Connect:Express® Windows

Programming Guide

Version 3.0.6.002

Sterling Commerce
An IBM Company

Connect:Express Windows
Programming Guide

Version 3.0.6.002
First Edition

INDEX

I Overview

1.1 General

The purpose of this document is to describe the application programming interface of **Connect:Express Windows.**
It explains the methods used to enable an application to communicate, via a Client/Server dialogue, with one or several Connect:Express transfer monitors.

The API functions are presented in two distinct function groups:

- *Administrative functions*, accessible locally using the standard product, or remotely, using the Activity Manager option.
- *Implementation and transfer activity functions*, accessible locally or remotely using the standard product.

The API functions update directly the initialization file specified using the file sharing system or, establish a client/server connection, via TCP/IP or Named pipe, to a Connect:Express monitor and ask the monitor to do the updates himself.

The setup of a Client/Server dialogue with a monitor is subject to certain authorization checks at API level and at monitor level.

1.2 Asset protection key

Connect:Express is supplied with a license.key file containing an authorization number (asset protection key) which allows the transfer monitor and the API to carry out the necessary checks for them to be used, depending on the licence acquired.

The checks made by the API concern:

- its location: the standard setting is for administration functions to be accessible locally , and for implementation and activity functions to be accessible locally and remotely
- the number of monitors connected at any time: the standard setting is for one connected monitor only at any given moment via the same API procedure.

The checks made by the monitor concern:

- the possibility to receive remote client connections
- the number of simultaneous remote client connections
- the API type used on the client station: Standard, Client/Server option or Activity Manager option

The license.key file must be placed in the API's execution directory.
Depending on this code either one or several monitors can be connected at the same time, and access to administration functions is either allowed or not allowed.

1.3 Initialization file

In order to use the API, an initialization file is required.

As standard, the API looks for initialization parameters in the **tomnt.ini** file in its execution directory. It is therefore necessary to pick up part of the **tomnt.ini** file when installing a remote client, with the monitor and the API using the same code.
For the Activity Manager and Client/Server options, the search is made in the **iutom.ini** file, the options and the monitor use different codes.
For a description of the information in the API initialization file please see the ***Connect:Express Windows Installation and Utilities***.

1.3 Network interfaces

The API interfaces with the following networks:

- TCP/IP
- Named Pipe

The network used should be activated in the configuration of the monitor to which you wish to connect, and the relevant network information (port number, name of named pipe, etc.) must be identical.

The monitor to which the connection is made via the API manages a timeout period (during which there is no traffic) that can be configured at monitor level. If the timeout period is exceeded, the connection is interrupted by the monitor.

1.4 Installation

The API is supplied on the CD-ROM of the Connect:Express Windows, and contains the following files:

- Apicxv3.dll: the dynamic library
- Apicxv3.lib: the import library
- Apicxv3d.h: the structure declarations and the constants for ‘C’ programming language
- Apicxv3p.h: function prototypes for ‘C’ programming language
- Apicxv3.txt: Visual Basic user types declarations and functions prototypes
- Tomreq.c: example of API implementation in ‘C’ programming language

To install the API, copy these files manually in a working directory. Don’t forget the initialization file described above.

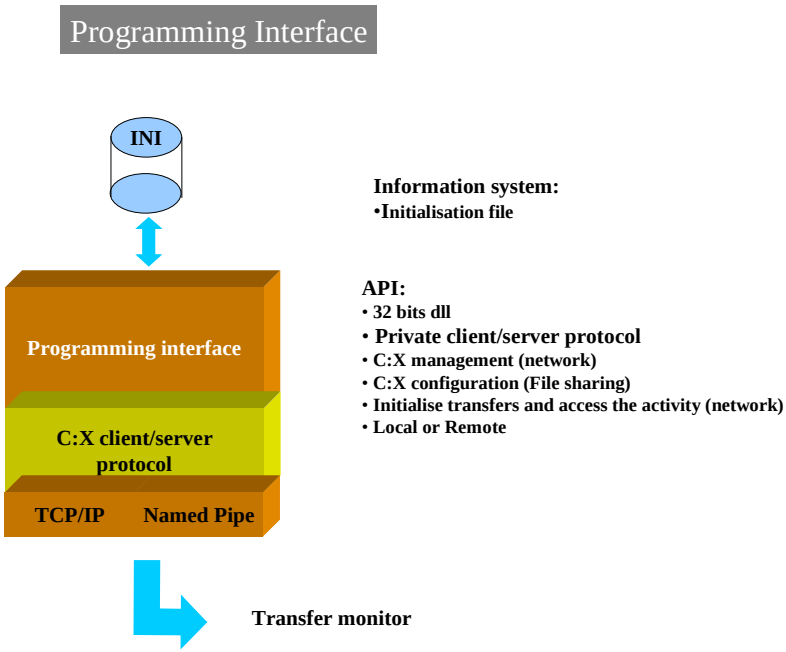
II Connect:Express API

2.1- Architecture

The API enables a Windows application to communicate with the Connect:Express transfer monitor and to access to its information system via a network connection or via the network's file-sharing system.

The TCP/IP or Named Pipe network connection and the establishment of a Client/Server dialogue allow users to do the following:

- gain update access to directories and tables
- initialize transfers
- access the transfer activity
- consult messages and the journal
- have access to stored notifications
- consult (connection) or update (file sharing) the monitor's settings



Besides the authorization code, an application can only use the API after the user CLIENT has been identified. A check is made of his access rights as defined in the Connect:Express monitor Client directory.

2.2 Integration in an application

2.2.1 C/C++ application

The API can be integrated in an application in two ways:

- ***STATICALLY*:** the application must include the 'APICXV3.LIB' import library when the links are made.
- ***DYNAMICALLY:*** the application should load the 'APICXV3.DLL' library dynamically and retrieve the addresses of the functions in order to implement them.

The call convention for API functions is of type: ***_stdcall***

The 'APICXV3D.H' file contains definitions, in 'C' programming language, of the structures and constants to be entered when calling the functions of this API.

The 'APICXV3P.H' file contains the prototypes, in C and C++ programming languages, of the functions for this API.

In a C++ application, the last header file must be used as following:

```
extern ″C″
        {
        #include ″apicxv3p.h″
        }
```

2.2.1 In a Visual Basic application

The file 'APICXV3.TXT' provides the user type declarations of the structures to be used when calling the API functions and, it contains also the function prototypes.
This file may be used by the Microsoft API Viewer utility to retrieve the declarations and function prototypes and to copy them in the Visual Basic application.

III Functions description

When the API functions are called, the applications indicate a structure address.

This structure, which is specific to each function, should be initialized as follows, and prior to every call:

- Fill up the structure with spaces along its entire length
- Enter information in the necessary fields using coherent values (use the constants that are specified in the 'APICXV3D.H' file).

The characters tables must be filled up with spaces, except when the table is specified as being a character string (null terminated), in which case it is terminated by binary zeros.

During the function execution, the API updates this structure as following:

- The execution return codes from the monitor
- The data concerned by the function

And return a direct return code.
This direct return code states if the function was successfully executed or not.
The return codes of the structure (ApiRc, SysRc, TcpRc or TomRc) gives more information on the error retrieved when the function fails.

A maximum period of silence between two calls to API functions and during connection to a monitor must be maintained so that the Connect:Express monitor can check that the communication has closed.

This period of time is stated in the Start settings of the monitor that is connected.

The API functions are presented in three distinct function groups:

- *Services functions*, to initialize and terminate the API (ApiCxInit & ApiCxTerm)
- *The Network functions,* using the client/server connection to access the information system of a monitor (ApiCxNet…)
- *The File Sharing functions,* that directly access to the initialization file (ApiCxShr…)

3.1 Services functions

3.1.1 API initialization (ApiCxInit)

Description
This function should be the first to be called by the application using the API and should be called only once. It enables the API to reserve the resources it needs to execute and to check the asset protection key included in the license.key file.

Input
The application indicates :

- *IniFName* : full name of the initialization file in which the API must look for the following information:
  - ✓ The indicator for implementation of the TCP/IP protocol
  - ✓ The value for the timeout period for connections

Output
The function's direct return code indicates the following:

- -1 : the address of the APICX_INIT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, and *TcpRc* codes are described in the '***Connect:Express Windows User's Guide'.***

If the function was executed correctly, the following fields are returned:

- AuthVer : this indicates the type of product authorized by this number.
- MaxCx : the maximum number of connections that can be made simultaneously to monitors (1 to 8)
- MaxMon : the maximum number of monitors specified in the resource bar of the graphics interface (1 to 255)
- FlagIp : O (Oui = Yes in French) or N (No) to indicate whether or not TCP/IP is implemented
- MonAuth : the list of C :X monitors to which the application may connect

3.1.2 API Termination (ApiCxTerm)

Description
This function should be the last to be called by the application using the API; it makes all other functions unusable unless the initialization function has been called in advance.
It allows the API to free the sources allocated for it to run once initialized.

Input
No settings.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICX_TERM structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, and *TcpRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.2 Client/Server functions

3.2.1 Access to a Connect:Express monitor

3.2.1.2 Connection to a monitor (ApiCxNetConn)

Description
This function allows the application to set up the Client/Server dialogue with a Connect:Express monitor on a type TCP/IP or Named Pipe connection.
To call this function the application must previously have successfully called the API initialization function.

Input
The application indicates the network information necessary to open the communication with the monitor:

- *Linktp* : link type (TCP/IP or Named Pipe)
- *IpAddr* : for TCP/IP – IP address of the system where the C :X monitor resides
- *IpName* : for TCP/IP – IP host name of the system where the C :X monitor resides
- *IpPort* : for TCP/IP – port number on which the C :X monitor is waiting for client calls
- *NpName* : For Named Pipe – Name of the pipe created by the Connect:Express monitor
- *CliName* : client name
- *CliPsw* : client password

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_CX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

If the function was executed correctly, the APICXN_CX structure is completed as follows:

- CxId : connection identifier. This value must be picked up by the application for this connection
- MonId : contains three descriptions identifying the type and version of the C :X monitor, the identifier of the client connection for the monitor, and the date and time of the connection.
- CliAuth : states the list of authorizations held by the connected client
- ApiType : this is the type of API implemented, which enables you to determine the accessible functions.
- MonType : this is the type of C :X monitor to which the application is connected.

3.2.1.2 Disconnection from a monitor (ApiCxNetDisc)

Description
This function allows the application to close a Client/Server dialogue that has previously been set up with a Connect:Express monitor on a type TCP/IP or Named Pipe connection.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : the identifier for the connection it wishes to close.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_DISC structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windowsbits User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.2 Partners directory access

3.2.2.1 List of Partners definitions (ApiCxNetListPart)

Description
This function is called to retrieve the list of partners defined in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of partners' symbolic names.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_PART structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_PART structure is filled as following :

- *NbNamesRet* : number of partner names stored in the list indicated upon entry
- *TotDef* : total number of names defined in the Connect:Express monitor partner directory

If the number of defined names is more than the number of names in memory, this indicates that the size of the memory location specified upon entry is not sufficient to allow the full list of partner names to be stored.

The list that is returned displays a series of symbolic partner names, each one made of 8 characters filled with blanks when necessary.

3.2.2.2 View a Partner definition (ApiCxNetViewPart)

Description
This function is called to retrieve the details of a partner definition defined in a Connect:Express monitor. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic partner name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ PART structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express for Win 32 User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_ PART structure is filled as following :

- *PartPsw* : partner password, made up of 8 upper case characters, and filled in with spaces
- *LocName* : local Connect:Express symbolic identifier name, made up of 8 upper case characters, and filled in with spaces
- *LocPsw* local Connect:Express identifier password, made up of 8 upper case characters, and filled in with spaces
- *LocTyp* : specify even if the local ID is static (APICX_LOCS) or dynamic (APICX_LOCD).
- *Comment* : description describing this partner definition, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *State* : status of the partner which may take on the value of 'APICX_STAE' for «Enable» or 'APICX_STAH' for «Disable»
- *PartType* : type of partner which may take on the value of 'APICX_PCE' for CONNECT :Express or 'APICX_POTH' for other
- *Restart* : this allows automatic restart with this partner : 'O' [Oui = French for Yes] for yes or 'N' for no
- *ProtType* : type of protocol used with this partner which may take on the value of 'APICX_PPD' for PeSIT version D, 'APICX_PPE' for PeSIT version E or 'APICX_PET3' for ETEBAC-3.
- *MaxSess* : maximum total number of simultaneous connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *MaxSessIn* : maximum total number of incoming connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *MaxSessOut* : maximum total number of outgoing connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *SessName* : name of the session table for the PeSIT protocol used for this partner, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *Linktp* : type of network used as default to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62'
- *IpAddr* : IP address of the partner in 'aaa.bbb.ccc.ddd' format with at least one zero character (\0) at the end.
- *IpName*: name of the TCP/IP 'host' for the partner, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *IpPort* : number of the TCP/IP port for the partner, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.

- *LuName* : name of the partner's LU, made up of 8 upper case characters, and filled in with spaces
- *ModName* : name of the partner's mode, made up of 8 upper case characters, and filled in with spaces
- *TpName* : name of the partner's transaction program, made up of 64 characters, and filled in with spaces
- *XLAddr* : local address specified when the partner is called, made up of a maximum of 15 numeric characters, with at least one zero character (\0) at the end.
- *XRAddr* : address of the partner made up of at least 15 numeric characters with at least one zero character (\0) at the end.
- *XPort* : number of the X.25 port to use to call this partner, made up of 2 numeric characters, with at least one zero character (\0) at the end.
- *XUdf* user data field to specify when calling this partner, made up of 8 upper case characters, and filled in with spaces
- *XFac* facilities field to specify when calling this partner, made up of 32 upper case characters, and filled in with spaces

3.2.2.2 View a Partner definition (Extended) (ApiCxNetViewPartEx)

Description
This function is called to retrieve the details of a partner definition defined in a Connect:Express monitor. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic partner name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ PART_EX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express for Win 32 User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_ PART structure is filled as following :

- *PartPsw* : partner password, made up of 8 upper case characters, and filled in with spaces
- *LocName* : local Connect:Express symbolic identifier name, made up of 8 upper case characters, and filled in with spaces
- *LocPsw* local Connect:Express identifier password, made up of 8 upper case characters, and filled in with spaces
- *LocTyp* : specify even if the local ID is static (APICX_LOCS) or dynamic (APICX_LOCD).
- *Comment* : description describing this partner definition, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *State* : status of the partner which may take on the value of 'APICX_STAE' for «Enable» or 'APICX_STAH' for «Disable»
- *PartType* : type of partner which may take on the value of 'APICX_PCE' for CONNECT :Express or 'APICX_POTH' for other
- *Restart* : this allows automatic restart with this partner : 'O' [Oui = French for Yes] for yes or 'N' for no
- *ProtType* : type of protocol used with this partner which may take on the value of 'APICX_PPD' for PeSIT version D, 'APICX_PPE' for PeSIT version E or 'APICX_PET3' for ETEBAC-3.
- *MaxSess* : maximum total number of simultaneous connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *MaxSessIn* : maximum total number of incoming connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *MaxSessOut* : maximum total number of outgoing connections with this partner, in 3 characters, which can take a minimum value of '000' and a maximum value of '128' or a space for 255
- *SessName* : name of the session table for the PeSIT protocol used for this partner, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *Linktp* : type of network used as default to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62'
- *IpAddr* : IP address of the partner in 'aaa.bbb.ccc.ddd' format with at least one zero character (\0) at the end.
- *IpName*: name of the TCP/IP 'host' for the partner, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *IpPort* : number of the TCP/IP port for the partner, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.

- *LuName* : name of the partner's LU, made up of 8 upper case characters, and filled in with spaces
- *ModName* : name of the partner's mode, made up of 8 upper case characters, and filled in with spaces
- *TpName* : name of the partner's transaction program, made up of 64 characters, and filled in with spaces
- *XLAddr* : local address specified when the partner is called, made up of a maximum of 15 numeric characters, with at least one zero character (\0) at the end.
- *XRAddr* : address of the partner made up of at least 15 numeric characters with at least one zero character (\0) at the end.
- *XPort* : number of the X.25 port to use to call this partner, made up of 2 numeric characters, with at least one zero character (\0) at the end.
- *XUdf* user data field to specify when calling this partner, made up of 8 upper case characters, and filled in with spaces
- *XFac* facilities field to specify when calling this partner, made up of 32 upper case characters, and filled in with spaces
- *SslUsed* : '1' indicates that SSL is used for transfers with this partner.
- *Sslparm* : Field of 8+1 characters indicating the symbolic name of an SSL client parameter definition. The name is terminated with a binary zero.
- *RemoteClientSubjectDn* : Field of 255+1 characters indicating authorization criteria for the subject DN of the certificate of the remote client (SSL/TCP transfer). Criteria are terminated with a binary zero.
- *RemoteClientRootDn* : Field of 255+1 characters indicating authorization criteria for the root DN of the certificate of the remote client (SSL/TCP transfer). Criteria are terminated with a binary zero.
- *RemoteServerSubjectDn* : Field of 255+1 characters indicating authorization criteria for the subject DN of the certificate of the remote server (SSL/TCP transfer). Criteria are terminated with a binary zero.
- *RemoteServerRootDn* : Field of 255+1 characters indicating authorization criteria for the root DN of the certificate of the remote server (SSL/TCP transfer). Criteria are terminated with a binary zero.

3.2.2.3 Add/Update a Partner definition (ApiCxNetUpdatePart)

Description
This function is called to add or update a partner definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic partner name
- *PartPsw* : partner password
- *LocName* : local Connect:Express symbolic identifier name
- *LocPsw* local Connect:Express identifier password
- *LocTyp* : specify even if the local ID is static (APICX_LOCS) or dynamic (APICX_LOCD).
- *Comment* : description describing this partner definition
- *State* : status of the partner which may take on the value of 'APICX_STAE' for «Enable» or 'APICX_STAH' for «Disable»
- *PartType* : type of partner which may take on the value of 'APICX_PCE' for CONNECT :Express or 'APICX_POTH' for other
- *Restart* : this allows automatic restart with this partner : 'O' [Oui = French for Yes] for yes or 'N' for no
- *ProtType* : type of protocol used with this partner which may take on the value of 'APICX_PPD' for PeSIT version D, 'APICX_PPE' for PeSIT version E or 'APICX_PET3' for ETEBAC-3.
- *MaxSess* : maximum total number of simultaneous connections with this partner
- *MaxSessIn* : maximum total number of incoming connections with this partner
- *MaxSessOut* : maximum total number of outgoing connections with this partner
- *SessName* : name of the session table for the PeSIT protocol used for this partner
- *Linktp* : type of network used as default to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62'
- *IpAddr* : IP address of the partner in 'aaa.bbb.ccc.ddd' format
- *IpName*: name of the TCP/IP 'host' for the partner
- *IpPort* : number of the TCP/IP port for the partner
- *LuName* : name of the partner's LU
- *ModName* : name of the partner's mode
- *TpName* : name of the partner's transaction program
- *XLAddr* : local address specified when the partner is called
- *XRAddr* : address of the
- *XPort* : number of the X.25 port to use to call this partner
- *XUdf* user data field to specify when calling this partner
- *XFac* facilities field to specify when calling this partner, made up of 32 upper case characters, and filled in with spaces

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ PART structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.2.3 Add/Update a Partner definition (Extended) (ApiCxNetUpdatePartEx)

Description
This function is called to add or update a partner definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic partner name
- *PartPsw* : partner password
- *LocName* : local Connect:Express symbolic identifier name
- *LocPsw* local Connect:Express identifier password
- *LocTyp* : specify even if the local ID is static (APICX_LOCS) or dynamic (APICX_LOCD).
- *Comment* : description describing this partner definition
- *State* : status of the partner which may take on the value of 'APICX_STAE' for «Enable» or 'APICX_STAH' for «Disable»
- *PartType* : type of partner which may take on the value of 'APICX_PCE' for CONNECT :Express or 'APICX_POTH' for other
- *Restart* : this allows automatic restart with this partner : 'O' [Oui = French for Yes] for yes or 'N' for no
- *ProtType* : type of protocol used with this partner which may take on the value of 'APICX_PPD' for PeSIT version D, 'APICX_PPE' for PeSIT version E or 'APICX_PET3' for ETEBAC-3.
- *MaxSess* : maximum total number of simultaneous connections with this partner
- *MaxSessIn* : maximum total number of incoming connections with this partner
- *MaxSessOut* : maximum total number of outgoing connections with this partner
- *SessName* : name of the session table for the PeSIT protocol used for this partner
- *Linktp* : type of network used as default to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62'
- *IpAddr* : IP address of the partner in 'aaa.bbb.ccc.ddd' format
- *IpName*: name of the TCP/IP 'host' for the partner
- *IpPort* : number of the TCP/IP port for the partner
- *LuName* : name of the partner's LU
- *ModName* : name of the partner's mode
- *TpName* : name of the partner's transaction program
- *XLAddr* : local address specified when the partner is called
- *XRAddr* : address of the
- *XPort* : number of the X.25 port to use to call this partner
- *XUdf* user data field to specify when calling this partner
- *XFac* facilities field to specify when calling this partner, made up of 32 upper case characters, and filled in with spaces
- *SslUsed* : '1' indicates that SSL is used for transfers with this partner.
- *Sslparm* : Field of 8+1 characters indicating the symbolic name of an SSL client parameter definition. The name must be terminated with a binary zero.
- *RemoteClientSubjectDn* : Optional field of 255+1 characters indicating authorization criteria for the subject DN of the remote client certificate (SSL/TCP transfers). If present, criteria must be terminated with a binary zero (else set the first byte to 0 or space).
- *RemoteClientRootDn* : Optional field of 255+1 characters indicating authorization criteria for the root DN of the remote client certificate (SSL/TCP transfers). If present, criteria must be terminated with a binary zero (else set the first byte to 0 or space).
- *RemoteServerSubjectDn* : Optional field of 255+1 characters indicating authorization criteria for the subject DN of the remote server certificate (SSL/TCP transfers). If present, criteria must be terminated with a binary zero (else set the first byte to 0 or space).

- *RemoteServerRootDn* : Optional field of 255+1 characters indicating authorization criteria for the root DN of the remote server certificate (SSL/TCP transfers). If present, criteria must be terminated with a binary zero (else set the first position to 0 or space).

Output

The function's direct return code indicates the following:

- -1 : the address of the APICXN_ PART_EX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.2.4 Delete a Partner definition (ApiCxNetDelPart)

Description
This function is called to delete a partner definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic partner name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ DEL_PART structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.3 Files directory access

3.2.3.1 List of Files definitions (ApiCxNetListFile)

Description
This function is called to retrieve the list of files defined in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of files' symbolic names.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_FILE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_FILE structure is filled as following :

- *NbNamesRet* : number of file names stored in the list indicated upon entry
- *TotDef* : total number of names defined in the Connect:Express monitor files directory

If the number of defined names is more than the number of names in memory, this indicates that the size of the memory location specified upon entry is not sufficient to allow the full list of files names to be stored.

The list that is returned displays a series of symbolic files names, each one made of 8 characters filled with blanks when necessary.

3.2.3.2 View a File definition (ApiCxNetViewFile)

Description
This function is called to retrieve the details of a file definition defined in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *FileName* : symbolic file name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_FILE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_FILE structure is filled as following :

- *Comment* : description describing the definition for this file, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *State* : status of the file which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable.
- *Definition :* definition rule for the file which may take on the values 'APICX_DEFF' for Fixed or 'APICX_DEFD' for Dynamic
- *Direction:* direction of transfer authorized for this file and it may take on the values 'APIV2_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Receive and Transmit
- *FileType* : type of file and may take on the value of 'TF' for Fixed Text 'TV' for Variable Text, 'BF' for Fixed Binary, or 'BI' for Undefined Binary.
- *ReceiveMode* : reception rule, which may take on the values 'APICX_RECVN' for New or 'APICX_RECVR' for Replace. To be compatible with the Transmit direction, this field takes on the value of 'APICX_SENDO' in order to Open an existing file
- *Origin* : name of the partner authorized to transmit this file, made up of 8 upper case characters, filled in with spaces
- *Destination :* name of the partner authorized to receive this file, made up of 8 upper case characters, filled in with spaces
- *PresName* : name of the presentation table, for the PeSIT or the ETEBAC-3 protocol, used for this file, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *PhysName* : physical name for this file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *RecordLg* : size of the records in this file, made up of maximum 5 characters, filled in with spaces

- *TrIex* : name of the exit for the start of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrTex* : name of the exit for the end of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReIex* : name of the exit for the start of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReTex* : name of the exit for the end of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrIcd* : name of the start transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrTcd* : name of the end transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReIcd* : name of the command to start reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReTcd* : name of the end reception command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *Errcd* : name of the transfer error command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *Notify* : this allows the transfer notification facility to be implemented for this file : 'O' [Oui = French for Yes] for yes or 'N' for no
- *CliName* : name of the client to be notified, made up of 8 upper case characters, filled in with spaces

3.2.3.3 View a File definition (Extended) (ApiCxNetViewFileEx)

Description
This function is called to retrieve the details of a file definition defined in a Connect:Express monitor.

This is an extension of the preceding ApiCxNetViewFile function, permitting to deal with the pi 37 (label) and pi 99 information. See the ***Exchanging pi 37 and pi 99 with PeSIT partners*** document. The used structure is APICXN_FILE_EX. Calls to this function can only be addressed to Connect:Express monitors in version V302 or later.

To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *FileName* : symbolic file name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_FILE_EX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_FILE_EX structure is filled as following :

- *Comment* : description describing the definition for this file, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *State* : status of the file which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable.
- *Definition :* definition rule for the file which may take on the values 'APICX_DEFF' for Fixed or 'APICX_DEFD' for Dynamic
- *Direction:* direction of transfer authorized for this file and it may take on the values 'APIV2_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Receive and Transmit
- *FileType* : type of file and may take on the value of 'TF' for Fixed Text 'TV' for Variable Text, 'BF' for Fixed Binary, or 'BI' for Undefined Binary.
- *ReceiveMode* : reception rule, which may take on the values 'APICX_RECVN' for New or 'APICX_RECVR' for Replace. To be compatible with the Transmit direction, this field takes on the value of 'APICX_SENDO' in order to Open an existing file
- *Origin* : name of the partner authorized to transmit this file, made up of 8 upper case characters, filled in with spaces
- *Destination :* name of the partner authorized to receive this file, made up of 8 upper case characters, filled in with spaces
- *PresName* : name of the presentation table, for the PeSIT or the ETEBAC-3 protocol, used for this file, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *PhysName* : physical name for this file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *RecordLg* : size of the records in this file, made up of maximum 5 characters, filled in with spaces

- *TrIex* : name of the exit for the start of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrTex* : name of the exit for the end of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReIex* : name of the exit for the start of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReTex* : name of the exit for the end of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrIcd* : name of the start transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *TrTcd* : name of the end transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReIcd* : name of the command to start reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *ReTcd* : name of the end reception command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *Errcd* : name of the transfer error command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *Notify* : this allows the transfer notification facility to be implemented for this file : 'O' [Oui = French for Yes] for yes or 'N' for no
- *CliName* : name of the client to be notified, made up of 8 upper case characters, filled in with spaces
- *Pi99LoadTypeT* : Value 'D'
- *Pi99OffsetT* : Pi 99 offset for transmission (3 numerical characters)
- *Pi99LengthT* : Pi 99 length for transmission (3 numerical characters)
- *Pi9ValueT* : Pi 99 value for transmission (254 characters) (padding space)
- *Pi99LoadTypeR* : Value 'D'
- *Pi99OffsetR* : Pi 99 offset for reception (3 numerical characters)
- *Pi99LengthR* : Pi 99 length for reception (3 numerical characters)
- *Pi9ValueR* : Pi 99 value for reception (254 characters) (padding space)
- *Label* : Pi 99 Value (80 characters) (padding space)
- *TypeOfNotification* : Type of HTTP Notification (used by the monitor only if the http notification component is installed). 1 character ('0' to '7'). '0': No http notification. '1': http notification at the beginning of the transfer. '2': http notification at the end of the transfer. '4': http notification if transfer error. The other possibilities are combinations with inclusive « OR » of these values. For example: '6' = '2' OR '4' for a notification at the end of the transfer or in case of transfer error.
- *Filler[255]* : RFU.

3.2.3.4 Add/Update a File definition (ApiCxNetUpdateFile)

Description
This function is called to add or update a file definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *FileName* : symbolic file name
  - *Comment* : description describing the definition for this file, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
  - *State* : status of the file which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable.
  - *Definition :* definition rule for the file which may take on the values 'APICX_DEFF' for Fixed or 'APICX_DEFD' for Dynamic
  - *Direction:* direction of transfer authorized for this file and it may take on the values 'APIV2_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Receive and Transmit
  - *FileType* : type of file and may take on the value of 'TF' for Fixed Text 'TV' for Variable Text, 'BF' for Fixed Binary, or 'BI' for Undefined Binary.
  - *ReceiveMode* : reception rule, which may take on the values 'APICX_RECVN' for New or 'APICX_RECVR' for Replace. To be compatible with the Transmit direction, this field takes on the value of 'APICX_SENDO' in order to Open an existing file
  - *Origin* : name of the partner authorized to transmit this file, made up of 8 upper case characters, filled in with spaces
  - *Destination :* name of the partner authorized to receive this file, made up of 8 upper case characters, filled in with spaces
  - *PresName* : name of the presentation table, for the PeSIT or the ETEBAC-3 protocol, used for this file, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
  - *PhysName* : physical name for this file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
  - *RecordLg* : size of the records in this file, made up of maximum 5 characters, filled in with spaces
  - *TrIex* : name of the exit for the start of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrTex* : name of the exit for the end of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReIex* : name of the exit for the start of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReTex* : name of the exit for the end of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrIcd* : name of the start transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrTcd* : name of the end transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReIcd* : name of the command to start reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReTcd* : name of the end reception command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *Errcd* : name of the transfer error command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *Notify* : this allows the transfer notification facility to be implemented for this file : 'O' [Oui = French for Yes] for yes or 'N' for no
  - *CliName* : name of the client to be notified, made up of 8 upper case characters, filled in with spaces

Output

The function's direct return code indicates the following:

- -1 : the address of the APICXN_FILE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.3.5 Add/Update a File definition (Extended) (ApiCxNetUpdateFileEx)

Description
This function is called to add or update a file definition in a Connect:Express monitor.

This is an extension of the preceding ApiCxNetUpdateFile function, permitting to deal with the pi 37 (label) and pi 99 information. See the ***Exchanging pi 37 and pi 99 with PeSIT partners*** document. The used structure is APICXN_FILE_EX. Calls to this function can only be addressed to Connect:Express monitors in version V302 or later.

To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *FileName* : symbolic file name
  - *Comment* : description describing the definition for this file, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
  - *State* : status of the file which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable.
  - *Definition :* definition rule for the file which may take on the values 'APICX_DEFF' for Fixed or 'APICX_DEFD' for Dynamic
  - *Direction:* direction of transfer authorized for this file and it may take on the values 'APIV2_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Receive and Transmit
  - *FileType* : type of file and may take on the value of 'TF' for Fixed Text 'TV' for Variable Text, 'BF' for Fixed Binary, or 'BI' for Undefined Binary.
  - *ReceiveMode* : reception rule, which may take on the values 'APICX_RECVN' for New or 'APICX_RECVR' for Replace. To be compatible with the Transmit direction, this field takes on the value of 'APICX_SENDO' in order to Open an existing file
  - *Origin* : name of the partner authorized to transmit this file, made up of 8 upper case characters, filled in with spaces
  - *Destination :* name of the partner authorized to receive this file, made up of 8 upper case characters, filled in with spaces
  - *PresName* : name of the presentation table, for the PeSIT or the ETEBAC-3 protocol, used for this file, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
  - *PhysName* : physical name for this file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
  - *RecordLg* : size of the records in this file, made up of maximum 5 characters, filled in with spaces
  - *TrIex* : name of the exit for the start of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrTex* : name of the exit for the end of transmission, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReIex* : name of the exit for the start of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReTex* : name of the exit for the end of reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrIcd* : name of the start transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *TrTcd* : name of the end transmission command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReIcd* : name of the command to start reception, made up of a maximum of 127 characters, with a zero character (\0) at the end.
  - *ReTcd* : name of the end reception command, made up of a maximum of 127 characters, with a zero character (\0) at the end.

- *Errcd* : name of the transfer error command, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *Notify* : this allows the transfer notification facility to be implemented for this file : 'O' [Oui = French for Yes] for yes or 'N' for no
- *CliName* : name of the client to be notified, made up of 8 upper case characters, filled in with spaces
- *Pi99LoadTypeT* : Value 'D'
- *Pi99OffsetT* : Pi 99 offset for transmission (3 numerical characters)
- *Pi99LengthT* : Pi 99 length for transmission (3 numerical characters)
- *Pi9ValueT* : Pi 99 value for transmission (254 characters) (padding space)
- *Pi99LoadTypeR* : Value 'D'
- *Pi99OffsetR* : Pi 99 offset for reception (3 numerical characters)
- *Pi99LengthR* : Pi 99 length for reception (3 numerical characters)
- *Pi9ValueR* : Pi 99 value for reception (254 characters) (padding space)
- *Label* : Pi 99 Value (80 characters) (padding space)
- *TypeOfNotification* : Type of HTTP Notification (used by the monitor only if the http notification component is installed). 1 character ('0' to '7'). '0': No http notification. '1': http notification at the beginning of the transfer. '2': http notification at the end of the transfer. '4': http notification if transfer error. The other possibilities are combinations with inclusive « OR » of these values. For example: '6' = '2' OR '4' for a notification at the end of the transfer or in case of transfer error.
- *Filler[255]* : RFU.

Output

The function's direct return code indicates the following:

- -1 : the address of the APICXN_FILE_EX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.3.6 Delete a File definition (ApiCxNetDelFile)

Description
This function is called to delete a file definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *FileName* : symbolic file name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ DEL_FILE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.4 Clients directory access

3.2.4.1 List of Clients definitions (ApiCxNetListClient)

Description
This function is called to retrieve the list of clients defined in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of clients' symbolic names.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_CLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_CLI structure is filled as following :

- *NbNamesRet* : number of client names stored in the list indicated upon entry
- *TotDef* : total number of names defined in the Connect:Express monitor clients directory

If the number of defined names is more than the number of names in memory, this indicates that the size of the memory location specified upon entry is not sufficient to allow the full list of clients names to be stored.

The list that is returned displays a series of symbolic clients names, each one made of 8 characters filled with blanks when necessary.

3.2.4.2 View a Client definition (ApiCxNetViewClient)

Description
This function is called to retrieve the details of a client definition defined in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic client name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_CLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_CLI structure is filled as following :

- *CliPsw* : client password, made up of 8 upper case characters, filled in with spaces
- *Comment* : description describing the definition for this client, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *State* : status of the client which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable
- *CliAuth :* list of client authorizations (see the structure APICXN_CX in the APICXV3D.H file)
- *FilesNb* : number of files authorized for transfer by this client, made up of 3 numeric characters from '000' to '100'.
- *Files* : list of symbolic names of files authorized to be transmitted by this client, made up of 8 upper case characters, and filled in with spaces
- *Notify* : this allows the client to receive transfer notifications: 'O' [Oui = French for Yes] for yes or 'N' for no
- *Linktp* : type of network used as default to communicate with this client which may take on the values 'APICX_TCPIP' or 'APICX_NP'
- *IpAddr* : IP address of the client in 'aaa.bbb.ccc.ddd' format with at least one zero character (\0) at the end.
- *IpName* : name of the TCP/IP 'host' for the client, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *IpPort* : number of the TCP/IP port for the client, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *NpName* : name of the Named Pipe for the client, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.2.4.3 Add/Update a Client definition (ApiCxNetUpdateClient)

Description
This function is called to add or update a client definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic client name
- *CliPsw* : client password
- *Comment* : description describing the definition for this client
- *State* : status of the client which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable
- *CliAuth :* list of client authorizations (see the structure APICXN_CX in the APICXV3D.H file)
- *FilesNb* : number of files authorized for transfer by this
- *Files* : list of symbolic names of files authorized to be transmitted by this client
- *Notify* : this allows the client to receive transfer notifications: 'O' [Oui = French for Yes] for yes or 'N' for no
- *Linktp* : type of network used as default to communicate with this client which may take on the values 'APICX_TCPIP' or 'APICX_NP'
- *IpAddr* : IP address of the client in 'aaa.bbb.ccc.ddd' format
- *IpName* : name of the TCP/IP 'host' for the client
- *IpPort* : number of the TCP/IP port for the client
- *NpName* : name of the Named Pipe for the client

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_CLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.4.4 Delete a Client definition (ApiCxNetDelClient)

Description
This function is called to delete a client definition in a Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic client name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ DEL_CLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.5 SSL Clients directory access

3.2.5.1 List of SSL Clients definitions (ApiCxNetListSslparmCli)

Description
This function is called to retrieve the list of SSL clients defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of SSL clients' symbolic names.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_SSLPARMCLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_SSLPARMCLI structure is filled as following :

- *NbNamesRet* : number of SSL client names stored in the list indicated upon entry
- *TotDef* : total number of names defined in the Connect:Express monitor clients directory

If the number of defined names is more than the number of names in memory, this indicates that the size of the memory location specified upon entry is not sufficient to allow the full list of clients names to be stored.

The list that is returned displays a series of symbolic SSL clients names, each one made of 8 characters filled with blanks when necessary.

3.2.5.2 View an SSL Client definition (ApiCxNetViewSslparmCli)

Description
This function is called to retrieve the details of an SSL client definition defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *Name* : symbolic SSL client name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_SSLPARMCLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_SSLPARMCLI structure is filled as following :

- *Enabled* : SSL client state ('1' : Enabled, '0' : Not enabled)
- *StoreProvider* : Certificate provider. Character string 'STORE_PROV_SYSTEM' terminated by a binary zero.
- *StoreLocation* : Certificate location. Null terminated character string having one of the following values : 'SYSTEM_STORE_LOCAL_MACHINE', 'SYSTEM_STORE_SERVICES' or 'SYSTEM_STORE_CURRENT_USER'.
- StoreName : Null terminated optional character string 'My', used in case of client authentication.
- *SubjectDn* : Null terminated optional character string containing the subject distinguished name of the client certificate used in case of client authentication.
- *IssuerDn* : Null terminated optional character string containing the issuer distinguished name of the client certificate used in case of client authentication.
- *Protocol* : SSL protocol version ('1' :TLSV1,'3' :SSLV3, '2' :SSLV2)
- *CipherSuites* : Cipher suites. Optional null terminated character string composed of a list of 2 digit numbers. This string indicates the cipher suites to use. The following number can be used :

```
'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,
'06'  SSL_CK_RC4_128_WITH_MD5,
'07'  SSL_CK_DES_64_CBC_WITH_MD5,
'08'  SSL_CK_RC2_128_CBC_WITH_MD5
Exemple : 000103\0 indicates the suites 00, 01 and 03.
```

- *Trace* : Trace level indicator ('0' : None, '1' : Partial, '2' : Complete)

3.2.5.3 View an SSL Client definition (ApiCxNetViewSslparmCliEx)

Description
This function is similar to the preceding one, but the following field has been added to the structure:

- *SslBytesHeader* : Indicates if unencrypted data are prefixed by a 2 bytes length header ('0' : No bytes length header, '1' : 2 bytes length header)

The structure is named APICXN_SSLPARMCLI_EX.

3.2.5.4 Add/Update an SSL Client definition (ApiCxNetUpdateSslparmCli)

Description
This function is called to add or update an SSL client definition in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *Name* : symbolic SSL client name
- *Enabled* : SSL client state (‘1’ : Enabled, ‘0’ : Not enabled)
- *StoreProvider* : Certificate provider. Character string ‘STORE_PROV_SYSTEM’ terminated by a binary zero.
- *StoreLocation* : Certificate location. Null terminated character string having one of the following values : ‘SYSTEM_STORE_LOCAL_MACHINE’, ‘SYSTEM_STORE_SERVICES’ or ‘SYSTEM_STORE_CURRENT_USER’.
- StoreName : Null terminated optional character string ‘My’, used in case of client authentication.
- *SubjectDn* : Null terminated optional character string containing the subject distinguished name of the client certificate used in case of client authentication.
- *IssuerDn* : Null terminated optional character string containing the issuer distinguished name of the client certificate used in case of client authentication.
- *Protocol* : SSL protocol version (‘1’ :TLSV1,’3’ :SSLV3, ‘2’ :SSLV2)
- *CipherSuites* : Cipher suites. Optional null terminated character string composed of a list of 2 digit numbers. This string indicates the cipher suites to use. The following number can be used :

```
'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,
'06'     SSL_CK_RC4_128_WITH_MD5,
'07'     SSL_CK_DES_64_CBC_WITH_MD5,
'08'     SSL_CK_RC2_128_CBC_WITH_MD5
Exemple : 000103\0 indicates the suites 00, 01 and 03.
```

- *Trace* : Trace level indicator (‘0’ : None, ‘1’ : Partial, ‘2’ : Complete)

3.2.5.5 Add/Update an SSL Client definition (ApiCxNetUpdateSslparmCliEx)

Description
This function is similar to the preceding one, but the following field has been added to the structure:

- *SslBytesHeader* : Indicates if unencrypted data are prefixed by a 2 bytes length header (‘0’ : No bytes length header, ‘1’ : 2 bytes length header)

The structure is named APICXN_SSLPARMCLI_EX.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_SSLPARMCLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.5.6 Delete an SSL Client definition (ApiCxNetDelSslparmCli)

Description
This function is called to delete an SSL client definition in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *Name* : symbolic SSL client name

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ DEL_SSLPARMCLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.6 Access to the monitor's activity

3.2.6.1 List of authorized names (ApiCxNetListAuth)

Description
This function is called by the application to:

- view the list of files authorized for transfer to a given client
- view the list of partners transmitters to a given file authorized to a given client,
- view the list of partners receivers to a given file authorized to a given client,

when the client is not authorized to consult the files directory.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListTyp :* type of list required and may be one of the following 'APICX_FILELIST', 'APICX_TRANLIST' or 'APICX_RECELIST'
- *CliName* : symbolic name of the client, made up of 8 upper case characters, and filled in with spaces
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_AUTH structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_AUTH structure is filled as following :

- *NamesNb* : number of names returned by the API, made up of 3 numeric characters from '000' to '500'.
- *Names* : list of symbolic names corresponding to the list type specified, made up of 8 upper case characters, and filled in with spaces

3.2.6.2 Details of an authorized file for a specified client (ApiCxNetViewAuth)

Description
This function is called by the application to view the details of a file authorized for transfer to a given client when the client is not authorized to consult the files directory.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the client, made up of 8 upper case characters, and filled in with spaces
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_DETAIL_AUTH structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_DETAIL_AUTH structure is filled as following :

- *Comment* : description describing the file definition, made up of a maximum of 80 upper case characters, with at least one zero character (\0) at the end.
- *State* : status of the file which may take on the value of 'APICX_STAE' for Enable or 'APICX_STAH' for Disable.
- *Definition :* definition rule for the file which may take on the value of 'APICX_DEFF' for Fixed or 'APICX_DEFD' for Dynamic
- *Direction:* direction of transfer authorized for this file and may take on the values 'APICX_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Receive and Transmit
- *Origin* : name of the partner authorized to transmit this file, made up of 8 upper case characters, and filled in with spaces
- *Destination :* name of the partner authorized to receive this file, made up of 8 upper case characters, and filled in with spaces
- *PhysName* : physical name of this file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.2.6.3 Definition type of a partner (ApiCxNetPartAuth)

Description
This function is called by the application to retrieve the local ID mode of a partner.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PartName* : symbolic name of the partner, made up of 8 upper case characters, and filled in with spaces

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PART_AUTH structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PART_AUTH structure is filled as following :

- *LocTyp* : indicates if the local ID is static (APICX_LOCS) or dynamic (APICX_LOCD).

3.2.6.4 Submit a transfer request (ApiCxNetSubmitReq)

Description
This function is called by the application to submit a transfer request to the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the client initiating the request, made up of 8 upper case characters, and filled in with spaces. The default value is the name of the connected client.
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces
- *PhysName* : physical name of the file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end. The physical name defined in the file directory is used as default
- *Direction:* direction of transfer authorized for the file and may take on the values 'APICX_DIRR' for Receive or 'APICX_DIRT' for Transmit, or a space to take on the default value. The default value is the direction that is defined in the directory
- *PartName* : symbolic name of the partner, made up of 8 upper case characters, and filled in with spaces Depending on the direction, the name of the transmitter or of the receiver defined in the files directory is used as the default value
- *Linktp* : type of network used to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62' or a space to take on the default value. The default value is the network type defined in the partners directory
- *Alias* : local name for a local dynamic ID
- *AliasPsw* : local password for a local dynamic ID
- *Priority* : priority of the transfer and may take on the values 'APICX_PRIL' for Low or 'APICX_PRIN' for Normal, or 'APICX_PRIU for High, or a space to take on the default value. The default value is Normal.
- *ReqType* : type of request and may take on the values 'APICX_RTYPD' for Inquiry 'APICX_RTYPN' for Normal, or 'APICX_RTYPA for Hold, or a space to take on the default value. The request type is Normal for a transmitted transfer, and Inquiry for a received transfer
- *CliIdent* : application identifier for this transfer, consisting of 16 freeform characters.
- *Notify* : this enables notification for this transfer to be implemented.
- *NotifyInfos* : structure that, in this version, contains only the name of the client to be notified, consisting of a maximum of 8 characters, filled in with spaces
- *AdHoc* : structure used within a free-service framework, containing the following :
  - ✓ the name of the user made up of 8 characters
  - ✓ the user's password made up of 8 characters
  - ✓ the physical name of the local file, made up of 44 characters
  - ✓ the physical name of the remote file, made up of 44 characters
- *Extend* : structure containing the extended information for the request and containing the following :
  - ✓ the origin of the file made up of 8 characters
  - ✓ the destination of the file made up of 8 characters
  - ✓ the transmitter of the file made up of 8 characters
  - ✓ the receiver of the file made up of 8 characters
  - ✓ the label of the file made up of 80 characters
- *TrfDate* : date the transfer is executed in YYYY/MM/DD format, or a space to take on the default value. The default value is today's date
- *TrfTime* : time the transfer is executed in HH :MM :SS format, or a space to take on the default value. The default is the current time
- *Etebac3* : structure containing the ETEBAC-3 card

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_SUB_REQ structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_SUB_REQ structure is filled as following :

- *ReqNb* : number of transfer requests submitted to the monitor, made up of 3 numeric characters from '000' to '100'.
- *ReqNumber* : list of transfer request number, made up of 12 upper case characters, and filled in with spaces

3.2.6.5 Submit a transfer request (Extended) (ApiCxNetSubmitReqEx)

Description
This function is called by the application to submit a transfer request to the Connect:Express monitor.

This is an extension of the preceding ApiCxNetSubmitReqEx function, permitting to deal with the pi 37 (label) and pi 99 information. See the ***Exchanging pi 37 and pi 99 with PeSIT partners*** document. The used structure is APICXN_SUB_REQ_EX. Calls to this function can only be addressed to Connect:Express monitors in version V302 or later.

To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the client initiating the request, made up of 8 upper case characters, and filled in with spaces. The default value is the name of the connected client.
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces
- *PhysName* : physical name of the file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end. The physical name defined in the file directory is used as default
- *Direction:* direction of transfer authorized for the file and may take on the values 'APICX_DIRR' for Receive or 'APICX_DIRT' for Transmit, or a space to take on the default value. The default value is the direction that is defined in the directory
- *PartName* : symbolic name of the partner, made up of 8 upper case characters, and filled in with spaces Depending on the direction, the name of the transmitter or of the receiver defined in the files directory is used as the default value
- *Linktp* : type of network used to communicate with this partner which may take on the values 'APICX_TCPIP' 'APICX_X25' or 'APICX_LU62' or a space to take on the default value. The default value is the network type defined in the partners directory
- *Alias* : local name for a local dynamic ID
- *AliasPsw* : local password for a local dynamic ID
- *Priority* : priority of the transfer and may take on the values 'APICX_PRIL' for Low or 'APICX_PRIN' for Normal, or 'APICX_PRIU for High, or a space to take on the default value. The default value is Normal.
- *ReqType* : type of request and may take on the values 'APICX_RTYPD' for Inquiry 'APICX_RTYPN' for Normal, or 'APICX_RTYPA for Hold, or a space to take on the default value. The request type is Normal for a transmitted transfer, and Inquiry for a received transfer
- *CliIdent* : application identifier for this transfer, consisting of 16 freeform characters.
- *Notify* : this enables notification for this transfer to be implemented.
- *NotifyInfos* : structure that, in this version, contains only the name of the client to be notified, consisting of a maximum of 8 characters, filled in with spaces
- *AdHoc* : structure used within a free-service framework, containing the following :
  - ✓ the name of the user made up of 8 characters
  - ✓ the user's password made up of 8 characters
  - ✓ the physical name of the local file, made up of 44 characters
  - ✓ the physical name of the remote file, made up of 44 characters
- *Extend* : structure containing the extended information for the request and containing the following :
  - ✓ the origin of the file made up of 8 characters
  - ✓ the destination of the file made up of 8 characters
  - ✓ the transmitter of the file made up of 8 characters
  - ✓ the receiver of the file made up of 8 characters
  - ✓ the label of the file made up of 80 characters
- *TrfDate* : date the transfer is executed in YYYY/MM/DD format, or a space to take on the default value. The default value is today's date

- *TrfTime* : time the transfer is executed in HH :MM :SS format, or a space to take on the default value. The default is the current time
- *Etebac3* : structure containing the ETEBAC-3 card
- *Pi99LoadTyp* : Value ‘D’
- *Pi99Offset* : Pi 99 offset (3 numerical characters)
- *Pi99Length* : Pi 99 length (3 numerical characters)
- *Pi9Value* : Pi 99 value (254 characters) (padding space)
- *TypeOfNotification* : Type of HTTP Notification (used by the monitor only if the http notification component is installed). 1 character (‘0’ to ‘7’). ‘0’: No http notification. ‘1’: http notification at the beginning of the transfer. ‘2’: http notification at the end of the transfer. ‘4’: http notification if transfer error. The other possibilities are combinations with inclusive « OR » of these values. For example: ‘6’ = ‘2’ OR ‘4’ for a notification at the end of the transfer or in case of transfer error.
- *Filler[255]* : RFU.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_SUB_REQ_EX structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_SUB_REQ_EX structure is filled as following :

- *ReqNb* : number of transfer requests submitted to the monitor, made up of 3 numeric characters from '000' to '100'.
- *ReqNumber* : list of transfer request number, made up of 12 upper case characters, and filled in with spaces

3.2.6.6 List of transfer requests (ApiCxNetListReq)

Description
This function is called by the application to retrieve the list of transfer requests for the Connect:Express monitor. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ReqUserType* type of the initiator of the request, and can take on the values 'APICX_USRTI' for Internal, 'APICX_USRTE' for external, or 'APICX_YSR' for both Internal and External.
- *CliName* : symbolic name of the client initiating the request, made up of 8 upper case characters, and filled in with spaces. An asterisk '*' can be used to select all clients.
- *CliIdent* : application identifier for this transfer, consisting of 16 freeform characters. An asterisk '*' can be used to select all identifiers.
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces. An asterisk '*' can be used to select all files.
- *Direction:* direction of transfer for the file and may take on the values 'APICX_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Transmit or Receive.
- *PartName* : symbolic name of the partner, made up of 8 upper case characters, and filled in with spaces An asterisk '*' can be used to select all partners.
- *ListAddr* : address of a memory location allocated by the application which will receive the list of requests responding to the selection criteria stated above
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_REQ structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_REQ structure is filled as following :

- *NbReqRet* : number of requests stored in the list indicated above
- *TotReq* total number of requests responding to the selection criteria

If the total number of requests is greater than the number of requests in memory, this indicates that the size of the memory location upon entry is not sufficient to allow the full list of requests to be stored.

The list that is returned is a sequence of APICX_DLIST_REQ structures which is made up, respectively, of the following :

- *ReqNumber* : number of the transfer request in 12 characters.
- *CliIdent* : application identifier for the transfer made up of 16 characters
- *CliName* : name of the client initiating the transfer request in 8 characters.
- *FileName* : symbolic name of the file in 8 characters.
- *Direction:* direction of the transfer and the values may be 'APICX_DIRT' or 'APICX_DIRR'
- *PartName* : symbolic name of the partner in 8 characters.
- *ReqStatus* status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place
- *ReqUserType* : type of the initiator of the request, which can take on the values 'APICX_USRTI' for Internal, 'APICX_USRTE' for external.

3.2.6.7 View a transfer request (ApiCxNetViewReq)

Description
This function is called by the application to retrieve the detailed information for a transfer request for the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:
- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ReqNumber* : number of the transfer request in 12 characters.

Output
The function's direct return code indicates the following:
- -1 : the address of the APICXN_REQ structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_REQ structure is filled as following :
- *CliIdent* : application identifier for this transfer, consisting of 16 freeform characters.
- *CliName* : symbolic name of the client initiating the request, made up of 8 characters
- *FileName* : symbolic name of the file in 8 characters.
- *Direction:* direction of transfer for the file and may take on the value of 'APICX_DIRR' for « Receive » or 'APICX_DIRT' for « Transmit»
- *PartName* : symbolic name of the partner in 8 characters.
- *PhysName* : physical name of the file, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.
- *ReqStatus* : status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place
- *ReqUserType* : type of the initiator of the request, and can take on the values 'APICX_USRTI' for Internal, 'APICX_USRTE' for external.
- *Trc* : Connect:Express code of the error
- *Prc* : protocol code of the error
- *Src* : siystem code of the error
- *Erc* : C_TREE code of the error
- *Nrc* : network code of the error
- *FileSize* : size of the file in bytes
- *TransSize* : size of the amount of the file transferred, in bytes
- *BegDate* : date that the transfer starts in YYYY/MM/DD format.
- *BegTime* : time that the transfer starts in HH :MM :SS format.

3.2.6.8 Action on a transfer request (ApiCxNetActionReq)

Description
This function is called by the application to react to a transfer request from the Connect:Express monitor. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ReqNumber* : number of the transfer request in 12 characters.
- *Action* : action to be taken on the request and the values may be 'APICX_ACTI' for Interrupt, 'APICX_ACTP' for Purge or 'APICX_ACTR' for Resume

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ACTION_REQ structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.7 Access to the monitor's messages

3.2.7.1 Liste of messages (ApiCxNetListLog)

Description
This function is called by the application to retrieve the list of messages for the Connect:Express monitor (32 bits only).
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *BegDate* : date that the message starts in YYYY/MM/DD format.
- *BegTime* time that the message starts in HH :MM :SS format.
- *EndDate* date that the messages end in YYYY/MM/DD format.
- *EndTime* : time that the messages end in HH :MM :SS format.
- *ListAddr* : address of a memory location allocated by the application which will receive the messages from the monitor that are included in the period indicated above
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_LOG structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_LOG structure is filled as following :

- *NbMsgRet* : number of messages stored in the list indicated upon entry
- *TotMsg* : total number of messages responding to the selection criteria

If the total number of messages is greater than the number of messages in memory, this indicates that the size of the memory location upon entry is not sufficient to allow the full list of messages to be stored.

The list that is returned displays a series of messages of maximum 255 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.2.8 Access to the journal of transfers

3.2.8.1 List of journal records (ApiCxNetListJnl)

Description
This function is called by the application to retrieve the list of journal records from the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the client initiating the request, made up of 8 upper case characters, and filled in with spaces. An asterisk '*' can be used to select all clients.
- *FileName* : symbolic name of the file, made up of 8 upper case characters, and filled in with spaces. An asterisk '*' can be used to select all files.
- *PartName* : symbolic name of the partner, made up of 8 upper case characters, and filled in with spaces An asterisk '*' can be used to select all partners.
- *Direction:* direction of transfer for the file and may take on the values 'APICX_DIRR' for Receive or 'APICX_DIRT' for Transmit, or 'APICX_DIR' for Transmit or Receive.
- *BegDate* : date that the journal records begin in YYYY/MM/DD format.
- *BegTime* : time that the journal records begin in HH :MM :SS format.
- *EndDate* : date that the journal records end in YYYY/MM/DD format.
- *EndTime* : time that the journal records end in HH :MM :SS format.
- *ListAddr* : address of a memory location allocated by the application which will receive the list of journal records responding to the selection criteria stated above
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_JNL structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_JNL structure is filled as following :

- *NbJnlRet* : number of journal records stored in the list indicated upon entry
- *TotJnl* : total number of journal records responding to the selection criteria

If the total number of journal records is greater than the number of journal records in memory, this indicates that the size of the memory location stated upon entry is not sufficient to allow the full list of journal records to be stored.

The list that is returned is a sequence of APICXN_DLIST_JNL structures which is made up, respectively, of the following :

- *JnlDate* : date that the journal record was registered in YYYY/MM/DD format.
- *JnlTime* : time that the journal record was registered in HH :MM : SS format.
- *ReqNumber* : number of the transfer request in 12 characters.
- *CliIdent* : application identifier for the transfer made up of 16 characters.
- *CliName* : name of the client initiating the transfer request in 8 characters.
- *FileName* : symbolic name of the file in 8 characters.
- *Direction:* direction of the transfer and the values may be 'APICX_DIRT' or 'APICX_DIRR'
- *PartName* : symbolic name of the partner in 8 characters.
- *ReqStatus* : status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place.

3.2.8.2 View a journal record (ApiCxNetViewJnl)

Description
This function is called by the application to retrieve the detailed information for a Connect:Express monitor journal record.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ReqNumber* : number of the transfer request .

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_JNL structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_JNL structure is filled as following :

- *JnlDate* : date that the journal record was registered in YYYY/MM/DD format.
- *JnlTime* time that the journal record was registered in HH :MM :SS format.
- *ExtReqNb* : number of the transfer request at the partner end in 8 characters.
- *CliIdent* : application identifier for this transfer.
- *TrfId :* PeSIT identifier for the transfer.
- *CliName* : symbolic name of the client initiating the request.
- *Trc* : C :X code of the error
- *Prc* : protocol code of the error
- *Src* : system code of the error
- *Erc* : C_TREE code of the error
- *Nrc* : network code of the error
- *TcpipRc* : TCP/IP code of the error
- *AppcPrc* : primary LU6.2 code of the error
- *AppcSrc* : secondary LU6.2 code of the error
- *X25rc* : X25 code of the error
- *X25cause* : CAUSE X25 code of the error
- *X25diag* : DIAGNOSTIC X25 code of the error
- *ReqState* it may be 'O' [Oui = French for Yes] if the request is purged by the monitor or 'N' if not
- *ReqStatus* : status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place.
- *ReqUserType* : type of the initiator of the request, and can take on the values 'APICX_USRTI' for Internal, 'APICX_USRTE' for external.
- *ReqType* : type of request and may take on the values 'APICX_RTYPD' for Inquiry 'APICX_RTYPN' for Normal, or 'APICX_RTYPA for Hold.

- *Direction:* direction of transfer authorized for the file and may take on the value of ‘APICX_DIRR’ for Receive or ‘APICX_DIRT’ for Transmit
- *Priority* : priority of the transfer and may take on the values ‘APICX_PRIL’ for Low or ‘APICX_PRIN’ for Normal, or ‘APICX_PRIU for High.
- *Linktp* : type of network used to communicate with this partner which may take on the values ‘APICX_TCPIP’ ‘APICX_X25’ or ‘APICX_LU62’
- *FileName :* symbolic name of the file
- *PhysName* : physical name of the file
- *PhysOname* : original physical name of the file
- *RemPhysName* : physical name of the file at the partner end
- *Label :* file label
- *CreDateTime* : indicated by the API, this is the time and date that the file was created in YYYMMDDHHMMSS format.
- *MajDateTime* : indicated by the API, this is the time and date that the file was updated in YYYMMDDHHMMSS format.
- *Origin* : indicated by the API, this is the origin of the file
- *Destination :* indicated by the API, this is the destination of the file
- *Transmitter* : indicated by the API, this is the transmitter of the file
- *Receiver* : indicated by the API, this is the receiver of the file
- *FileOrg* : indicated by the API, this is the way the file is organised and may take on the value of ‘APIV2_ORGI’ for Indexed or ‘APIC2_ORGS’ for Sequential, or ‘APIV2_ORGR’ for Relative
- *FileType* : indicated by the API, this is the type of file which may take on the value of ‘TF’ for Fixed Text ‘TV’ for Variable Text, ‘BF’ for Fixed Binary, or ‘BI’ for Undefined Binary.
- *Definition :* indicated by the API, this is the definition rule for the file, which may take on the value of ‘APIV2_DEFF’ for Fixed or ‘APIV2_DEFD’ for Dynamic.
- *Compress* : indicated by the API, this is the type of compression requested and may be the following : ‘APIV2_NOCOMP’ for No Compression, ‘APIV2_HCOMP’ for Horizontal compression, ‘APIV2_VCOMP’ for Vertical compression, or ‘APIV2_MCOMP’ for Mixed.
- *RealComp* : indicated by the API, this is the type of compression carried out and may be the following : ‘APIV2_NOCOMP’ for No Compression, ‘APIV2_HCOMP’ for Horizontal compression, ‘APIV2_VCOMP’ for Vertical compression, or ‘APIV2_MCOMP’ for Mixed.
- *DataType* : indicated by the API, this is the type of data contained in the file and may take on the values ‘APIV2_DATA’ for ASCII or ‘APIV2_DATE’ for EBCDIC, or ‘APIV2_DATB’ for Binary
- *RecordLg* : indicated by the API, this is the size of the file records in bytes
- *FileSize :* indicated by the API, this is the size of the file in bytes
- *MsgSize* : indicated by the API, this is the size of the network messages, in bytes
- *LocName* : indicated by the API, this is the local C :X symbolic name
- *PartName* : indicated by the API, this is the symbolic name of the partner
- *CliIdent :* indicated by the API, this is the client identifier for this transfer, consisting of 16 freeform characters.
- *Direction:* indicated by the API, this is the direction of transfer for the file and may take on the value of ‘APIV2_DIRR’ for Receive or ‘APIV2_DIRT’ for Transmit
- *PartName* : indicated by the API, this is the symbolic name of the file in 8 characters.
- *PartType* : indicated by the API, this is the type of partner and may take on the values ‘APIV2_PCE’ for CONNECT :Express or ‘APIV2_POTH’ for Other.
- *ProtType* : indicated by the API, this is the type of protocol used with this partner which may take on the values ‘APIV2_PPD’ or ‘APIV2_PPE’
- *CrcOpt* : indicated by the API, it may be ‘O’ [Oui = French for Yes] if the CRC check mechanism has been used during the transfer or ‘N’ if not
- *Sync* : indicated by the API, this is the synchronisation interval in Kilobytes
- *Wind:* indicated by the API, this is the synchronisation window

- *TcpipPort :* indicated by the API, this is the partner's TCP/IP port number
- *TcpipAddr* : indicated by the API, this is the partner's IP address in 'aaa.bbb.ccc.ddd' format
- *TcpipName* : indicated by the API, this is the partner's TCP/IP 'host'
- *LuName :* indicated by the API, this is the name of the partner's LU
- *ModName* : indicated by the API, this is the name of the partner's mode
- *TpName :* indicated by the API, this is the name of the partner's transaction program
- *XRAddr :* indicated by the API, this is the partner's X.25 address
- *XLAddr* : indicated by the API, this is the local X.25 address specified when the partner makes the call
- *XnPort :* indicated by the API, this is the number of the X.25 port used for this partner
- *Xudata :* indicated by the API, this is the user data field specified when the partner makes the call
- *XFac* : indicated by the API, this is the facilities field specified when the partner makes the call
- *BegDate* : indicated by the API, this is the date that the transfer starts in YYYY/MM/DD format.
- *BegTime* : indicated by the API, this is the time that the transfer starts in HH :MM :SS format.
- *EndDate* : indicated by the API, this is the date that the transfer ends in YYYY/MM/DD format.
- *EndTime* : indicated by the API, this is the time that the transfer ends in HH :MM :SS format.
- *RestartCnt* : indicated by the API, this is the number of restarts made for this transfer.
- *OpenAct :* indicated by the API, this is the action carried out when the transferred file is opened, and may take on the following values : 'APIV2_RECVN' for New file created, 'APIV2_RECVR' for existing file Replaced, or 'APIV2_SENDO' for existing file can be opened in 'read only' mode.
- *Resync* : indicated by the API, this is the number of resynchronisations made for this transfer.
- *TotByte* : indicated by the API, this is the total number of bytes for this transfer.
- *TotRec* : indicated by the API, this is the number of records transferred.
- *TotIO :* indicated by the API, this is the number network entries/exits made for this transfer.
- *ScheDate* : indicated by the API, this is the date that the requested transfer starts in YYYY/MM/DD format.
- *ScheTime* : indicated by the API, this is the time that the requested transfer starts in HH :MM :SS format.

3.2.9 Access to the notification of transfers

3.2.9.1 List of notifications (ApiCxNetListNot)

Description
This function is called by the application to retrieve the list of notifications from the Connect:Express monitor. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the client initiating the request, made up of 8 upper case characters, and filled in with spaces. An asterisk '*' can be used to select all clients.
- *ListAddr* : address of a memory location allocated by the application which will receive the list of notifications responding to the selection criteria stated above
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_NOT structure is filled as following :

- *NbNotRet* : number of notifications stored in the list indicated upon entry
- *TotNot* : total number of notifications responding to the selection criteria

If the total number of notifications is greater than the number of notifications in memory, this indicates that the size of the memory location stated upon entry is not sufficient to allow the full list of notifications to be stored.

The list that is returned is a sequence of APICXN_DLIST_NOT structures which is made up, respectively, of the following :

- *NotDate* : date of notification in YYYY/MM/DD format.
- *NotTime* : time of notification in HH/MM/SS format.
- *ReqNumber* : number of the transfer request in 12 characters.
- *CliIdent :* application identifier for the transfer made up of 16 characters.
- *CliName :* name of the notified client in 8 characters.
- *FileName* : symbolic name of the file in 8 characters.
- *ReqStatus* : status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place.
- *PhysName :* physical name of the file.
- *ReadInd:* indicator stating if the notification was already read (O= Yes) or not (N= No)

3.2.9.2 View a notification (ApiCxNetViewNot)

Description
This function is called by the application to retrieve the detailed information for a Connect:Express monitor notification.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the notified client.
- *ReqNumber* : number of the transfer request .

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_NOT structure is filled as following :

- *NotDate* : date of notification in YYYY/MM/DD format.
- *NotTime* : time of notification in HH :MM :SS format.
- *CliIdent* : application identifier for the transfer, consisting of 16 characters.
- *ReqStatus* : status of the transfer and the values may be 'APICX_REQSE' for Transfer Ended, 'APICX_REQSS' for Selection Error, 'APICX_REQSI' for Interrupted, 'APICX_REQSC' for Transfer currently taking place, 'APICX_REQST' for Selected, 'APICX_REQSW' for awaiting selection of transfer or 'APICX_REQSR' for automatic restart currently taking place
- *Trc* : C :X code of the error
- *Prc* : protocol code of the error
- *Src* : system code of the error
- *Erc* : C_TREE code of the error
- *Nrc* : network code of the error
- *Direction*: direction of the transfer and the values may be 'APICX_DIRT' or 'APICX_DIRR'
- *FileName* : symbolic name of the file in 8 characters.
- *PhysName :* physical name of the file
- *RemPhysName* : physical name of the file at the partner end
- *Label :* file label
- *Origin* : origin of the file
- *Destination :* destination of the file

3.2.9.3 Action on a notification (ApiCxNetActionNot)

Description
This function is called by the application to do an action on a Connect:Express monitor notification.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *CliName* : symbolic name of the notified client.
- *ReqNumber* : number of the transfer request .
- *Action* : action to be taken on the notification and the values may be 'APICX_ACTP' for Purge, or 'APICX_ACTR' for Route
- *NewClient* : symbolic name of the client to whom the notification is to be routed.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_ACTION_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10 Monitor tables access

3.2.10.1 List of PeSIT presentation tables (ApiCxNetListPPres)

Description
This function is called by the application to retrieve the list of PeSIT presentation tables from the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application which will receive the list of tables
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_PPRES structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_PPRES structure is filled as following :

- *NbNamesRet* : number of table names stored in the list indicated upon entry
- *TotDef* : total number of table names defined

If the total number of names is greater than the number of names in memory, this indicates that the size of the memory location stated upon entry is not sufficient to allow the full list of table names to be stored.

The list that is returned displays a series of names of maximum 50 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.2.10.2 View a PeSIT presentation table (ApiCxNetViewPPres)

Description
This function is called by the application to retrieve the detailed information for a Connect:Express monitor PeSIT presentation table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PPRES structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PPRES structure is filled as following :

- *CompMode* : type of compression which may be the following : 'APICX_NOCOMP' for No Compression, 'APICX_HCOMP' for Horizontal compression, 'APICX_VCOMP' for Vertical compression, or 'APICX_MCOMP' for Mixed.
- *ConcMode* : can be 'O' [Oui = French for Yes] to implement concatenation of protocol messages in the network messages, or 'N' if not.
- *MultMode* : can be 'O' [Oui = French for Yes] to implement concatenation of protocol messages in the network messages, or 'N' if not.
- *SegmMode* : can be 'O' [Oui = French for Yes] to locate file records in several messages for the protocol, or 'N' if not.
- *TransMode* : can be 'O' [Oui = French for Yes] to implement ASCII / EBCDIC translation, or 'N' for no translation.
- *AtoEName* : name of the ASCII - > EBCDIC translation table
- *EtoAName* : name of the EBCDIC - > ASCII translation table

3.2.10.3 Add/Update a PeSIT presentation table (ApiCxNetUpdatePPres)

Description
This function is called by the application to add or update a Connect:Express monitor PeSIT presentation table. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *CompMode* : type of compression which may be the following : 'APICX_NOCOMP' for No Compression, 'APICX_HCOMP' for Horizontal compression, 'APICX_VCOMP' for Vertical compression, or 'APICX_MCOMP' for Mixed.
- *ConcMode* : can be 'O' [Oui = French for Yes] to implement concatenation of protocol messages in the network messages, or 'N' if not.
- *MultMode* : can be 'O' [Oui = French for Yes] to implement concatenation of protocol messages in the network messages, or 'N' if not.
- *SegmMode* : can be 'O' [Oui = French for Yes] to locate file records in several messages for the protocol, or 'N' if not.
- *TransMode* : can be 'O' [Oui = French for Yes] to implement ASCII / EBCDIC translation, or 'N' for no translation.
- *AtoEName* : name of the ASCII - > EBCDIC translation table
- *EtoAName* : name of the EBCDIC - > ASCII translation table

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PPRES structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10.4 Delete a PeSIT presentation table (ApiCxNetDelPPres)

Description
This function is called by the application to delete a Connect:Express monitor PeSIT presentation table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_DEL_PPRES structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10.5 List of PeSIT Session tables (ApiCxNetListPSess)

Description
This function is called by the application to retrieve the list of PeSIT session tables from the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application which will receive the list of tables
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_PSESS structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_PSESS structure is filled as following :

- *NbNamesRet* : number of table names stored in the list indicated upon entry
- *TotDef* : total number of table names defined

If the total number of names is greater than the number of names in memory, this indicates that the size of the memory location stated upon entry is not sufficient to allow the full list of table names to be stored.

The list that is returned displays a series of names of maximum 50 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.2.10.6 View a PeSIT session table (ApiCxNetViewPSess)

Description
This function is called by the application to retrieve the detailed information for a Connect:Express monitor PeSIT session table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *SessName* : name of the session table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PSESS structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PSESS structure is filled as following :

- *MsgSz* : maximum size of the network messages, in bytes
- *SyncInt* : size of the synchronisation interval in Kilobytes
- *Direction:* direction allowed for transfers that will take place during the session. May take the following values : 'APICX_DIRR' for Receive only, 'APICX_DIRT' pour Transmission only or 'APICX_DIR' for Receive and Transmission
- *CrcMode* : can be 'O' [Oui = French for Yes] if the CRC check mechanism is to be implemented, or 'N' if not
- *ResyncCnt* : maximum number of resynchronisations

3.2.10.7 Add/Update a PeSIT session table (ApiCxNetUpdatePPres)

Description
This function is called by the application to add or update a Connect:Express monitor PeSIT session table. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *SessName* : name of the session table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *MsgSz* : maximum size of the network messages, in bytes
- *SyncInt* : size of the synchronisation interval in Kilobytes
- *Direction:* direction allowed for transfers that will take place during the session. May take the following values : 'APICX_DIRR' for Receive only, 'APICX_DIRT' pour Transmission only or 'APICX_DIR' for Receive and Transmission
- *CrcMode* : can be 'O' [Oui = French for Yes] if the CRC check mechanism is to be implemented, or 'N' if not
- *ResyncCnt* : maximum number of resynchronisations

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PSESS structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10.8 Delete a PeSIT session table (ApiCxNetDelPSess)

Description
This function is called by the application to delete a Connect:Express monitor PeSIT session table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *SessName* : name of the session table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_DEL_PSESS structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10.9 List of ETEBAC-3 presentation tables (ApiCxNetListPEtb3)

Description
This function is called by the application to retrieve the list of ETEBAC-3 presentation tables from the Connect:Express monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application which will receive the list of tables
- *ListSize* : size in bytes of the memory location

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_PETB3 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_PETB3 structure is filled as following :

- *NbNamesRet* : number of table names stored in the list indicated upon entry
- *TotDef* : total number of table names defined

If the total number of names is greater than the number of names in memory, this indicates that the size of the memory location stated upon entry is not sufficient to allow the full list of table names to be stored.

The list that is returned displays a series of names of maximum 50 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.2.10.10 View an ETEBAC-3 presentation table (ApiCxNetViewPEtb3)

Description
This function is called by the application to retrieve the detailed information for a Connect:Express monitor ETEBAC-3 presentation table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PETB3 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PETB3 structure is filled as following :

- *TransMode* : can be 'O' [Oui = French for Yes] to implement ASCII / EBCDIC translation, or 'N' for no translation.
- *AtoEName* : name of the ASCII - > EBCDIC translation table
- *EtoAName* : name of the EBCDIC - > ASCII translation table

3.2.10.11 Add/Update an ETEBAC-3 presentation table (ApiCxNetUpdatePEtb3)

Description
This function is called by the application to add or update a Connect:Express monitor ETEBAC-3 presentation table.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.
- *TransMode* : can be 'O' [Oui = French for Yes] to implement ASCII / EBCDIC translation, or 'N' for no translation.
- *AtoEName* : name of the ASCII - > EBCDIC translation table
- *EtoAName* : name of the EBCDIC - > ASCII translation table

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PETB3 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.10.12 Delete an ETEBAC-3 presentation table (ApiCxNetDelPEtb3)

Description
This function is called by the application to delete a Connect:Express monitor ETEBAC-3 presentation table. To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *PresName* : name of the presentation table, made up of a maximum of 50 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_DEL_PETB3 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.2.11 Monitor parameters access

3.2.11.1 View startup parameters (ApiCxNetViewPrmStr)

Description
This function is called by the application to view the Connect:Express monitor startup parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_STR structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_STR structure is filled as following :

- *RestartCnt* : maximum number of automatic restarts the monitor may effect on a transfer and the values may be from '00' to '99'.
- *WaitInt* : number of seconds to wait before an automatic restart is effected, and the values may be from '000' to '300'.
- *NetworkTimer* : number of seconds for the timeout period on the network during a transfer, and the values may be from '000' to '300'.
- *ClientTimer* : number of seconds for the timeout period on the network during a Client/Server connection, and the values may be from '000' to '300'.
- *StartType* : type of start for the monitor and may take on the values 'APICX_STARTC' for Hot [Chaud in French] or 'APICX_STARTF' for Cold [Froid in French].

3.2.11.2 View service parameters (ApiCxNetViewPrmSrv)

Description
This function is called by the application to view the Connect:Express monitor service parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_SRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_SRV structure is filled as following :

- *SrvState* : can be 'O' [Oui = French for Yes] if the monitor is installed as Windows NT or 'N' if not
- *Services* : list of dependent services

The list that is returned displays a series of service names of maximum 127 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.2.11.3 View files parameters (ApiCxNetViewPrmFil)

Description
This function is called by the application to view the Connect:Express monitor files parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:
- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:
- -1 : the address of the APICXN_PRM_FIL structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_FIL structure is filled as following :
- *InitJnl* : takes on the value of 'O' [Oui = French for yes] if the journal file is to be reinitialized when the monitor is started, or 'N' if not
- *JnlSize* : maximum number of records that the journal file should contain after reinitialization
- *InitLog* : takes on the value of 'O' [Oui = French for yes] if the messages file is to be reinitialized when the monitor is started, or 'N' if not
- *LogSize* : maximum number of records that the messages file should contain after reinitialization
- *Language* : left blank – not used in V3.01. The language used is the one selected in the Windows settings or English if it is not supported by Connect:Express.
- *Trace* : takes on the value of 'O' [Oui = French for yes] if internal trace is to be activated, or 'N' if not
- *InitNot* : takes on the value of 'O' [Oui = French for yes] if the notifications file is to be reinitialized when the monitor is started, or 'N' if not
- *NotSize* : maximum number of records that the notifications file should contain after reinitialization

3.2.11.4 View authorization parameters (ApiCxNetViewPrmAut)

Description
This function is called by the application to view the Connect:Express monitor authorization parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.
Note :
There software authorization parameters set in the ALIAS and NUMERO keywords of the initialization file are valid only for old authorization keys and are maintained for compatibility. The new asset protection key is delivered by Sterling Commerce as a file named license.key (See next paragraph).

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_AUT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_AUT structure is filled as following :

- *Alias (Identity code)* : alias that makes up a part of the authorization code for the Connect:Express product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *Number* : number that makes up a part of the authorization code for the C :X product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.

3.2.11.5 View Asset Protection Key elements (ApiCxNetListLicense)

Description
This function is called to retrieve the list of the Asset protection key elements as validated by the monitor from the authorization file license.key delivered by Sterling Commerce and located in the monitor's installation directory. The application gives as input parameter the address of a previously allocated buffer, with sufficient size to receive the list of the Asset protection key elements. A 4 Kbytes buffer is sufficient.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with the monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of AP key elements.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_LICENSE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

The list that is returned displays in the ListAddr buffer a series of lines having the following format: line000=élément$_0$,line001=élément$_1$, ….,line00n=élément$_n$

3.2.11.6 View notification parameters (ApiCxNetViewPrmNot)

Description
This function is called by the application to view the Connect:Express monitor notification parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_NOT structure is filled as following :

- *Notify* : takes on the value of 'O' [Oui = French for yes] if global notification is active, or 'N' if not
- *CliName* : symbolic name of the client to be notified, made up of 8 upper case characters, and filled in with spaces

3.2.11.7 View TCP/IP parameters (ApiCxNetViewPrmIp)

Description
This function is called by the application to view the Connect:Express monitor TCP/IP parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_IP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_IP structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the TCP/IP is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's IP address is to be effected, or 'N' if not
- *TrfPort* : number of the IP port receiving incoming calls for transfers, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *CliPort* : number of the IP port receiving incoming calls for Client connections, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL TCP/IP giving a Windows Sockest V1.1 to use, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.2.11.8 View SNA LU6.2 parameters (ApiCxNetViewPrmSna)

Description
This function is called by the application to view the Connect:Express monitor SNA LU6.2 parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_SNA structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_SNA structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the LU6.2 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's LU name is to be effected, or 'N' if not
- *LuName* : name of the local LU for receiving incoming calls, made up of 8 upper case characters, and filled in with spaces
- *AppcDllNm* : name of the SNA SERVER DLL APPC to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *CsvDllNm* : name of the SNA SERVER DLL CSV to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

3.2.11.9 View the X.25 parameters (ApiCxNetViewPrmX25)

Description
This function is called by the application to view the Connect:Express monitor X.25 parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:
- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:
- -1 : the address of the APICXN_PRM_X25 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_X25 structure is filled as following :
- *State* : takes on the value of 'O' [Oui = French for yes] if the X.25 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's address is to be effected, or 'N' if not
- *LocAddr* : local sub-address for receiving incoming calls, made up of a maximum of 15 characters, with at least one zero character (\0) at the end.
- *PortNb* : number of the port for receiving incoming calls, made up of a maximum of 2 characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL EX25 EICON to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

3.2.11.10 View Named Pipe parameters (ApiCxNetViewPrmNp)

Description
This function is called by the application to view the Connect:Express monitor named pipe parameters.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_PRM_NP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_PRM_NP structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the Named Pipe network is operational, or 'N' if not
- *NpName* : name of the Named Pipe for the client connections, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.2.11.11 List of SSL Servers definitions (ApiCxNetListSslparmSrv)

Description
This function is called to retrieve the list of SSL servers defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *ListAddr* : address of a memory location allocated by the application, which will receive the list of SSL servers' symbolic names.
- *ListSize* : size in bytes of the memory location indicated

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LIST_SSLPARMSRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_LIST_SSLPARMSRV structure is filled as following :

- *NbNamesRet* : number of SSL server names stored in the list indicated upon entry
- *TotDef* : total number of names defined in the Connect:Express

If the number of defined names is more than the number of names in memory, this indicates that the size of the memory location specified upon entry is not sufficient to allow the full list of SSL server names to be stored.

The list that is returned displays a series of symbolic SSL servers names, each one made of 8 characters filled with blanks when necessary.

3.2.11.12 View an SSL Server definition (ApiCxNetViewSslparmSrv)

Description
This function is called to retrieve the details of an SSL server definition defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:
- *CxId* : client/server connection ID
- *Version* : Connect:Express API version
- *Name* : symbolic SSL server name

Output
The function's direct return code indicates the following:
- -1 : the address of the APICXN_SSLPARMSRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXN_SSLPARMSRV structure is filled as following :
- *Enabled* : SSL server state ('1' : Enabled, '0' : Not enabled)
- *StoreProvider* : Certificate provider. Character string 'STORE_PROV_SYSTEM' terminated by a binary zero.
- *StoreLocation* : Certificate location. Null terminated character string having one of the following values : 'SYSTEM_STORE_LOCAL_MACHINE', 'SYSTEM_STORE_SERVICES' or 'SYSTEM_STORE_CURRENT_USER'.
- StoreName : Null terminated optional character string 'My'.
- *SubjectDn* : Null terminated optional character string containing the subject distinguished name of the server certificate.
- *IssuerDn* : Null terminated optional character string containing the issuer distinguished name of the server certificate.
- *Protocol* : SSL protocol version ('1' :TLSV1,'3' :SSLV3, '2' :SSLV2)
- *CipherSuites* : Cipher suites. Optional null terminated character string composed of a list of 2 digit numbers. This string indicates the cipher suites to use. The following number can be used :

```
'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,
'06'  SSL_CK_RC4_128_WITH_MD5,
'07'  SSL_CK_DES_64_CBC_WITH_MD5,
'08'  SSL_CK_RC2_128_CBC_WITH_MD5
Exemple : 000103\0 indicates the suites 00, 01 and 03.
```

- *Trace* : Trace level indicator ('0' : None, '1' : Partial, '2' : Complete)
- *ServerPort* : TCP/IP port of the server (Maximum 5 digits terminated by a binary zero)

3.2.11.13 View an SSL Server definition (ApiCxNetViewSslparmSrvEx)

Description
This function is similar to the preceding one, but the following field has been added to the structure:
- *SslBytesHeader* : Indicates if unencrypted data are prefixed by a 2 bytes length header ('0' : No bytes length header, '1' : 2 bytes length header)

The structure is named APICXN_SSLPARMSRV_EX.

3.3 File sharing access functions

3.3.1 Monitor startup parameters

3.3.1.1 View startup parameters (ApiCxShrViewPrmStr)

Description
This function is called by the application to view the Connect:Express monitor startup parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_STR structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_STR structure is filled as following :

- *RestartCnt* : maximum number of automatic restarts the monitor may effect on a transfer and the values may be from '00' to '99'.
- *WaitInt* : number of seconds to wait before an automatic restart is effected, and the values may be from '000' to '300'.
- *NetworkTimer* : number of seconds for the timeout period on the network during a transfer, and the values may be from '000' to '300'.
- *ClientTimer* : number of seconds for the timeout period on the network during a Client/Server connection, and the values may be from '000' to '300'.
- *StartType* : type of start for the monitor and may take on the values 'APICX_STARTC' for Hot [Chaud in French] or 'APICX_STARTF' for Cold [Froid in French].

3.3.1.2 Update startup parameters (ApiCxShrUpdatePrmStr)

Description
This function is called by the application to update the Connect:Express monitor startup parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *RestartCnt* : maximum number of automatic restarts the monitor may effect on a transfer and the values may be from '00' to '99'.
- *WaitInt* : number of seconds to wait before an automatic restart is effected, and the values may be from '000' to '300'.
- *NetworkTimer* : number of seconds for the timeout period on the network during a transfer, and the values may be from '000' to '300'.
- *ClientTimer* : number of seconds for the timeout period on the network during a Client/Server connection, and the values may be from '000' to '300'.
- *StartType* : type of start for the monitor and may take on the values 'APICX_STARTC' for Hot [Chaud in French] or 'APICX_STARTF' for Cold [Froid in French].

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_STR structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***

3.3.2 Monitor service parameters

3.3.2.1 View service parameters (ApiCxShrViewPrmSrv)

Description
This function is called by the application to view the Connect:Express monitor service parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_SRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_SRV structure is filled as following :

- *SrvState* : can be 'O' [Oui = French for Yes] if the monitor is installed as Windows NT or 'N' if not
- *Services* : list of dependent services

The list that is returned displays a series of service names of maximum 127 characters, each separated by a zero character (\0) and with the last one followed by at least two zero characters (\0\0).

3.3.2.2 Update service parameters

The service parameters may be updated only using the C :X utility enbaling the Installation/Uninstallation of the NT service.
This utility is documented in '***Connect:Express Windows Installation and Utilities***'.

3.3.3 Monitor files parameters

3.3.3.1 View files parameters (ApiCxShrViewPrmFil)

Description
This function is called by the application to view the Connect:Express monitor files parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_FIL structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_FIL structure is filled as following :

- *InitJnl* : takes on the value of 'O' [Oui = French for yes] if the journal file is to be reinitialized when the monitor is started, or 'N' if not
- *JnlSize* : maximum number of records that the journal file should contain after reinitialization
- *InitLog* : takes on the value of 'O' [Oui = French for yes] if the messages file is to be reinitialized when the monitor is started, or 'N' if not
- *LogSize* : maximum number of records that the messages file should contain after reinitialization
- *Language* : left blank – not used in V3.01. The language used is the one selected in the Windows settings or English if it is not supported by Connect:Express.
- *Trace* : takes on the value of 'O' [Oui = French for yes] if internal trace is to be activated, or 'N' if not
- *InitNot* : takes on the value of 'O' [Oui = French for yes] if the notifications file is to be reinitialized when the monitor is started, or 'N' if not
- *NotSize* : maximum number of records that the notifications file should contain after reinitialization

3.3.3.2 Update files parameters (ApiCxShrUpdatePrmFil)

Description
This function is called by the application to update the Connect:Express monitor files parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *InitJnl* : takes on the value of 'O' [Oui = French for yes] if the journal file is to be reinitialized when the monitor is started, or 'N' if not
- *JnlSize* : maximum number of records that the journal file should contain after reinitialization
- *InitLog* : takes on the value of 'O' [Oui = French for yes] if the messages file is to be reinitialized when the monitor is started, or 'N' if not
- *LogSize* : maximum number of records that the messages file should contain after reinitialization
- *Language* : left blank – not used in V3.01. The language used is the one selected in the Windows settings or English if it is not supported by Connect:Express.
- *Trace* : takes on the value of 'O' [Oui = French for yes] if internal trace is to be activated, or 'N' if not
- *InitNot* : takes on the value of 'O' [Oui = French for yes] if the notifications file is to be reinitialized when the monitor is started, or 'N' if not
- *NotSize* : maximum number of records that the notifications file should contain after reinitialization

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_FIL structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.4 Monitor authorization parameters

3.3.4.1 View authorization parameters (ApiCxShrViewPrmAut)

Description
This function is called by the application to view the Connect:Express monitor authorization parameters. To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_AUT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_AUT structure is filled as following :

- *Alias (Identity code)* : alias that makes up a part of the authorization code for the Connect:Express product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *Number* : number that makes up a part of the authorization code for the C :X product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.

3.3.4.2 Update authorization parameters (ApiCxShrUpdatePrmAut)

Description
This function is called by the application to update the Connect:Express monitor authorization parameters.
To call this function the application must have successfully called the API initialization function.
Note :
There software authorization parameters set in the ALIAS and NUMERO keywords of the initialization file are valid only for old authorization keys and are maintained for compatibility. The new asset protection key is delivered by Sterling Commerce as a file named license.key.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *Alias (Identity code)* : alias that makes up a part of the authorization code for the Connect:Express product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.
- *Number* : number that makes up a part of the authorization code for the C :X product, made up of a maximum of 80 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_AUT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.4. Operation to Load the license.key file (ApiCxNetLoadLicense)

Description
This function asks the monitor to load the parameters contained in the asset protection file license.key, previously placed in the monitor's installation directory. This file contains the software protection key and indicates to the monitor the options authorized as well as their expiration date. It is delivered by Sterling Commerce. If an asset protection key expires or if a new option has been subscribed, it is possible, by using this function, to update the key without having to stop the monitor.
To call this function the application must have successfully called the API initialization function, and have successfully set up a connection with a monitor.

Input
The application indicates:

- *CxId* : client/server connection ID
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXN_LOAD_LICENSE structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.3.5 Monitor notification parameters

3.3.5.1 View notification parameters (ApiCxShrViewPrmNot)

Description
This function is called by the application to view the Connect:Express monitor notification parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_NOT structure is filled as following :

- *Notify* : takes on the value of 'O' [Oui = French for yes] if global notification is active, or 'N' if not
- *CliName* : symbolic name of the client to be notified, made up of 8 upper case characters, and filled in with spaces

3.3.5.2 Update notification parameters (ApiCxShrUpdatePrmNot)

Description
This function is called by the application to update the Connect:Express monitor notification parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *Notify* : takes on the value of 'O' [Oui = French for yes] if global notification is active, or 'N' if not
- *CliName* : symbolic name of the client to be notified, made up of 8 upper case characters, and filled in with spaces

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_NOT structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the ***'Connect:Express Windows User's Guide'.***

3.3.6 Monitor TCP/IP parameters

3.3.6.1 View TCP/IP parameters (ApiCxShrViewPrmIp)

Description
This function is called by the application to view the Connect:Express monitor TCP/IP parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_IP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_IP structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the TCP/IP is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's IP address is to be effected, or 'N' if not
- *TrfPort* : number of the IP port receiving incoming calls for transfers, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *CliPort* : number of the IP port receiving incoming calls for Client connections, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL TCP/IP giving a Windows Sockest V1.1 to use, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.3.6.2 Update TCP/IP parameters (ApiCxShrUpdatePrmIp)

Description
This function is called by the application to update the Connect:Express monitor TCP/IP parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *State* : takes on the value of 'O' [Oui = French for yes] if the TCP/IP is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's IP address is to be effected, or 'N' if not
- *TrfPort* : number of the IP port receiving incoming calls for transfers, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *CliPort* : number of the IP port receiving incoming calls for Client connections, made up of a maximum of 5 numeric characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL TCP/IP giving a Windows Sockest V1.1 to use, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_IP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.7 Monitor SNA/LU6.2 parameters

3.3.7.1 View SNA/LU6.2 parameters (ApiCxShrViewPrmSna)

Description
This function is called by the application to view the Connect:Express monitor SNA LU6.2 parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_SNA structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_SNA structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the LU6.2 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's LU name is to be effected, or 'N' if not
- *LuName* : name of the local LU for receiving incoming calls, made up of 8 upper case characters, and filled in with spaces
- *AppcDllNm* : name of the SNA SERVER DLL APPC to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *CsvDllNm* : name of the SNA SERVER DLL CSV to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

3.3.7.2 Update SNA/LU6.2 parameters (ApiCxShrUpdatePrmSna)

Description
This function is called by the application to update the Connect:Express monitor SNA LU6.2 parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *State* : takes on the value of 'O' [Oui = French for yes] if the LU6.2 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's LU name is to be effected, or 'N' if not
- *LuName* : name of the local LU for receiving incoming calls, made up of 8 upper case characters, and filled in with spaces
- *AppcDllNm* : name of the SNA SERVER DLL APPC to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.
- *CsvDllNm* : name of the SNA SERVER DLL CSV to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_SNA structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.8 Monitor X.25 parameters

3.3.8.1 View X.25 parameters (ApiCxShrViewPrmX25)

Description
This function is called by the application to view the Connect:Express monitor X.25 parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_X25 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_X25 structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the X.25 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's address is to be effected, or 'N' if not
- *LocAddr* : local sub-address for receiving incoming calls, made up of a maximum of 15 characters, with at least one zero character (\0) at the end.
- *PortNb* : number of the port for receiving incoming calls, made up of a maximum of 2 characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL EX25 EICON to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

3.3.8.2 Update X.25 parameters (ApiCxShrUpdatePrmX25)

Description
This function is called by the application to update the Connect:Express monitor X.25 parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *State* : takes on the value of 'O' [Oui = French for yes] if the X.25 network is operational, or 'N' if not
- *CheckAddr* : takes on the value of 'O' [Oui = French for yes] if a check on the caller's address is to be effected, or 'N' if not
- *LocAddr* : local sub-address for receiving incoming calls, made up of a maximum of 15 characters, with at least one zero character (\0) at the end.
- *PortNb* : number of the port for receiving incoming calls, made up of a maximum of 2 characters, with at least one zero character (\0) at the end.
- *DllName* : name of the DLL EX25 EICON to use, made up of a maximum of 127 characters, with a zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_X25 structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.9 Monitor Named Pipe parameters

3.3.9.1 View Named Pipe parameters (ApiCxShrViewPrmNp)

Description
This function is called by the application to view the Connect:Express monitor named pipe parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_NP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

When the function returns successfully the APICXS_PRM_NP structure is filled as following :

- *State* : takes on the value of 'O' [Oui = French for yes] if the Named Pipe network is operational, or 'N' if not
- *NpName* : name of the Named Pipe for the client connections, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

3.3.9.2 Update Named Pipe parameters (ApiCxShrUpdatePrmNp)

Description
This function is called by the application to update the Connect:Express monitor named pipe parameters.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *State* : takes on the value of 'O' [Oui = French for yes] if the Named Pipe network is operational, or 'N' if not
- *NpName* : name of the Named Pipe for the client connections, made up of a maximum of 127 characters, with at least one zero character (\0) at the end.

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_PRM_NP structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc* and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***

3.3.10 SSL server parameters

3.3.10.1 List of SSL servers (ApiCxShrListSslparmSrv)

Description
This function is called to retrieve the list of the SSL servers defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_LIST_SSLPARMSRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.*** The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXS_LIST_SSLPARMSRV structure is filled as following :

- *ListBuffer* : Buffer containing the list of the SSL server symbolic names
- *NbNamesRet* : number of SSL server names stored in the list
- *TotDef* : total number of names defined in the Connect:Express

The list that is returned displays a series of symbolic SSL server names, each one made of 8 characters filled with blanks when necessary.

3.3.10.2 View an SSL server definition (ApiShrViewSslparmSrv)

Description
This function is called to retrieve the details of an SSL server definition defined in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *Name* : SSL server symbolic name (8 uppercase characters completed by spaces)

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_SSLPARMSRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express for Win 32 User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

When the function returns successfully the APICXS_SSLPARMSRV structure is filled as following :

- *Enabled* : SSL server state ('1' : Enabled, '0' : Not enabled)
- *StoreProvider* : Certificate provider. Character string 'STORE_PROV_SYSTEM' terminated by a binary zero.
- *StoreLocation* : Certificate location. Null terminated character string having one of the following values : 'SYSTEM_STORE_LOCAL_MACHINE', 'SYSTEM_STORE_SERVICES' or 'SYSTEM_STORE_CURRENT_USER'.
- StoreName : Null terminated optional character string 'My'.
- *SubjectDn* : Null terminated optional character string containing the subject distinguished name of the server certificate.
- *IssuerDn* : Null terminated optional character string containing the issuer distinguished name of the server certificate.
- *Protocol* : SSL protocol version ('1' :TLSV1,'3' :SSLV3, '2' :SSLV2)
- *CipherSuites* : Cipher suites. Optional null terminated character string composed of a list of 2 digit numbers. This string indicates the cipher suites to use. The following number can be used :

```
'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,
'06'  SSL_CK_RC4_128_WITH_MD5,
'07'  SSL_CK_DES_64_CBC_WITH_MD5,
'08'  SSL_CK_RC2_128_CBC_WITH_MD5
Exemple : 000103\0 indicates the suites 00, 01 and 03.
```

- *Trace* : Trace level indicator ('0' : None, '1' : Partial, '2' : Complete)
- *ServerPort* : TCP/IP port of the server (Maximum 5 digits terminated by a binary zero)

3.3.10.3 View an SSL server definition (ApiShrViewSslparmSrvEx)

Description
This function is similar to the preceding one, but the following field has been added to the structure:

- *SslBytesHeader* : Indicates if unencrypted data are prefixed by a 2 bytes length header ('0' : No bytes length header, '1' : 2 bytes length header)

The structure is named APICXS_SSLPARMSRV_EX.

3.3.10.4 Add/Update of an SSL server definition (ApiCxShrUpdateSslparmSrv)

Description
This function is called to add or update an SSL server definition in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:

- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor's initialization file name
- *Version* : Connect:Express API version
- *Name* : SSL server symbolic name (8 uppercase characters completed by spaces)
- *Enabled* : SSL server state ('1' : Enabled, '0' : Not enabled)
- *StoreProvider* : Certificate provider. Character string 'STORE_PROV_SYSTEM' terminated by a binary zero.
- *StoreLocation* : Certificate location. Null terminated character string having one of the following values : 'SYSTEM_STORE_LOCAL_MACHINE', 'SYSTEM_STORE_SERVICES' or 'SYSTEM_STORE_CURRENT_USER'.
- StoreName : Null terminated optional character string 'My'.
- *SubjectDn* : Null terminated optional character string containing the subject distinguished name of the sever certificate.
- *IssuerDn* : Null terminated optional character string containing the issuer distinguished name of the server certificate.
- *Protocol* : SSL protocol version ('1' :TLSV1,'3' :SSLV3, '2' :SSLV2)
- *CipherSuites* : Cipher suites. Optional null terminated character string composed of a list of 2 digit numbers. This string indicates the cipher suites to use. The following number can be used :

```
'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,
'06'  SSL_CK_RC4_128_WITH_MD5,
'07'  SSL_CK_DES_64_CBC_WITH_MD5,
'08'  SSL_CK_RC2_128_CBC_WITH_MD5
Exemple : 000103\0 indicates the suites 00, 01 and 03.
```

- *Trace* : Trace level indicator ('0' : None, '1' : Partial, '2' : Complete)
- *ServerPort* : TCP/IP port of the server (Maximum 5 digits terminated by a binary zero)

Output
The function's direct return code indicates the following:

- -1 : the address of the APICXS_SSLPARMSRV structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the '***Connect:Express Windows User's Guide'.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).

3.3.10.5 Add/Update of an SSL server definition (ApiCxShrUpdateSslparmSrvEx)

Description
This function is similar to the preceding one, but the following field has been added to the structure:

- *SslBytesHeader* : Indicates if unencrypted data are prefixed by a 2 bytes length header (‘0’ : No bytes length header, ‘1’ : 2 bytes length header)

The structure is named APICXS_SSLPARMSRV_EX.

3.3.10.6 Delete an SSL server definition (ApiCxShrDelSslparmSrv)

Description
This function is called to delete an SSL server definition in a Connect:Express monitor. It is only available if the SSL option is activated.
To call this function the application must have successfully called the API initialization function.

Input
The application indicates:
- *CliName* : symbolic client name
- *CliPsw* : client password
- *IniFileName* : monitor’s initialization file name
- *Version* : Connect:Express API version
- *Name* : SSL server symbolic name (8 uppercase characters completed by spaces)

Output
The function’s direct return code indicates the following:
- -1 : the address of the APICXN_ DEL_CLI structure indicated by the application is incorrect
- 0 : the function has failed, see API return code
- 1 : the function has executed correctly

The *ApiRc*, *SysRc*, TcpRc and *TomRc* codes are described in the ‘***Connect:Express for Win 32 User’s Guide’.***
The TomRc code is accompanied by a description of the error given by the monitor (*LibErr*).