/false)	5		B
ClientToNotify	CLIN	Name of the client to notify	8		S
Pi99OffsetT	S99O	Offset of Pi99ValueT for pi99 to transmit	3		i
Pi99LengthT	S99L	Length of Pi99ValueT for pi99 to transmit	3		i
Pi99ValueT	SP99	Value of Pi99 to transmit	254		s
Pi99OffsetR	R99O	Offset of Pi99ValueT for pi99 to receive	3		i
Pi99LengthR	R99L	Length of Pi99ValueT for pi99 to receive	3		i
Pi99ValueR	RP99	Value of Pi99 to receive	254		s
FileLabel	FLAB	File label (user identification)	80		s
Priority	PRIO	Priority (0 : Urgent, 1 : Normal, 2 : Low)		1	C
RemotePhysicalName	RPHN	Remote physical name		44	S
FtpOptions	FTOP	FTP Options (Type/Structure/Mode) Type :(E :Ebcdic, A :Ascii, B : Binary, * : any) Structure (F : File, R: Record, *: any) Mode (B: Bloc, S: Stream, *: any)		3	S
ParamFileUsed	PARM	Parameter card file option (true/false)		5	B
SpaceAllocationUsed	SPAO	Space allocation option (true/false)		5	B
FtpStoreUniqueUsed	FTSU	Ftp « store unique » option (true/false)		5	B
FileAgentUsed	FLAO	File agent option (true/false)		5	B
TypeOfNotification	TYPN	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	C
Mode	MODE	create, replace or update	7	7	S

For modes **create** and **replace**, the following parameters are required:

Unix : FileName, ServerId, Mode, FileDirection, PresentationTableId, TypeOfFile, FileSender, FileReceiver, LocalPhysicalName.

Windows : FileName, ServerId, Mode, FileName, FileDirection, PresentationTableId.
if FileDirection=R or *, FileSender.
if FileDirection=T or *, FileReceiver and FileRecordLength.
if NotifyUsed=true, ClientToNotify.

For mode **update** the following parameters are required: FileName, ServerId and Mode.

Defaults :
For modes **create** and **replace,** the following non-standard defaults apply, if the corresponding parameter is omitted : FileState = E, TypeOfAllocation=D, FileOpenOption=R (Unix,Windows). FtpOptions=”***” (Unix).

(*) For user exits and commands fields the characters range are : A-Z, 0-9, a-z, ESPACE, $, ., /, & , _

Modes Delete - parameters

Long name	Short name	Description	Win	Unix	Data Type
FileName	FNAM	Symbolic file name	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
Mode	MODE	delete	4	4	S

Parameters FileName,ServerId and Mode are required.

Modes Get and List - parameters

Long name	Short name	Description	Win	Unix	Data Type
FileName	FNAM	Symbolic file name or wildcard	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard)	64	64	S
Mode	MODE	get or list	4	4	S

Parameters FileName,ServerId and Mode are required.

Example1

```
file
      FileName                = FILE01,
      ServerId                = ADMIN@LIN01,
      FileState               = E,
      TypeOfAllocation        = D,
      FileDirection           = *,
      TypeOfFile              = TV,
      FileOpenOption          = R,
      FileSender              = $$ALL$$,
      FileReceiver            = $$ALL$$,
      PresentationTableId     = 1,
      LocalPhysicalName       = $TOM_DIR/in/&REQNUMB.txt,
      FileRecordLength        = 2048,
      Priority                = 1,
      Mode                    = replace
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct File FileName=FILE01,ServerId=ADMIN@LIN01,FileState=E,TypeOfAllocation=D,
FileDirection=*,TypeOfFile=TV,FileOpenOption=R,FileSender=$$ALL$$,FileReceiver=$$ALL$$,PresentationT
ableId=1,LocalPhysicalName=$TOM_DIR/in/&REQNUMB.txt,FileRecordLength=2048,Priority=1,Mode=replace
_Correct file FILE01 replaced
```

Example2

```
file
      FileName                = FILE01,
      ServerId                = ADMIN@LIN01,
      Mode                    = get
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct File FileName=FILE01,ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 1
File
      FileName                = FILE01,
      ServerId                = ADMIN@LIN01,
      FileState               = E,
      TypeOfAllocation        = D,
      FileDirection           = *,
      TypeOfFile              = TV,
      FileOpenOption          = R,
      FileSender              = $$ALL$$,
      FileReceiver            = $$ALL$$,
      PresentationTableId     = 1,
      LocalPhysicalName       = $TOM_DIR/in/&REQNUMB.txt,
      FileRecordLength        = 2048,
      StartTransmitExit       = ,
      EndTransmitExit         = ,
      StartReceiveExit        = ,
      EndReceiveExit          = ,
      StartTransmitCommand    = ,
      EndTransmitCommand      = ,
      StartReceiveCommand     = ,
      EndReceiveCommand       = ,
      Priority                = 1,
      RemotePhysicalName      = ,
      FtpOptions              = ***,
      ParamFileUsed           = false,
      SpaceAllocationUsed     = false,
      FtpStoreUniqueUsed      = false,
      FileAgentUsed           = false,
      TypeOfNotification      = 0,
      Mode                    = replace
```

Journal

Use « journal » to display the journal records. The access key to the journal file is « RequestNumber ».

Format

```
journal
      RequestNumber=(reqno|wildcard),
      ServerId=id,
      Requestor==(name|*),
      FileName=(name|*),
      PartnerName=(name|*),
      TransferDirection=(T|R|*)
      Limit=(0|Limit),
      Interval=(0|seconds),
      MinimumDate=(date|*),
      MaximumDate=(date|*),
      Status==(status|*),
      Ofmt=(std|xml|col|col:hdr]),
      Mode=(get|list)
```

Parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
RequestNumber	REQN	Transfer request number	12	8	S
Requestor	REQR	Name of the requestor of the transfer request (User name, process name, …)	8		S
FileName	FNAM	Symbolic file name	8	8	S
PartnerName	PNAM	Symbolic partner name	8	8	S
TransferDirection	TDIR	Transfer direction (T : Transmission, R : Reception, * : Both)	1	1	C
Limit	LIM	Maximum number of journal records expected. (0 : no limit)	12	12	L
Interval	INVL	Interval in seconds, to select the last records written in the journal. (0 : no interval)	12	12	i
MinimumDate	MIND	Selection begin date YYYY/MM/DDHH:MM:SS, AA /MM/JJ or * YY/MM/DD is equivalent to YY/MM/DD00:00:00	18	18	D
MaximumDate	MAXD	Selection end date YYYY/MM/DDHH:MM:SS, AA /MM/JJ or * YY/MM/DD is equivalent to YY/MM/DD00:00:00	18	18	D
Status	RSTA	Status of the transfer, or * Unix : A : Wait selecction, H : Held, K : wait restart, C : Executing, E : Ended, D : Differed, J : waiting , O : Interrupted		1	C
Ofmt	OFMT	Format of the information by cxcmd - std (standard)	64	64	S
Mode	MODE	get or list	4	4	S

Parameters ServerId and Mode are required.

Example1

```
journal
      ServerId          = ADMIN@LIN01,
      RequestNumber     = 088*,
      PartnerName       = PART01,
      Mode              = get
```

Display detailed information of transfers of day 088 with partner PART01 :

```
_Correct Connected to server ADMIN@LIN01
_Correct journal ServerId=ADMIN@LIN01,RequestNumber=088*,Mode=get
_Correct Count = 276
Journal
Transfer
      RequestNumber           = 08800109,
      TransferIdent           = 58006C,
      Requestor               = pga,
      Trc                     = 0000,
      TrcMessage              = ,
      Prc                     = 0000,
      Src                     = 0000,
      Nrc                     = 0000,
      TcpipRc                 = 0000,
      X25Rc                   = ,
      X25Cause                = ,
      X25Diagnostic           = ,
      Status                  = E,
      StatusMessage           = Ended,
      TypeOfUser              = E,
      TypeOfRequest           = N,
      TransferDirection       = R,
      Priority                = 1,
      TypeOfLink              = T,
      FileName                = FILE02,
      PhysicalName            = $TOM_DIR/in/A8800109.tmp,
      LocalPhysicalName       = $TOM_DIR/in/A8800109.tmp,
      RemotePhysicalName      = ,
      FileLabel               = $TOM_DIR/config/sysin.txt,
      TransferOrigin          = BOUCLE,
      TransferDestination     = BOUCLE,
      TransferSender          = BOUCLE,
      TransferReceiver        = BOUCLE,
      FileOrganization        = S,
      TypeOfFile              = B,
      TypeOfCompression       =  ,
      TypeOfData              = B,
      FileRecordLength        = 20000,
      NetworkMessageSize      = 50000,
      LocalName               = BOUCLE,
      PartnerName             = BOUCLE,
      TypeOfPartner           = O,
      Protocol                = 0,
      CrcUsed                 = false,
      TcpipPort               = 6677,
      TcpipHostName           = ,
      TcpipAddress            = 127.0.0.1,
      X25RemoteAddress        = ,
      X25LocalAddress         = ,
      X25LocalPort            = ,
      X25UserDataField        = ,
      X25Facilities           = ,
      TransferBeginningDate   = "2006/09/28 15:00:01",
      TransferEndDate         = "2006/09/28 15:00:01",
      RetryNumber             = 0,
      FileOpenOption          = R,
      NetworkBytes            = 000002992314,
      FileNumberOfRecords     = 150,
      DateOfExecution         = "2006/09/28 15:00:01",
      PartnerPassword         = ,
      UserDataReceived        = ,
      UserDataSent            = ,
      OriginPhysicalName      = ,
      FileApi                 = ,
      ProcessId               = 6197,
      FileAgentUsed           = false,
      MultiArticleUsed        = false,
      TranslationTableNumber  = 0,
      TypeOfNotification      =  ,
      FtpOptions              = ,
      FtpStoreUniqueUsed      = false,
      SpaceAllocationUsed     = false,
      SessionTimer            = 0,
      TransferTimer           = 0,
      MaxRetries              = 0,
      SsslparmId              = ,
      XlocalPhysicalNameOvf   = false,
       XlocalPhysicalName       = /u1/pga/TOM140/tom1/config/sysin.txt,
      Mode                    = get
Transfer
      RequestNumber           = 08800108,
      TransferIdent           = 58006C,
      Requestor               = pga,
      Trc                     = 0000,
…
```

Example2

```
journal
      ServerId          = ADMIN@WIN01,
      RequestNumber     = 088*,
      Ofmt              = col:hdr,
      Mode              = list
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct journal ServerId=ADMIN@LIN01,RequestNumber=088*,Ofmt=col:hdr,Mode=list
_Correct Count = 276
-------------------------------------------------------------------------------
Request       Requestor  FileName  PartnerName  Dir  Status Date
-------------------------------------------------------------------------------
08800109      pga        FILE02    BOUCLE       R    E      2006/09/28 15:00:01
08800108      pga        FILE02    BOUCLE       T    E      2006/09/28 15:00:01
08800107      pga        FILE02    BOUCLE       R    E      2006/09/28 15:00:01
...
```

NodeVersion

Use « nodeversion » to display the version number of a monitor.

Format

```
nodeversion
       ServerId=id,
       Ofmt=(std|xml),
       Mode=get
```

Parameters ServerId and Mode are required.

Parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returnde by cxcmd - std (standard)	64	64	S
Mode	MODE	get	3	3	S

Example1

```
nodeversion
      ServerId          = ADMIN@LIN01,
      Mode              = get
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct nodeversion ServerId=ADMIN@LIN01,Mode=get
NodeVersion
      NodeName                = 10.87.15.119:9000,
      OsType                  = UNIX,
      Version                 = "C:E/UNIX 142-1",
      Mode                    = get
```

Partner

Use « partner » to manage the symbolic partners of a monitor. The access key to the symbolic partners table is the symbolic partner name « PartnerName ».

Format

```
partner
      PartnerName=name,
      ServerId=id,
      PartnerPassword=password,
      LocalNameType=(D|S),
      LocalName=name,
      LocalPassword=password,
      PartnerComment=text,
      PartnerState=(E|H),
      TypeOfPartner=(O|T),
      RestartUsed=(true|false),
      Protocol=(D|E|3|F),
      MaxSession=count,
      MaxSessionIn=count,
      MaxSessionOut=count,
      SessionTableId=id,
      TypeOfLink=(T|X|P|M|S),
      TcpipAddress=addr,
      TcpipHostName=name,
      TcpipPort=port,
      SnaLuName=name,
      AppcModeName=name,
      AppcTpName=name,
      X25LocalAddress=addr,
      X25LocalPort=port,
      X25RemoteAddress=addr,
      X25UserDataField=udf,
      X25Facilities=fac,
      FtpDefaultFile=file-name,
      RetryNumber=cont,
      SessionTimer=tim,
      TransferTimer=tim,
      SslParmId=name,
      SslUsed=(true|false),
      RemoteClientSubjectDn=crit,
      RemoteClientRootDn=crit,
      RemoteServerSubjectDn=crit,
      RemoteServerRootDn=crit,
      Ofmt=(std|xml),
      Mode=(create|replace|update|delete|get|list)
```

Modes Create, Replace and Update - parameters

Long name	Short name	Description	Win	Unix	Data Type
PartnerName	PNAM	Symbolic partner name. Character range : A-Z,0-9,Blank	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
PartnerPassword	PPSW	Symbolic partner password. Character range : A-Z,0-9,Blank	8	8	S
LocalNameType	LNTP	Type of the local identification (D:Dynamic, S:Static)	1		C
LocalName	LNAM	Local alias name of Connect:Express	8	8	s
LocalPassword	LPSW	Local alias password of Connect:Express	8	8	s
PartnerComment	PTXT	Description of the symbolic partner entry	80		S
PartnerState	PSTA	Status of the partner (E : Enabled, H : Hold)	1	1	C
TypeOfPartner	TYPP	Type of partner (O : Non Connect:Express, T : Connect:Express)	1	1	C
RestartUsed	RETO	Automatic restart option (true, false)	5		B
Protocol	PROT	Transfer protocol Unix: 1: Etebac3, 2: FTP, 3: PeSIT. Note: on Unix, PeSIT version, D or E, is defined in the session table Windows : D: PeSIT-D, E: PeSIT-E, 3: Etebac3.	1	1	C
MaxSession	MSES	Maximum simultaneous sessions	3	2	i
MaxSessionIn	MSIN	Maximum simultaneous inbound sessions	3	2	i
MaxSessionOut	MSOU	Maximum simultaneous outbound sessions	3	2	i
SessionTableId	STAB	Name or identification number of the session table. Note : On Unix the session table are from « 0 » to « 9 »	50	1	S
TypeOfLink	TYPL	Type of network link Unix : T : TCP/IP, X : X25, P : PAD, M : TCP/IP+X25 Windows: T: TCP/IP, S: SNA LU6.2	1	1	C
TcpipAddress	TCPA	TCP/IP address of the remote partner	15	15	S
TcpipHostName	TCPH	TCP/IP host name of the remote partner	127	32	s
TcpipPort	TCPP	TCP/IP port of the remote partner	5	5	i
SnaLuName	SNAL	SNA address of the remote partner	8		S
AppcModeName	APPM	LU6.2 mode name of the remote partner	8		S
AppcTpName	APPT	LU6.2 transaction program name of the remote partner	64		S
X25LocalAddress	X25L	Local X25 address Character range : 0-9	15	15	S
X25LocalPort	X25P	Local X25 port	2	1	S
X25RemoteAddress	X25A	X25 address of the remote partner Character range : 0-9	15	15	S
X25UserDataField	X25U	X25 user data Character range: 0-9,A-F	8	8	S
X25Facilities	X25F	X25 Facilities Character range: 0-9,A-F	32	16	S
FtpDefaultFile	FTDF	Default FTP profile name Character range: A-Z,0-9,Blank		8	S
RetryNumber	RETN	Maximum number of retries for this partner		8	i
SessionTimer	STMR	Session retry timer		2	i
TransferTimer	TTMR	Transfer retry timer		2	i
SsslParmId	SSLP	SSLPARM, SSL profile identifier Character range: A-Z,0-9,Blank	8	8	S
SslUsed	SSLO	SSL option (true, false)	5		B
RemoteClientSubjectDn	RCSD	Criteria for remote client subject DN control	256		s
RemoteClientRootDn	RCRD	Criteria for remote client root DN control	256		s
RemoteServerSubjectDn	RSSD	Criteria for remote server subject DN control	256		s
RemoteServerRootDn	RSRD	Criteria for remote server root DN control	256		s
Ofmt	OFMT	Format of the information returned by cxcmd	64	64	S

Mode	MODE	create, replace or update	7	7	S

For modes **create** and **replace**, the following parameters are required: PartnerName, ServerId, Mode, TypeOfPartner, Protocol, SessionTableId, TypeOfLink.

On Windows :
For TCP/IP, TcpipAddress or TcpipHostName is required.
For SNA, SnaLuName and AppcModeName are required.
For X25, X25LocalPort is required.

For mode **update** , the following parameters are required: PartnerName, ServerId and Mode.

Defaults :
For modes **create** and **replace,** the following non-standard defaults apply, if the corresponding parameter is omitted :
PartnerState = E (Unix,Windows).
RestartUsed=true, MaxSession=2 , MaxSessionIn=1, MacSessionOut=1 (Windows).

Modes Delete - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
PartnerName	PNAM	Symbolic partner name	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
Mode	MODE	delete	6	6	S

Parameters PartnerName, ServerId and Mode are required.

Modes Get and List - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
PartnerName	PNAM	Symbolic partner name	8	8	S
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard)	64	64	S
Mode	MODE	get or list	4	4	S

Parameters PartnerName, ServerId and Mode are required.

Example1

```
Partner
      PartnerName             = PART05,
      ServerId                = ADMIN@LIN01,
      PartnerPassword         = PSW,
      LocalName               = PART01,
      LocalPassword           = PSW,
      PartnerState            = E,
      TypeOfPartner           = O,
      Protocol                = 3,
      MaxSession              = 20,
      MaxSessionIn            = 10,
      MaxSessionOut           = 10,
      SessionTableId          = 1,
      TypeOfLink              = T,
      TcpipAddress            = 10.87.15.72,
      TcpipPort               = 06677,
      Mode                    = replace
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct Partner PartnerName=PART05,ServerId=ADMIN@LIN01,PartnerPassword=PSW,
LocalName=PART01,LocalPassword=PSW,PartnerState=E,TypeOfPartner=O,Protocol=3,MaxSession=20,MaxSessio
nIn=10,MaxSessionOut=10,SessionTableId=1,TypeOfLink=T,TcpipAddress=10.87.15.72,TcpipPort=06677,Mode=
replace
_Correct partner PART05 replaced
_Correct file FILE01 replaced
```

Example2

```
partner
      PartnerName             = PART05,
      ServerId                = ADMIN@LIN01,
      Mode                    = get
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct Partner PartnerName=PART05,ServerId=ADMIN@LIN01,Mode=get
_Correct Count = 1
Partner
      PartnerName             = PART05,
      ServerId                = ADMIN@LIN01,
      PartnerPassword         = PSW,
      LocalName               = PART01,
      LocalPassword           = PSW,
      PartnerState            = E,
      TypeOfPartner           = O,
      Protocol                = 3,
      MaxSession              = 20,
      MaxSessionIn            = 10,
      MaxSessionOut           = 10,
      SessionTableId          = 1,
      TypeOfLink              = T,
      TcpipAddress            = 10.87.15.72,
      TcpipPort               = 6677,
      TcpipHostName           = ,
      X25LocalAddress         = ,
      X25RemoteAddress        = ,
      X25LocalPort            = ,
      X25UserDataField        = ,
      X25Facilities           = ,
      FtpDefaultFile          = ,
      RetryNumber             = 0,
      SessionTimer            = 0,
      TransferTimer           = 0,
      SslparmId               = ,
      Mode                    = replace
```

Example3

```
partner
      PartnerName             = *,
      ServerId                = ADMIN@LIN01,
      Mode                    = list
```

Display :

```
_Correct Connected to server ADMIN@LIN01
_Correct Partner PartnerName=*,ServerId=ADMIN@LIN01,Mode=list
_Correct Count = 12
PartnerList
      PartnerName             = BCLDSA,
      PartnerName             = BOUCLE,
      PartnerName             = BOUCLE0,
      PartnerName             = BOUCLE1,
      PartnerName             = BOUCLE2,
      PartnerName             = DNSY,
      PartnerName             = DYSN,
      PartnerName             = ETB01,
      PartnerName             = ETB02,
      PartnerName             = FTP01,
      PartnerName             = GFIPSR4S,
      PartnerName             = PART05
```

Presentation

Use « presentation » to manage PeSIT and Etebac3 presentation tables of an Unix monitor, and Etebac3 presentation tables of a Windows monitor. The access key to the presentation table is « PresentationTableId ».
For Connect:Express Unix the presentation tables names are fixed, from « 0 » to « 9 » : you can neither create nor delete them. For Connect :Express windows, the presentation tables are identified by the user with any string : you can create and delete them.

Note : use « etb3presentation » to manage the Etebac3 presentation tables of Windows monitor.

Format

```
presentation
      PresentationTableId=id,
      ServerId=id,
      TypeOfCompression=(N|H|V|M),
      ConcatenationUsed=(true|false),
      MultiArticleUsed=(true|false),
      SegmentationUsed=(true|false),
      TranslationUsed=(true|false),
      TranslationToEbcdic=name,
      TranslationToAscii=name,
      Ofmt=(std|xml),
      Mode=(create|replace|update|delete|get|list)
```

Modes Create (Windows), Replace and Update - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
PresentationTableId	PTAB	Name or identification number of the presentation table	50	1	S
ServerId	SRVID	Identification of the server	128	128	S

TypeOfCompression	TYPC	Type of compression (N :None, H : Horizontal, V : Vertical, M : Horizontal and vertical)	C	C	C
ConcatenationUsed	CONC	Concatenation option for PeSIT FPDU (true/false)	5		B
MultiArticleUsed	MULT	Multi-article option for PeSIT FPDU (true/false)	5	5	B
SegmentationUsed	SEGM	Segmentation option for PeSIT FPDU (true/false)	5		B
TranslationUsed	TRAO	Transcodage option (true/false)	5		B
TranslationToEbcdic	TRAE	Name or identification number of the ASCII – EBCDIC translation table	127	1	S
TranslationToAscii	TREA	Name or identification number of the EBCDIC – ASCII translation table	127		S
Mode	MODE	create (Windows uniquement), replace, update	7	7	S

For modes **create** and **replace** , the following parameters are required:

Unix replace: PresentationTableId, ServerId, TypeOfCompression, TranslationToEbcdic, and Mode.

Windows : PresentationTableId, ServerId and Mode.

For mode **update** , the following parameters are required: PresentationTableId, ServerId and Mode.

Defaults :
For modes **create** and **replace,** the following non-standard defaults apply, if the corresponding parameter is omitted : TypeOfCompression=N, TranslationToEbcdic=TOMNT.ASC, TranslationToAscii=TOMNT.ASC (Windows).

Modes Delete (Windows) - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
PresentationTableId	PTAB	Name of the presentation table	50		s
ServerId	SRVID	Identification of the server	128		S
Mode	MODE	delete (Windows uniquement)	6		S

Parameters PresentationTableId,ServerId and Mode are required.

Modes Get and List - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
PresentationTableId	PTAB	Name or identification number of the presentation table, or wildcards	50	1	s
ServerId	SRVID	Identification of the server	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard)	64	64	S
Mode	MODE	get, or list	4	4	S

Parameters PresentationTableId,ServerId and Mode are required.

Example1

```
presentation
      PresentationTableId     = *,
      ServerId          = ADMIN@WIN01,
      Mode              = get
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct presentation PresentationTableId=*,ServerId=ADMIN@WIN01,Mode=get
_Correct Count = 5
Presentation
      PresentationTableId     = "PeSIT presentation with mixed compression",
      ServerId                = ADMIN@WIN01,
      TypeOfCompression       = M,
      MultiArticleUsed        = true,
      TranslationToEbcdic     = TOMNT.ASC,
      ConcatenationUsed       = true,
      SegmentationUsed        = true,
      TranslationUsed         = false,
      TranslationToAscii      = TOMNT.EBC,
      Mode                    = replace
Presentation
      PresentationTableId     = "PeSIT presentation without compression",
      ServerId                = ADMIN@WIN01,
      TypeOfCompression       = N,
      MultiArticleUsed        = true,
      TranslationToEbcdic     = TOMNT.ASC,
      ConcatenationUsed       = true,
      SegmentationUsed        = true,
      TranslationUsed         = false,
      TranslationToAscii      = TOMNT.EBC,
      Mode                    = replace
…
```

Example2

```
Presentation
      PresentationTableId     = "PeSIT presentation with mixed compression",
      ServerId                = ADMIN@WIN01,
      TypeOfCompression       = N,
      MultiArticleUsed        = false,
      TranslationUsed         = false,
      Mode                    = update
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct Presentation PresentationTableId="PeSIT presentation with mixed compression",
ServerId=ADMIN@WIN01,TypeOfCompression=N,MultiArticleUsed=false,TranslationUsed=false,Mode=update
_Correct presentation PeSIT presentation with mixed compression updated
```

Server

Use « server » to configure locally the session parameters to logon to a Connect:Express API server. Session information are stored in the cxcmd_servers.txt file placed in the directory where cxcmd has been installed. The access key to the servers table is « ServerId ».
Connect:Express Unix does not control the client name and password. Use UserName=ADMIN, UserPassword=ADMIN.

Format

```
server
      ServerId=id,
      Address=(ip-address|hostname),
      Port=port,
      OSType=(UNIX|WINDOWS),
      UserName=name,
      UserPassword=password,
      Limit=(0|count),
      Timeout=(0|seconds),
      LocalPortRange=(0|port-range-string),
      Mode=(create|replace|update|delete|get|list|connect|disconnect)
```

Modes Create, Replace and Update - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Identification of the server	128	128	S
Address	ADDR	TCP/IP address or host name	128	128	s
Port	PORT	TCP/IP port	5	5	i
OSType	OS	Operating system of the server (UNIX or WINDOWS)	7	7	S
UserName	UNAM	Client name	8	8	S
UserPassword	UPWD	Client password	8	8	S
Trace	TRC	Trace option	5	5	B
Limit	LIM	Maximum number of statistics records expected. (0 = no limit)	12	12	L
Timeout	TIMO	Session timeout in seconds. (0 = no timeout. Default=30 seconds)	5	5	i
LocalPortRange	LPRG	Local port range to use. Default = 0 : the system determines the port number automatically . Examples : 6670-6700 ; 6670-6700|7000|8000-9000	64	64	S
Mode	MODE	create, replace or update	7	7	S

For modes **create** and **replace**, the following parameters are required: ServerId, Address, Port, OSType, UserName, UserPassword and Mode.

For mode **update** , the following parameters are required : ServerId and Mode.

Modes Delete, Connect and Disconnect - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Identification of the server	128	128	S
Mode	MODE	Delete	6	6	S

Parameters ServerId and Mode are required.

Note : Use mode « connect » to check the link with a server.

Modes Get and List - parameters

Long Name	Short Name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server (Wildcard supported for modes get and list)	128	128	S
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard)	64	64	S
Mode	MODE	get or list	7	7	S

Parameters ServerId and Mode are required.

Example1

```
server
     ServerId      = ADMIN@WIN01,
     Address       = 10.87.15.92,
     Port          = 7000,
     OSType        = WINDOWS,
     UserName      = ADMIN,
     UserPassword  = ADMIN,
     Mode              = replace
```

Display :

```
_Correct                              server                              ServerId=ADMIN@WIN01,
Address=10.87.15.92,Port=7000,OSType=WINDOWS,UserName=ADMIN,
UserPassword=ADMIN,Mode=replace
```

Example2

```
server
      ServerId          = *,
      Mode              = get
```

Display :

```
Server
      ServerId                = ADMIN@WIN01,
      Address                 = 10.87.15.92,
      Port                    = 7000,
      OSType                  = WINDOWS,
      Trace                   = false,
      Limit                   = 0,
      UserName                = ADMIN,
      UserPassword            = ********,
      Timeout                    = 30,
      LocalPortRange          = 0,
      Mode                    = replace
Server
      ServerId                = ADMIN@LIN01,
      Address                 = 10.87.15.119,
      Port                    = 9000,
      OSType                  = UNIX,
      Trace                   = false,
      Limit                   = 0,
      UserName                = ADMIN,
      UserPassword            = ********,
      Timeout                    = 30,
      LocalPortRange          = 0,
      Mode                    = replace
…
```

Session

Use « session » to manage session tables of a monitor. The access key is « SessionTableId ».
For Connect:Express Unix the session tables are fixed, identified from « 0 » to « 9 ». You can neither create nor delete them. For Connect:Express windows, the session tables are identified by the user with any string : you can create and delete them.

Format

```
session
      SessionTableId=id,
      ServerId=id,
      BaseMessageSize=size,
      BaseSynchronizationSize=size,
      SessionDirection=(T|R|*),
      CrcUsed=(true|false),
      ResynchronizationNumber=count,
      BaseWindowSize=size,
      ProtocolVersion=(1|2),
      RetryNumber=count,
      Ofmt=(std|xml),
      Mode=(create|replace|update|delete|get|list)
```

Modes Create (Windows), Replace and Update - parameters

Long name	Short name	Description	Win	Unix	Data Type
SessionTableId	STAB	Name or identification number of the session table	50	1	S
ServerId	SRVID	Identification of the server	128	128	s
BaseMessageSize	BMSG	Network message size (negotiated) Windows : 256-4096, Unix : 0-6535	4	5	i
BaseSynchronizationSize	BSNC	Synchronization buffer size in KBytes (negotiated) Windows : 1-32, Unix : 0-99	2	2	i
SessionDirection	SDIR	Direction (T : Transmission, R : Reception, * : Both)	1		C
CrcUsed	CRCO	CRC option (true/false)	5	5	B
ResynchronizationNumber	RSYN	Number of resynchronisations for a transfer request	2		i
BaseWindowSize	BWIN	Window size (negotiated) 0-16		2	i
ProtocolVersion	PVER	PeSIT protocol version (1 : PeSIT-D,2 : PeSIT-E)		1	C
RetryNumber	RETN	Number of retries for a transfer request		2	i
Mode	MODE	create, replace or update	7	7	S

For modes **create** and **replace**, the following parameters are required:

Unix replace: SessionTableId, ServerId, TypeOfCompression, TranslationToEbcdic, Mode

Windows : SessionTableId, ServerId and Mode

For mode **update** , the following parameters are required: SessionTableId, ServerId and Mode.

Defaults :
For mode **update** the following non-standard default apply, if the corresponding parameter is omitted :

ProtocolVersion=2 (Unix)

Modes Delete (Windows) - parameters

Long name	Short name	Description	Win	Unix	Data Type
SessionTableId	STAB	Name or identification number of the session table	50		S
ServerId	SRVID	Identification of the server	128		s
Mode	MODE	delete	6		S

Parameters SessionTableId, ServerId and Mode are required.

Modes Get and List - parameters

Long name	Short name	Description	Win	Unix	Data Type
SessionTableId	STAB	Name or identification number of the session table	50	1	S
ServerId	SRVID	Identification of the server	128	128	s
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard)	64	64	S
Mode	MODE	get or list	4	4	S

Parameters SessionTableId, ServerId and Mode are required.

Example1

```
session
      SessionTableId    = *,
      ServerId          = ADMIN@WIN01,
      Mode              = get
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct session SessionTableId=*,ServerId=ADMIN@WIN01,Mode=get
_Correct Count = 3
Session
      SessionTableId          = "PeSIT session (TCP/IP)",
      ServerId                = ADMIN@WIN01,
      BaseMessageSize         = 4096,
      BaseSynchronizationSize = 32,
      CrcUsed                 = false,
      SessionDirection        = *,
      ResynchronizationNumber = 0,
      Mode                    = replace
Session
      SessionTableId          = "PeSIT session (LU6.2)",
      ServerId                = ADMIN@WIN01,
      BaseMessageSize         = 4094,
      BaseSynchronizationSize = 32,
      CrcUsed                 = false,
      SessionDirection        = T,
      ResynchronizationNumber = 0,
      Mode                    = replace
Session
      SessionTableId          = "PeSIT session (X25)",
      ServerId                = ADMIN@WIN01,
      BaseMessageSize         = 4096,
      BaseSynchronizationSize = 32,
      CrcUsed                 = false,
      SessionDirection        = *,
      ResynchronizationNumber = 0,
      Mode                    = replace
```

Example2

```
Session
      SessionTableId          = "PeSIT session (TCP/IP)",
      ServerId                = ADMIN@WIN01,
      BaseMessageSize         = 2048,
      Mode                    = update
```

Display :

```
_Correct Connected to server ADMIN@WIN01
_Correct Session SessionTableId="PeSIT session (TCP/IP)",ServerId=ADMIN@WIN01,
BaseMessageSize=2048,Mode=update
_Correct session PeSIT session (TCP/IP) updated
```

Statistics

Use « statistics » to display statistics : this is possible only if the statistics are activated in the monitor configuration.

Note : The statistics file of a monitor contains a lot of lines, and it is advised to limit the number of lines returned with « limit » .

Format

```
statistics
       ServerId=id,
       MinimumDate=(date|*),
       MaximumDate=(date|*),
       RequestNumber=(reqno|*),
       Limit=(0|Limit),
       Interval=(0|seconds),
       Limit=limit,
       Ofmt=(raw|std|xml),
       Mode=get
```

Parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
MinimumDate	MIND	Selection begin date YYYY/MM/DDHH:MM:SS, AA /MM/JJ, or * YY/MM/DD is equivalent to YY/MM/DD00:00:00	18	18	D
MaximumDate	MAXD	Selection end date YYYY/MM/DDHH:MM:SS, AA /MM/JJ, or * YY/MM/DD is equivalent to YY/MM/DD00:00:00	18	18	D
RequestNumber	REQN	Request number, or *	12	8	S
Interval	INVL	Interval in second, to select the last records written in the statistics file. (0 : no interval)	12	12	i
Limit	LIM	Maximum number of statistics record expected. (0 = no limit)	12	12	L
Ofmt	OFMT	Format of the information returned by cxcmd raw : statistics lines are not formatted std : standard format xml : xml format	64	64	S

Mode	MODE	Get	3	3	S

Parameters ServerId and Mode are required.

Example1

```
statistics
      ServerId         = ADMIN@LIN01,
      Interval         = 30,
      Ofmt             = raw,
      Mode             = get
```

Display statistics during last 30 seconds, not formatted:

```
_Correct Connected to server ADMIN@LIN01
_Correct statistics Interval=48000,ServerId=ADMIN@LIN01,Ofmt=raw,Mode=get
RDA=2006/09/2910:56:27,RID=SB,MID=UXOL134I,TOM=FEDORA,OS=Unix,REQN=08900002,RSTA=,EVT=Connecti
onStarted,TXT=Communication started - COMMUNICATION OPENED (I) WITH: BOUCLE    REQ: 08900002
PESIT   TCPIP,PNAM=BOUCLE,LNAM=BOUCLE,TYPP=O,PROT=E,TYPL=T,TCPA=127.0.0.1,TCPP=06677,RETN=0
RDA=2006/09/2910:56:27,RID=SB,MID=UXPL133I,TOM=FEDORA,OS=Unix,REQN=08900001,RSTA=,EVT=Connecti
onStarted,TXT=Communication started - COMMUNICATION OPENED (O) WITH: BOUCLE    REQ: 08900001
PESIT   TCPIP,PNAM=BOUCLE,LNAM=BOUCLE,TYPP=O,PROT=E,TYPL=T,TCPA=127.0.0.1,TCPP=06677,RETN=0
```

Example2

```
statistics
      ServerId         = ADMIN@LIN01,
      Interval         = 30,
      Mode             = get
```

Display statistics during last 30 seconds, standard format:

```
_Correct Connected to server ADMIN@LIN01
_Correct statistics Interval=30,ServerId=ADMIN@LIN01,Mode=get
Statistics
      DateOfStatisticsRecord  = 2006/09/2910:55:25,
      RecordId                = NUIC,
      MessageId               = UMBL124I,
      MonitorId               = FEDORA,
      OperatingSystem         = Unix,
      Event                   = NodeStatus,
      MessageText               = "Monitor information - C:E INITIALIZATION COMPLETE  V142-1
RUN=(C)",
      AliasName               = "$SSL-LABS LINUX",
      Dpcsid                  = FEDORA,
      Dpcpsw                  = PSW,
      FtpDefaultFile          = FTPFILE,
      MaxTransfer             = 6,
      LogSize                 = 10000,
      SessionTimer            = 1,
      TransferTimer           = 1,
      RunType                 = C,
      FtpListAllUsed          = 1,
      TraceUsed               = 0,
      SyslogUsed              = 0,
      BuildDatabaseUsed       = 0,
      TcpipListeningAddress   = ,
      TcpipListeningPort      = 6677,
      FtpListeningAddress     = ,
      FtpListeningPort        = 0,
      ApiAddress              = ,
      ApiPort                 = 9000,
      NotificationsUsed       = 0,
      ProductInfo             = "C:E/UNIX 142-1 ",
      SystemInfo             =  "Sysname:Linux.Release:2.6.17-1.2157_FC5.Version:#1 Tue Jul 11
22:55:46 EDT 2006.Machine:i686.Nodename:localhost.localdomain",
Statistics
      DateOfStatisticsRecord  = 2006/09/2910:55:25,
      RecordId                = NUIS,
      MessageId               = UMIL125I,
      MonitorId               = FEDORA,
      OperatingSystem         = Unix,
      Event                   = NodeStatus,
      MessageText                    =  "Monitor  information  -  LARGE  FILE  (>2Gbytes)  SUPPORT
AVAILABLE",
Statistics
      DateOfStatisticsRecord  = 2006/09/2910:56:27,
      RecordId                = PS,
      MessageId               = URKL021I,
      MonitorId               = FEDORA,
      OperatingSystem         = Unix,
      RequestNumber           = 08900001,
      Status                  = A,
      Event                   = NodeCommand,
      MessageText                 = "Request  accepted  - REQUEST  08900001  <-  pga        ACCEPTED
(N)",
      PartnerName             = BOUCLE,
      FileName                = FILE02,
      TransferDirection       = T,
      Requestor               = pga,
      Priority                = 1,
      TypeOfRequest           = N,
      DateOfExecution         = 2006/09/2910:56:27,
      FileLabel               = $TOM_DIR/config/sysin.txt,
      TypeOfFile              = BU,
      FileRecordLength        = 20000,
      UserDataSent            = ,
      UserDataReceived        = ,
      LocalPhysicalName       = $TOM_DIR/config/sysin.txt,
      FileApi                 = ,
      TransferSender          = ,
      TransferReceiver        = ,
      LocalName               = BOUCLE,
      TransferOrigin          = BOUCLE,
      TransferDestination     = BOUCLE,
      TypeOfLink              = T,
      FtpOptions              = ***,
      FtpStoreUniqueUsed      = N,
      TypeOfUser              = I,
Statistics
      DateOfStatisticsRecord  = 2006/09/2910:56:27,
      RecordId                = SB,
      MessageId               = UXOL134I,
      MonitorId               = FEDORA,
      OperatingSystem         = Unix,
      RequestNumber           = 08900002,
      Status                  = ,
      Event                   = ConnectionStarted,
      MessageText              = "Communication started - COMMUNICATION OPENED (I) WITH: BOUCLE
REQ: 08900002 PESIT   TCPIP",
      PartnerName             = BOUCLE,
      LocalName               = BOUCLE,
      TypeOfPartner           = O,
      Protocol                = E,
      TypeOfLink              = T,
      TcpipAddress            = 127.0.0.1,
      TcpipPort               = 06677,
      RetryNumber             = 0,
…
```

Transfer

Use « transfer » to submit transfer requests, interrupt requests, restart requests or purge requests.

For Connect:Express Windows, you can purge transfers from the activity list (CHK file), but not from the journal.
For Connect:Express Unix, you can purge transfers from the RENC file in which both active transfers and journal records are written.

Format

```
transfer
      ServerId=id,
      FileName=name,
      PhysicalName=file-name,
      TransferDirection=(T|R),
      PartnerName=name,
      Priority=(0|1|2),
      TypeOfRequest=(N|I|H),
      TypeOfLink=(T|X|P|M|S),
      Requestor=name,
      UserRequestId=id,
      LocalName=name,
      LocalPassword=pwd,
      TypeOfFile=(TF|TV|BF|BU),
      FileRecordLength=(0|length),
      FileApi=param,
      FileLabel=label,
      DateOfExecution=date,
      TypeOfNotification=(type),
      NotifyUsed=(true|false),
      ClientToNotify=name,
      FtpStoreUniqueUsed=(true|false),
      FtpOptions=type-struct-mode,
      Pi99Offset=id,
      Pi99Length=id,
      Pi99Value=id,
      AdHocUser=name,
      AdHocPassword=pwd,
      OriginPhysicalName=file-name,
      RemotePhysicalName=file-name,
      TransferOrigin=org,
      TransferDestination=dest,
      TransferSender=sender,
      TransferReceiver=receiver,
      FileAgentUsed=(true|false),
      RequestNumber=reqno,
      Status=status,
      NotAfter=date,
      DaysToKeep=days,
      Mode=(submit|interrupt|restart|purge)
```

Mode Submit - parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
FileName	FNAM	Symbolic file name	8	8	S

PhysicalName	PHN	Physical file name	127	44	s
TransferDirection	TDIR	Transfer direction (T : Transmission, R : Reception)	1	1	C
PartnerName	PNAM	Symbolic partner name	8	8	S
Priority	PRIO	Priority (0 : Urgent, 1 : Normal, 2 : Low)	1	1	C
TypeOfRequest	TYPR	Type of request (N: Normal, I: Inquiry, H: Hold) If TransferDirection=T, TypeOfRequest=I is not valid	1	1	C
TypeOfLink	TYPL	Type of network link Unix : T : TCP/IP, X : X25, P : PAD, M : TCP/IP+X25 Windows: T: TCP/IP, S: SNA LU6.2	1	1	C
Requestor	REQR	Name of the transfer requestor (User name, process name, …)	8		S
UserRequestId	REQU	User identification of the request	16		s
LocalName	LNAM	Local alias name of Connect :Express	8	8	S
LocalPassword	LPSW	Local alias password of Connect :Express	8	8	S
TypeOfFile	TYPF	Type of file (TF : Text fixed, TV : Text variable, BF : Binary fixed, BU : Binary undefined)		2	S
FileRecordLength	FRLG	Record length If TransferDirection=T, FileRecordLength=0 is invalid		5	i
FileApi	FAPI	User description of the file transfer (Etebac3 Card)	80	88	s
FileLabel	FLAB	Label (user identification of the file)	80	80	s
DateOfExecution	DATE	Scheduling date of the request	18	18	D
TypeOfNotification	TYPN	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	C
NotifyUsed	NFYO	Notification option (true/false)	5		B
ClientToNotify	CLIN	Name of the client to notify, required if NotifyUsed=true	255		S
FtpStoreUniqueUsed	FTSU	FTP « store unique » option (true/false)		5	B
FtpOptions	FTOP	FTP Options (Type/Structure/Mode) Type :(E :Ebcdic, A :Ascii, B : Binary, * : any) Structure (F : File, R: Record, *: Any) Mode (B: Bloc, S: Stream, *: Any)		3	S
Pi99Offset	P99O	Offset in Pi99Value for pi99	3		i
Pi99Length	P99L	Length in Pi99Value for Pi99	3		i
Pi99Value	PI99	Value of pi99	254	254	s
AdHocUser	AHUS	AdHoc : remote user name	8		S
AdHocPassword	AHPW	AdHoc : remote user password	8		S
OriginPhysicalName	OPHN	File name sent to remote as local file name, or Pi99	44		s
RemotePhysicalName	RPHN	AdHoc : remote data set name	44	44	s
TransferOrigin	TORG	Originator of the transfer	8	8	S
TransferDestination	TDST	Destination of the transfer	8	8	S
TransferSender	TSND	Sender of the transfer (Pi 61)	24	24	S
TransferReceiver	TRCV	Receiver of the transfer (Pi 62)	24	24	S
FileAgentUsed	FLAO	File agent option (true/false)		5	B
Mode	MODE	submit	6	6	S

For mode **submit**, the following parameters are required: FileName, TransferDirection, TypeOfRequest and Mode.

Modes Interrupt and Restart - parameters

Long name	**Short name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Identification of the server	128	128	S
RequestNumber	REQN	Request number (not a wildcard)	12	8	S
Mode	MODE	Interrupt or restart	9	9	S

Parameters RequestNumber , ServerId and Mode are required.

Mode purge (Windows) - parameters du

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
RequestNumber	REQN	Request number (not a wildcard)	12	8	S
Mode	MODE	Purge	9	9	S

Parameters RequestNumber , ServerId and Mode are required.

Modes Purge (Unix) - parameters

Long name	Short name	Description	Win	Unix	Data Type
ServerId	SRVID	Identification of the server	128	128	S
FileName	FNAM	Symbolic file name, or *	8	8	S
PartnerName	PNAM	Symbolic partner name, or *	8	8	S
TransferDirection	TDIR	Transfer direction (T : Transmission, R : Reception, * : Both)	1	1	C
RequestNumber	REQN	Request number, or *	12	8	S
Status	RSTA	Transfer status, or *		1	S
NotAfter	NAFT	Date after which transfer must not be purged by the automatic process, or *		18	D
DaysToKeep	DAYS	Number of days to keep by the automatic process Note : The current day is included		3	S
Mode	MODE	Purge	6	6	S

Parameters ServerId and Mode are required.

Example1

```
transfer
      ServerId          = ADMIN@WIN01,
      FileName          = FILE01,
      TransferDirection = T,
      PartnerName       = BOUCLE,
      LocalName         = BOUCLE,
      PhysicalName            = c:\u2\CExpress\out\tosend.txt,
      Mode              = submit
```

Submit a transfer request and display the request number returned :

```
_Correct Connected to server ADMIN@WIN01
_Correct transfer ServerId=ADMIN@WIN01,FileName=FILE01,TransferDirection=T,
PartnerName=BOUCLE,LocalName=BOUCLE,PhysicalName=c:\u2\CExpress\out\tosend.txt,Mode=submit
_Correct Requests submitted : 200627200001
```

Example2

```
transfer
      DaysToKeep        = 7,
      ServerId          = ADMIN@LIN01,
      Mode              = purge
```

Purge the transfer requests of Connect:Express Unix, keep 7 days transfers, and display the result :

```
_Correct Connected to server ADMIN@LIN01
_Correct transfer DaysToKeep=7,ServerId=ADMIN@LIN01,Mode=purge
_Correct Requests purged
```

Version

Use « version » to display the version number of the current cxcmd utility and the version number of the current Connect:Express Jai.

Format

```
version
      Mode=get
```

Parameters

Long name	Short name	Description	Win	Unix	Data Type
Ofmt	OFMT	Format of the information returned by cxcmd – std (standard)	64	64	S
Mode	MODE	get	3	3	S

Parameter Mode is required.

Example1

```
version mode=get
```

Display the version numbers :

```
_Correct version mode=get
Version
      CxcmdVersion            = 1.3.6,
      CxjaiVersion            = 1.3.6,
      Mode                    = get
```

SslClientParameter

Use « sslclientparameter » to manage SSL client parameter definitions. The access key is the parameter name.

Format

```
sslclientparameter
  ServerId=server-id,
  ParameterName=name,
  Status=(E|H),
  CertificateId=certificate-id,
  VerificationOption=(0|1|2),
  CipherList=file-name,
  Tslv1Enabled=(true|false),
  Sslv3Enabled=(true|false),
  Sslv2Enabled=(true|false),
  SslBytesHeaderUsed=(true|false),
  StoreLocation=(SYSTEM_STORE_LOCAL_MACHINE|
                 SYSTEM_STORE_CURRENT_USER|SYSTEM_STORE_SERVICES),
  StoreName=(My),
  Subject=name,
  Issuer=name,
  Protocol=(1,3,2),
  CipherSuites=cipher-suites,
  TraceLevel=(0,1,2),
  Mode=(create|replace|update|delete|get|list)
```

Modes Create, Replace and Update - parameters

Long Name	Short Name	Description	Win	Unix	Data Type
ParameterName	SNAM	Name of this parameter definition	8	8	S
ServerId	SRVID	Server Id	128	128	S
Status	SSTA	Status, E=Enabled, H=Disabled	1		C
StoreLocation	STOL	Store location: SYSTEM_STORE_LOCAL_MACHINE, SYSTEM_STORE_CURRENT_USER or SYSTEM_STORE_SERVICES	64		S
StoreName	STNM	Store name (My)	128		S
Subject	SUBJ	Certificate subject distinguished name	256		S
Issuer	ISSU	Certificate issuer distinguished name	256		s
Protocol	PROT	Protocol – ‘1’:TLSV1, ‘3’:SSLV3, ‘2’:SSLV2	1		C
CipherSuites	CISU	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	45		s

		08: SSL_CK_RC2_128_CBC_WITH_MD5			
TraceLevel	TRLV	‘0’:None, ‘1’:Medium, ‘2’: Full	1		C
SslBytesHeaderUsed	SBHD	A 2 bytes length header is inserted before each PeSIT data sent (true/false)	1	5	B
VerificationOption	VOPT	Authentication ‘0’:VERIFY_NONE, ‘1’: VERIFY_PEER		1	C
CertificateId	CERT	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	S
CipherList	CILI	Name of a file located in $TOM_DIR/config/ssl/ciphlist and containing a list of ciphers that can be used		16	s
Tlsv1	TLS1	TLSV1 available (true/false)		5	B
Sslv3	SSL3	SSLV3 available (true/false)		5	B
Sslv2	SSL2	SSLV2 available (true/false		5	B
Mode	MODE	create, replace or update	7	7	S

In **create** mode, the following parameters are required:
Unix : ParameterName, ServerId, VerificationOption, CertificateId, Mode
Windows : ParameterName, ServerId, StoreLocation, Subject, CipherSuites and Mode

In **update** mode, the following parameters are required: ParameterName, ServerId and Mode.

Défaults :
In update mode, the following non-standard defaults apply if the parameter is not indicated in the command:
Unix :
Status=E, VerificationOption=0 (VERIFY_NONE), Tlsv1Enabled=true, Sslv3Enabled=false, Sslv2Enabled=false
Windows :
Status=E, StoreName=My, Protocol=1 (TLSV1), TraceLevel=0 (None),
Sslv2Enabled=false

Mode Delete - parameters

Long Name	Short Name	Description	Win	Unix	Data Type
ParameterName	SNAM	Name of this parameter definition	8	8	S
ServerId	SRVID	Server identification	128	128	S
Mode	MODE	Delete	6	6	S

ParameterName, ServerId and Mode are required.

Mode Get and List - parameters

Long Name	Short Name	Description	Win	Unix	Data Type
ParameterName	SNAM	Name of this parameter definition or simple wildcard	8	8	S
ServerId	SRVID	Server Id	128	128	s
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard) or xml	64	64	S
Mode	MODE	get or list	4	4	S

ParameterName, ServerId and Mode are required.

Examples

The command:

```
sslclientparameter
```

```
      ServerId          = ADMIN@WIN01,
      ParameterName     = *,
      ofmt              = std,
      Mode              = list
```

Displays:

```
_Correct Connected to server ADMIN@WIN01
_Correct sslclientparameter serverid=ADMIN@WIN01,parametername=*,ofmt=std,mode=list
_Correct Count = 15
SslClientParameterList
      ParameterName           = CLIENT1,
      ParameterName           = SSLFED,
…
```

The command:

```
sslclientparameter
      ServerId          = ADMIN@WIN01,
      ParameterName     = *,
      ofmt              = std,
      Mode              = get
```

Displays:

```
_Correct Connected to server ADMIN@WIN01
_Correct sslclientparameter serverid=ADMIN@WIN01,parametername=*,ofmt=std,mode=get
_Correct Count = 15
SslClientParameter
      ParameterName           = CLIENT1,
      ServerId                = ADMIN@WIN01,
      Status                  = E,
      StoreLocation           = SYSTEM_STORE_LOCAL_MACHINE,
      StoreName               = My,
      Subject                =
      "CN=Test cli1,OU=tlabs01,O=tlabs,L=Paris,S=Paris,C=FR",
      Issuer                  = ,
      Protocol                = 1,
      CipherSuites            = TLS_RSA_WITH_3DES_EDE_CBC_SHA:,
      TraceLevel              = 2,
      SslBytesHeaderUsed      = false,
      Mode                    = replace
SslClientParameter
      ParameterName           = SSLFED,
      ServerId                = ADMIN@WIN01,
      Status                  = E,
      StoreLocation           = SYSTEM_STORE_LOCAL_MACHINE,
      StoreName               = My,
      Subject                 = "Test rsa1024 windows cli",
      Protocol                = 1,
      CipherSuites            = TLS_RSA_WITH_3DES_EDE_CBC_SHA:,
      TraceLevel              = 2,
      SslBytesHeaderUsed      = false,
      Mode                    = replace
….
```

The command:

```
SslClientParameter
      ParameterName           = CLIENT1,
      ServerId                = ADMIN@WIN01,
      Subject                =
      "CN=My new cert,OU=tlabs01,O=tlabs,L=Paris,S=Paris,C=FR",
      Issuer                  = ,
      Protocol                = 1,
      CipherSuites            = TLS_RSA_WITH_DES_CBC_SHA,
      TraceLevel              = 2,
      SslBytesHeaderUsed      = false,
      Mode                    = replace
```

Displays:

```
_Correct Connected to server ADMIN@WIN01
_Correct sslclientparameter
parametername=CLIENT1,serverid=ADMIN@WIN01,status=E,storelocation=SYSTEM_STORE_LOCAL_MACHINE,
storename=My,subject="CN=My new cert,OU=tlabs01,O=tlabs,L=Paris,S=Paris,C=FR",
protocol=1,ciphersuites=TLS_RSA_WITH_DES_CBC_SHA,tracelevel=2,sslbytesheaderused=false,
mode=replace
_Correct ssl client parameter CLIENT1 replaced
```

SslServerParameter

Use « sslserverparameter » to manage SSL server parameter definitions. The access key is the parameter name.

Format

```
sslserverparameter
  ServerId=server-id,
  ParameterName=(name|wildcard),
  Mode=(get|list)
```

Mode Get and List - parameters

Long Name	Short Name	Description	Win	Unix	Data Type
ParameterName	SNAM	Name of this parameter definition or simple wildcard	8	8	S
ServerId	SRVID	Server Id	128	128	s
Ofmt	OFMT	Format of the information returned by cxcmd - std (standard) or xml	64	64	S
Mode	MODE	get or list	4	4	S

ParameterName, ServerId and Mode are required.

Examples

The command:

```
sslserverparameter
      ServerId          = ADMIN@WIN01,
      ParameterName     = *,
      ofmt              = std,
      Mode              = list
```

Displays:

```
_Correct Connected to server ADMIN@LIN01
_Correct sslserverparameter serverid=ADMIN@LIN01,parametername=*,ofmt=std,mode=list
_Correct Count = 2
SslServerParameterList
      ParameterName           = SERVER1,
      ParameterName           = SERVER2,
…
```

The command:

```
sslserverparameter
      ServerId          = ADMIN@LIN01,
      ParameterName     = *,
      ofmt              = std,
      Mode              = get
```

Displays:

```
_Correct Connected to server ADMIN@LIN01
_Correct sslserverparameter serverid=ADMIN@LIN01,parametername=*,ofmt=std,mode=get
_Correct Count = 2
SslServerParameter
      ParameterName           = SERVER1,
      ServerId                = ADMIN@LIN01,
      Status                  = H,
      CertificateId           = R1024XSR,
      VerificationOption      = 2,
      CipherList              = cipher_3des.txt,
      Tlsv1Enabled            = true,
      Sslv3Enabled            = true,
      Sslv2Enabled            = true,
      SslBytesHeaderUsed      = false,
      TcpipPort               = 6690,
      TcpipAddress            = ,
      CAList                  = STROOT,
      DHParamFile             = ,
      Mode                    = replace
SslServerParameter
      …
```

Certificate

Use « certificate » to display X509 certificates that can be accessed by Connect :Express. The command admits different parameters, depending on the OS type of the server (Unix or Windows).
On Unix, the certificates are imported into a database, private to Connect :Express, by using the $sterm tool.
On Windows, the certificates are imported into the certificate stores of the Windows system by using the « Microsoft Management Console ».

Unix Format

```
sslserverparameter
  ServerId=server-id,
  CertificateId=(name|wildcard|*),
  CertificateType=(P|C|*),
  Subject=(name|wildcard),
  Issuer=(name|wildcard),
  Ofmt=(std|xml),
  Mode=(get|list)
```

Windows Format

```
sslserverparameter
  ServerId=server-id,
  StoreLocation=(SYSTEM_STORE_LOCAL_MACHINE| SYSTEM_STORE_CURRENT_USER|
                 SYSTEM_STORE_SERVICES),
  StoreName=(My|Root|CA|Trust),
  Subject=(name|expression),
  Issuer=(name|expression),
  FullDn=(true|false),
  ReverseFlag=(true|false),
  Ofmt=(std|xml),
  Mode=(get|list)
```

Long Name	**Short Name**	**Description**	**Win**	**Unix**	**Data Type**
ServerId	SRVID	Server Id	128	128	S
CertificateId	CERT	Certificate Id or simple pattern (for example “CACERT1” or “*CACERT*” or “*”) This parameter is mandatory (Unix).		8	S
CertificateType	CTTP	Type of certificate (P:Personal,C:Certificate authority). Optional.		1	
StoreLocation	STOL	Store location: SYSTEM_STORE_LOCAL_MACHINE, SYSTEM_STORE_CURRENT_USER or	64		S

		SYSTEM_STORE_SERVICES This parameter is mandatory (Windows).			
StoreName	STNM	Store name (My, Root, CA or Trust) This parameter is mandatory (Windows).	128		S
Subject	SUBJ	Certificate subject distinguished name. Optional. Lists on Windows: A string that must be found in the subject DN (for example “My Cert”. Lists on Unix: A simple pattern that must be found in the subject DN (for example “*My Cert*”	256	256	S
Issuer	ISSU	Certificate issuer distinguished name. Optional. Lists on Windows: A string that must be found in the issuer DN (for example “CA Cert”. Lists on Unix: A simple pattern that must be found in the issuer DN (for example “*CA Cert*”	256	256	S
FullDn	FUDN	For lists: indicates if the complete distinguished names are displayed instead of value of common names. This is the default.	5		B
ReverseFlag	REVE	For lists: indicates if, when complete DNs are returned, the components of the DN are aranged starting with the common name. This is the default.	5		B
Ofmt	OFMT	Format d’affichage de la commande cxcmd – std (standard) ou xml	64	64	S
Mode	MODE	get ou list	7	7	S

Certificate list

For a Unix server, the following command :

```
certificate
      ServerId          = ADMIN@LIN01,
      certificateId     = *,
      ofmt              = std,
      Mode              = list
```

Displays :

```
_Correct certificate serverid=ADMIN@LIN01,certificateid=*,ofmt=std,mode=list
_Correct Connected to server ADMIN@LIN01
_Correct Count = 19
CertificateList
Certificate
      CertificateId           = CA21FED,
      Type                    = C,
      Version                 = "3 (0x2)",
      SignatureAlgorithm      = sha1WithRSAEncryption,
      Subject                 =
      "C=FR, ST=Paris, L=Paris, O=tlabs, OU=tlabs02, CN=CA2-1 rsa1024 cert",
      Issuer                  =
      "C=FR, ST=Paris, L=Paris, O=tlabs, OU=tlabs02, CN=CA2 rsa1024 cert",
      SerialNumber            = "305 (0x131)",
      NotBefore               = "2007/09/28 14:18:58",
      NotAfter                = "2017/09/25 14:18:58",
      Mode                    = list
Certificate
      CertificateId           = MSOFT2,
      Type                    = C,
      Version                 = "3 (0x2)",
      SignatureAlgorithm      = sha1WithRSAEncryption,
      Subject                 =
      "DC=com, DC=microsoft, CN=Microsoft Root Certificate Authority",
      Issuer                  =
      "DC=com, DC=microsoft, CN=Microsoft Root Certificate Authority",
      SerialNumber            = 79:ad:16:a1:4a:a0:a5:ad:,
      NotBefore               = "2001/05/10 00:19:22",
      NotAfter                = "2021/05/10 00:28:13",
      Mode                    = list
```

For a Windows server, the following command :

```
certificate
      ServerId          = ADMIN@WIN01,
      StoreLocation     = SYSTEM_STORE_LOCAL_MACHINE,
      StoreName         = Root,
      ofmt              = std,
      Mode              = list
```

Displays :

```
_Correct certificate serverid=ADMIN@WIN01,storelocation=SYSTEM_STORE_LOCAL_MACHINE,
storename=Root,ofmt=std,mode=list
_Correct Connected to server ADMIN@WIN01
_Correct Count = 122
CertificateList
Certificate
      Version                 = 3,
      SignatureAlgorithm      = sha1RSA,
      Subject                 = "C=FR, S=Paris, L=Paris, O=tlabs, OU=tlabs01, CN=CATEST002",
      Issuer                  = "C=FR, S=Paris, L=Paris, O=tlabs, OU=tlabs01, CN=CATEST002",
      SerialNumber            = 00e0d5aaaf3e683f18,
      NotBefore               = "2009/10/05 17:54:43",
      NotAfter                = "2019/10/03 17:54:43",
      Mode                    = list
Certificate
      Version                 = 3,
      SignatureAlgorithm      = sha1RSA,
      Subject                 =
      "DC=com, DC=microsoft, CN=Microsoft Root Certificate Authority",
      Issuer                  =
      "DC=com, DC=microsoft, CN=Microsoft Root Certificate Authority",
      SerialNumber            = 79ad16a14aa0a5ad4c7358f407132e65,
      NotBefore               = "2001/05/10 00:19:22",
      NotAfter                = "2021/05/10 00:28:13",
      Mode                    = list
….
```

Note:
It is possible to limit the lists, by indicating character strings to search for in the subject and issuer DNs.
For example :

```
Certificate
      ServerId          = ADMIN@WIN01,
      StoreLocation     = SYSTEM_STORE_LOCAL_MACHINE,
      StoreName         = Root,
      Subject           = "Thawte",
      ofmt              = std,
      Mode              = list
```

This command will return the list of all certificates containing the string « Thawte ».

With a Unix server, use « *Thawte *» instead of « Thawte ».

Detail of a certificate

For a Unix server, the following command :

```
certificate
      ServerId          = ADMIN@LIN01,
      certificateId     = CACERT01,
      ofmt              = std,
      Mode              = list
```

Displays :

```
_Correct certificate serverid=ADMIN@LIN01,certificateid=CACERT01,ofmt=std,mode=get
_Correct Connected to server ADMIN@LIN01
_Correct Count = 1
Certificate
      CertificateId           = CACERT01,
      PEM                     =
-----BEGIN CERTIFICATE-----
MIIDBzCCAnCgAwIBAgIJAKE8xcxHPXamMA0GCSqGSIb3DQEBBQUAMGExCzAJBgNV
BAYTAkZSMQ4wDAYDVQQIEwVQYXJpczEOMAwGA1UEBxMFUGFyaXMxETAPBgNVBAoT
CFN0ZXJsaW5nMQ0wCwYDVQQLEwRMYWJzMRAwDgYDVQQDEwdDQSB0ZXN0MB4XDTA2
MDMwNzE1MzcxNloXDTE2MDMwNDE1MzcxNlowYTELMAkGA1UEBhMCRlIxDjAMBgNV
BAgTBVBhcmlzMQ4wDAYDVQQHEwVQYXJpczERMA8GA1UEChMIU3RlcmxpbmcxDTAL
BgNVBAsTBExhYnMxEDAOBgNVBAMTB0NBIHRlc3QwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBANEHiRTbyoCnHjzYURdhfjzcne9Lhy1/Voe49ePupFklKG527tPk
IUs2zkQ+KVkraky/IqTjlV2W5JrgbfGKeMTv5Tgwaq2oImxyZyDRQgan3Xi+/YN5
xrc97NYZLalvPHlsysYQUXY0siyKxYTvIwfmmrncTOVzcQVbmZdlQjGjAgMBAAGj
gcYwgcMwHQYDVR0OBBYEFKsKre9+pniL+IguTGta/zlnpsvzMIGTBgNVHSMEgYsw
gYiAFKsKre9+pniL+IguTGta/zlnpsvzoWWkYzBhMQswCQYDVQQGEwJGUjEOMAwG
A1UECBMFUGFyaXMxDjAMBgNVBAcTBVBhcmlzMREwDwYDVQQKEwhTdGVybGluZzEN
MAsGA1UECxMETGFiczEQMA4GA1UEAxMHQ0EgdGVzdIIJAKE8xcxHPXamMAwGA1Ud
EwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEAcTC1xydgC7E5XzBKq/MNfWZce09+
L6D8CbWGcBw9+7VzfKuqZsrL/E5j6sJDRNg40sQ4enNUXS8DKvnleDicj/Akm4uO
5Wdj03BTs+EX7h6RKN3vZ3phGufjfW3seCP5wYgSZ+3BJj3iCURDA3mFs8keHN1v
FJoNB+9mzAxsv00=
-----END CERTIFICATE-----
,
      Description             =
-----BEGIN DESCRIPTION-----
[
[
  Version: V3
  Subject: CN=CA test, OU=Labs, O=Sterling, L=Paris, ST=Paris, C=FR
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  Sun RSA public key, 1024 bits
  modulus: 1467854613624531473524330627743440752563074195510011664996608496822
4662551052154811558345661524760379004413624105493838003393502830401395774338701
7716392061144428590804479889085235402451680690592512571325786312883094419094739
7348760092980245169160545686955997803314849686324508791001042016319412101980567
76099
  public exponent: 65537
  Validity: [From: Tue Mar 07 16:37:16 CET 2006,
               To: Fri Mar 04 16:37:16 CET 2016]
  Issuer: CN=CA test, OU=Labs, O=Sterling, L=Paris, ST=Paris, C=FR
  SerialNumber: [    a13cc5cc 473d76a6]

Certificate Extensions: 3
[1]: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: AB 0A AD EF 7E A6 78 8B   F8 88 2E 4C 6B 5A FF 39  ......x....LkZ.9
0010: 67 A6 CB F3                                        g...
]
]
[2]: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: AB 0A AD EF 7E A6 78 8B   F8 88 2E 4C 6B 5A FF 39  ......x....LkZ.9
0010: 67 A6 CB F3                                        g...
]

[CN=CA test, OU=Labs, O=Sterling, L=Paris, ST=Paris, C=FR]
SerialNumber: [    a13cc5cc 473d76a6]
]

[3]: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]

]
  Algorithm: [SHA1withRSA]
  Signature:
0000: 71 30 B5 C7 27 60 0B B1   39 5F 30 4A AB F3 0D 7D  q0..'`..9_0J....
0010: 66 5C 7B 4F 7E 2F A0 FC   09 B5 86 70 1C 3D FB B5  f\.O./.....p.=..
0020: 73 7C AB AA 66 CA CB FC   4E 63 EA C2 43 44 D8 38  s...f...Nc..CD.8
0030: D2 C4 38 7A 73 54 5D 2F   03 2A F9 E5 78 38 9C 8F  ..8zsT]/.*..x8..
0040: F0 24 9B 8B 8E E5 67 63   D3 70 53 B3 E1 17 EE 1E  .$....gc.pS.....
0050: 91 28 DD EF 67 7A 61 1A   E7 E3 7D 6D EC 78 23 F9  .(..gza....m.x#.
0060: C1 88 12 67 ED C1 26 3D   E2 09 44 43 03 79 85 B3  ...g..&=..DC.y..
0070: C9 1E 1C DD 6F 14 9A 0D   07 EF 66 CC 0C 6C BF 4D  ....o.....f..l.M

]
-----END DESCRIPTION-----
,
      Mode                    = get
```

For a Windows server, the following command:

```
certificate
      ServerId          = ADMIN@WIN01,
      StoreLocation     = SYSTEM_STORE_LOCAL_MACHINE,
      StoreName         = Root,
      Subject           = "CN=Microsoft Root Certificate Authority",
      ofmt              = std,
      Mode              = get
```

Displays the detail of the certificate whose common name is : `Microsoft Root Certificate Authority`.

Appendix

The tables below indicate, depending on the operating system, the availability and the size of the different data fields of the Connect:Express servers.
The column “Key” contains the keywords used to identify the data in the statistics records.

Symbolic Partner Data

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=PesitE, O=Oftp, 3=Etebac3, F=ftp Unix: 0: PeSIT, 1: FTP, 2: Etebac3 (See ProtocolVersion in session table to get the version level of PeSIT, D or 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

Symbolic File Data

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
GDGN	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

Session Table Data

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

Presentation Table Data

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

Request Submission Parameters

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	
FJAI	FileJAI	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	RacfGroup	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	RemotelDisposition1	3	AdHoc remote allocation disposition (SHR, NEW, OLD) (MVS dcb)			3
RDS2	RemotelDisposition2	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		

Journal Data

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	
FJAI	FileJAI	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 Pi99)		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
PRIO	Priority	1	Transfer priority, 0 = Urgent, 1 = Normal , 2 = slow	1	1	1
PRID	ProcessId	12	Identifiation of the process that executed the request (PID, APMEFF ..)		12	5
PROT	Protocol	1	Transfer protocol, D=PesitD, E=PesitE, O=Oftp, 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	Indentification 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	1	1	

			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.			
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	

Active Transfer Data

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			
FJAI	FileJAI	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	Identifiation of the process that executed the request (PID, APMEFF ..)			12
PROT	Protocol	1	Transfer protocol, D=PesitD, E=PesitE, O=Oftp, 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	Indentification 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

Monitor’s Configuration Data

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	
JAIA	JAIAddress	15	JAI listening address	x		15
JAIP	JAIPort	5	JAI listening port	X	5	5
JAIV	JAIVersion	3	JAI version Number	X		
AUXX	APLinexx	100	Asset protection line numer xx (n lines)			
APPV	AppcVersion	3	Appc JAI 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 JAI 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 or 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		

SSL client parameters

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	

SSL server parameters

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	