Connection Services APIs

Introduction to Connection Services

The Connection Service calls allow developers to establish and destroy logical connections to a NetWare file server (creating utilities similar to the NetWare LOGIN, ATTACH and LOGOUT utilities), and return status information about those connections. Connection Services enable applications to do the following:

- Log in or attach objects to file servers
- Log out or detach objects from file servers
- Return information about a connection
- Return a clientConnID or a serverConnID

Connection Information

Connection information must be maintained by both the server and the connected client. The file server maintains two related tables:

- The File Server Connection Table
- The Password Table

The number of entries allowed in the table depends upon which version of the operating system the file server is running. Each entry in the File Server Connection table contains the network address of a client. The corresponding entry in the Password table contains the bindery object ID of the object type that established the connection between that client and the file server. The file server identifies a connection (both the connected client and the object attached through that client) by the connection's position (1 to 250) in these tables. This connection is known to the client as the clientConnID.

The following information is maintained in order to maintain a client connection:

- serverConnID
- clientConnID

The serverConnID is a number which represents a server to a client. The serverConnID is returned to the client by the NWAttachToServerPlatform and the NWGetServerConnID function calls.

NWAttachToServerPlatform

This function attaches the default client to the named file server.

Synopsis

```
#include ªnwapi.hº

int             ccode;
char            fileServerName[NWMAX_SERVER_NAME_LENGTH];
uint16          serverConnID;

ccode=NWAttachToServerPlatform( fileServerName, &serverConnID );
```

Input

fileServerName passes a pointer to the name of the target file server.

serverConnID passes a pointer to the space allocated for the file server connection ID.

Output

serverConnID receives the file server connection ID.

Return Values

0   Successful.

-1   Unsuccessful. One of the following error codes is placed in NWErrno:

      0xF8   Already Attached to Server
      0xF9   No Free Connection Slots
      0xFC   Unknown File Server
      0xFF   No Response From Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

This function initializes and sets up a client connection to a server and allows the client to log in to the server by using the NWLoginToServerPlatform function. This function returns a file server connection ID (serverConnID) for the new connection and places the newly attached file server's connection information in the client's connection tables. After using this function, the client can now log in to the file server as an object.

The fileServerName array should contain either the name of the file server to attach to or an asterisk (*). If an asterisk is contained in the fileServerName array, the application will attach to the nearest file server. The asterisk may be used for utilities which do not require logging in.

Notes

This function will automatically open a transport.

See Also

NWLoginToServerPlatform

NWClearClientConnID

This function clears a client connection number on the file server.

Synopsis

#include ªnwapi.hº

```
int              ccode;
uint16           serverConnID;
uint16           clientConnID;
```

ccode=**NWClearClientConnID**(serverConnID, clientConnID);

Input

serverConnID passes the file server connection ID.

clientConnID passes the client connection number to be cleared.

Output

None.

Return Values

0   Successful.

-1 Unsuccessful. One of the following error codes is placed in NWErrno:

>0x35 Server Connection Lost
>0xFC No Such Object
>0xEF Illegal Name
>0xC1 No Account Balance
>0xC5 Login Lockout
>0x35 Server Connection Lost

See Appendix B for a listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

Clearing a connection will log a client off the network. The client must then reattach and log in again to establish a new connection.

The calling application must be logged in as supervisor or have equivalent rights.

See Also

NWGetObjectClientConnIDs

NWCloseTask

This function closes the process's task on the connection specified by serverConnID.

Synopsis

```
#include ªnwapi.hº

int              ccode;
uint16           serverConnID;

ccode=NWCloseTask( serverConnID );
```

Input

serverConnID passes the file server connection ID.

Output

None.

Return Values

0 Successful.
-1 Unsuccessful. One of the following error codes is placed in NWErrno:

>0xFF Connection Does Not Exist
>0xFC Unknown File Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

Closing a NetWare task will destroy process-specific context at the server and will free up a management portal device on the client. Generally it should be regarded as similar to closing a file. A new task will automatically be generated if another API call is made using that connection ID.

NWDetachFromServerPlatform

This function breaks a client-file server connection.

Synopsis

#include ªnwapi.hº

```
int              ccode;
uint16           serverConnID;
```

ccode=**NWDetachFromServerPlatform**(serverConnID);

Input

serverConnID passes the file server connection ID.

Output

None.

Return Values

 0   Successful.

-1   Unsuccessful. One of the following error codes is placed in NWErrno:

    0xFF   Connection Does Not Exist
    0xFC   Unknown File Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

Detaching from a file server is not the same as logging out from a file server. Detaching relinquishes the connection number the client was using and breaks the connection. Before the client can send further requests to that file server, it must be reattached. Logging out from a file server preserves the connection ID and allows the client to log in again without reattaching.

See Also

NWAttachToServerPlatform
NWLogoutFromServerPlatform

NWGetClientConnID

This function returns the connection number that the requesting client uses to communicate with the default file server.

Synopsis

#include ªnwapi.hº

```
int              ccode;
uint16           serverConnID;
uint16           clientConnID;
```

ccode=**NWGetClientConnID**(serverConnID, &clientConnID);

Input

serverConnID passes the file server connection ID.

clientConnID passes a pointer to a location that will store the client connection number.

Output

clientConnID receives the requesting client connection ID number.

Return Values

0   Successful.

-1   Unsuccessful. One of the following error codes is placed in NWErrno:

      0xFB   Invalid Parameters
      0x04   Not Connected To Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

The clientConnID parameter is an index into the Connection Table maintained by the file server.

See Also

NWAttachToServerPlatform
NWGetServerConnIDList

NWGetConnectionInformation

This function allows you to get information about a file server connection.

Synopsis

#include ªnwapi.hº

```
int             ccode;
uint16          serverConnID;
uint16          clientConnID;
char            clientObjectName[NWMAX_OBJECT_
                      NAME_LENGTH];
uint16          clientObjectType;
uint32          clientObjectID;
uint8           clientLoginTime[NWMAX_LOGIN_TIME_LENGTH];
```

ccode=**NWGetConnectionInformation**(serverConnID, clientConnID, clientObjectName, &clientObjectType, &clientObjectID, clientLoginTime);

Input

serverConnID passes the file server connection ID.

clientConnID passes the client connection ID.

clientObjectName passes a pointer to the space allocated for the client's object name.

clientObjectType passes a pointer to the space allocated for the client's object type.

clientObjectID passes a pointer to the space allocated for the client's object ID.

clientLoginTime passes a pointer to the space allocated for the client's login time.

Output

clientObjectName receives the client's object name.

clientObjectType receives the client's object type.

clientObjectID receives the client's object ID.

clientLoginTime receives the client's login time.

Return Values

0   Successful.

-1   Unsuccessful. One of the following error codes is placed in NWErrno:

        0xFF   No Response From Server
        0xFE   Server Bindery Locked
        0xFC   No Such Object

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

The clientLoginTime is returned in an array of 7 uint8s. The array will be filled with the following:

1st uint8=year	(0 through 99; for example: 90=1990)
2nd uint8=month	(1 through 12)
3rd uint8=day	(1 through 31)
4th uint8=hour	(0 through 23)
5th uint8=minute	(0 through 59)
6th uint8=second	(0 through 59)
7th uint8=dayOfWeek	(0 through 6, 0 = Sunday)

NWGetInternetAddress

This function gets the internetwork address of any client on the network.

Synopsis

#include ªnwapi.hº

int	ccode;
uint16	serverConnID;
uint16	clientConnID;
uint8	internetAddress[**NWMAX_INTERNET_ADDRESS_**
	LENGTH];

ccode=**NWGetInternetAddress**(serverConnID, clientConnID, internetAddress);

Input

serverConnID passes the file server connection ID.

clientConnID passes the client connection ID.

internetAddress passes a pointer to the space allocated for the internet address.

internetAddress receives the internet address.

Return Values

0	Successful.
-1	Unsuccessful. One of the following error codes is placed in NWErrno:

0x01	Invalid Parameter Length
0x04	Not Connected To Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

An internetwork address is comprised of the networkNumber, the physicalNodeAddress and the socketNumber. The internetwork address uniquely identifies a client throughout an internetwork. This address can be used to send packets directly to the client. The physicalNodeAddress is the address of the client's LAN board.

NWGetObjectClientConnIDs

This function returns a list of server-maintained client connection ID numbers for a specified logged-in object.

Synopsis

#include ªnwapi.hº

int	ccode;
uint16	serverConnID;
char	objectName[**NWMAX_OBJECT_NAME_LENGTH**];
uint16	objectType;
uint16	numberOfConnections;
uint16	connectionList[n];
uint16	maxListElements;

ccode=**NWGetObjectClientConnIDs**(serverConnID, objectName, objectType, &numberOfConnections, connectionList, maxListElements);

Input

serverConnID passes the file server connection ID.

objectName passes a pointer to the bindery object name of the object whose file server connection numbers are being returned (wildcards not allowed).

objectType passes the bindery object type of the object whose file server connection numbers are being obtained. (See Appendix A, ªBindery Object Types.º)

numberOfConnections passes a pointer to the space allocated for the number of server connections found for the specified object.

connectionList passes a pointer to the array allocated for the object's server connection ID numbers.

maxListElements passes the number of connectionList elements that have been allocated (n).

Output

numberOfConnections receives the number of server connections found for the specified object.

connectionList receives the server connection numbers for the specified object.

Return Values

   0   Successful.

  -1   Unsuccessful. One of the following error codes is placed in NWErrno:

        0xFC   No Such Object
        0xEF   Illegal Name
        0xC1   No Account Balance
        0xC5   Login Lockout

See Appendix B for a listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

You should allocate as many elements as you think necessary for the connectionList parameter and then put that amount in the maxListElements parameter. This function will return less than or equal the amount that the application specifies. If fewer clientConnIDs were found than were requested, only the numberOfConnections amount will be copied into connectionList.

See Also

NWClearClientConnID

NWGetPrimaryConnectionID

This function returns the connection ID of the primary file server.

Synopsis

#include ªnwapi.hº

int   ccode;
uint16   serverConnID;

ccode=**NWGetPrimaryConnectionID**(&serverConnID);

Input

serverConnID passes a pointer to the space allocated for the primary server connection ID.

Output

serverConnID receives the primary file server connection ID.

Return Values

   0   Successful.

  -1   Unsuccessful. An error code will be placed in NWErrno:

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

This function gets the primary server connection ID (serverConnID), which is used in most other function calls to specify the server connection the client will access. The primary server connection ID is set with the NWSetPrimaryConnectionID call. This ID is the same for all processes with the same e_uid on a client. It can be set with the authenticator program. This call will only return a serverConnID if a connection has already been established and registered as the primary connection.

NWGetServerConnID

This function returns the file server connection ID.

Synopsis

#include ªnwapi.hº

int	ccode;
char	fileServerName[**NWMAX_SERVER_NAME_LENGTH**];
uint16	serverConnID;

ccode=**NWGetServerConnID**(fileServerName, &serverConnID);

Input

fileServerName passes a pointer to the name of the target file server.

serverConnID passes a pointer to the space allocated for the server connection ID.

Output

serverConnID receives the file server connection ID.

Return Values

0   Successful.
-1   Unsuccessful. An error code will be placed in NWErrno.

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

This function gets the server connection ID (serverConnID), which is used in most other function calls to specify the server connection the client will access. This call will only return a serverConnID if a connection has already been established.

NWGetServerConnIDList

This function returns a list of all current server connection IDs for the requesting client.

Synopsis

int	ccode;
uint16	connectionListBuffer[**NWMAX_CONNECTION_**
	LIST_LENGTH];
uint16	bufferSize;
uint16	numberOfConnsReturned;

```
ccode=NWGetServerConnIDList( connectionListBuffer, bufferSize,
&numberOfConnsReturned );
```

Input

connectionListBuffer passes a pointer to the buffer allocated for the connection list.

bufferSize passes the number of connectionListBuffer elements which were allocated.

numberOfConnsReturned passes a pointer to the buffer in which the number of connections is returned.

Output

connectionListBuffer receives the connection list.

numberOfConnsReturned receives the number of connections that were actually placed in the connectionListBuffer.

Return Values

0	Successful.
-1	Unsuccessful. One of the following error codes is placed in NWErrno:

0xFB	Invalid Parameters
0x04	Not Connected To Server

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

A client can be connected to a maximum of 15 servers. This value is the maximum size of the connectionListBuffer array.
NWMAX_ CONNECTION_LIST_LENGTH sets the limit.

The returned connection list contains all the server connection ID numbers in use by a client. This function will not return any more connection numbers than the number you specify with the bufferSize parameter. This function allows the application to control how many connections it is aware of.

See Also

NWGetClientConnID
NWGetServerPlatformInformation

NWLoginToServerPlatform

This function logs an object in to a connected file server.

Synopsis

#include ªnwapi.hº

```
int                 ccode;
uint16              serverConnID;
char                name[NWMAX_OBJECT_NAME_LENGTH];
uint16              objectType;
char                password[NWMAX_PROPERTY_NAME_LENGTH];

ccode=NWLoginToServerPlatform( serverConnID, name, objectType,
password );
```

Input

serverConnID passes the file server connection ID.

name passes the name of the client to be logged in.

objectType passes the client object type to be logged in. (See Appendix A, ªBindery Object Types.º)

password passes the object password.

Output

None.

Return Values

0   Successful.
-1   Unsuccessful. One of the following error codes is placed in NWErrno:

0xFC   No Such Object
0xEF   Illegal Name
0xC1   No Account Balance
0xC5   Login Lockout
0xDE   Bad Password
0xDF   Old Password

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

This call enables a client to identify itself to a file server and thereby be cleared for access to the network's file system and resources. The requesting client must first use NWAttachToServerPlatform to establish a connection with the specified file server before logging in to the server.

Notes

If the object does not have a password, a null should be passed in place of the password parameter. Attaching to a file server is not the same as logging in. A client attaches to a file server to obtain a connection number (clientConnID). The client can then log in to the file server using that connection number.

See Also

NWAttachToServerPlatform
NWGetClientConnID
NWGetServerConnID
NWLogoutFromServerPlatform

NWLogoutFromServerPlatform

This function logs out the requesting client from the specified server.

Synopsis

#include ªnwapi.hº

```
int              ccode;
uint16           serverConnID;
```

ccode=**NWLogoutFromServerPlatform**(serverConnID);

Input

serverConnID passes the file server connection ID.

Output

None.

Return Values

  0   Successful.

-1   Unsuccessful. One of the following error codes is placed in NWErrno:

      0xFF   Hardware Error
      0x96   Server Out Of Memory

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

NWLogoutFromServerPlatform removes the requesting client's clientConnID from the file server but does not relinquish the serverConnID. Therefore a connection still exists to that server.

Detaching from a file server relinquishes the connection ID and breaks the connection, forcing the client to reattach before sending any further requests to the server.

See Also

NWDetachFromServerPlatform
NWLoginToServerPlatform

NWSetPrimaryConnectionID

This function sets the connection ID of the primary file server.

Synopsis

#include ªnwapi.hº

int   ccode;
uint16   serverConnID;

ccode=**NWSetPrimaryConnectionID**(serverConnID);

Input

serverConnID passes the primary server connection ID.

Output

None

Return Values

  0   Successful.

-1   Unsuccessful. An error code will be placed in NWErrno.

See Appendix B for a complete listing of possible NetWare errors and a description of the four bytes in NWErrno.

Description

This function sets the primary server connection ID (serverConnID). The primary server connection ID is the ID that is returned by the
NWSetPrimaryConnID call. This ID is the same for all processes with the same e_uid on a client. It is the connection ID used as a default by programs that do not get a server name from the user.

See Also

NWGetPrimaryConnectionID