Connection Services

Connection services are used to create connections to file servers and return information about those connections and the clients that made them.

This chapter is divided into the following sections:

- Attaching and Detaching
- Logging In and Logging Out
- Connection Tables

Attaching and Detaching

In order to access and manage NetWare file servers and their associated network resources, an API client must first obtain a task on a connection to a server. The call NWGetServerConnID performs this function if a server connection for the current user already exists. If no connection exists, NWAttachToServerPlatform will both create a connection and allocate a task on that connection.

NWGetServerConnID requests a list from the kernel of all servers that the current user has an open connection to and, if the requested server is in the list, returns its connection ID. (In this case the actual task is created by the first API call that needs to send a packet over the wire.) NWAttachToServerPlatform passes the name of the desired server to the kernel, which is responsible for creating the connection and opening the task.

Once a connection is established for a user, that connection remains until it is either deliberately broken by the user, or the server or client reboots. Tasks are destroyed when the application terminates or when NWCloseTask is called. Because tasks consume system resources, NWCloseTask should be called if a task will not be used for a period of time. The connection ID will still be valid and a new task will be created if needed.

Generally speaking, connections will be created and destroyed by the user using the utilities provided and should not be destroyed by an application. Because multiple applications as well as the file system may be using the same connection, one application doing an NWDetachFromServerPlatfom may cause another application or the file system to fail.

NWClearConnID can be used to destroy any connection on the server that the user has permission to destroy. This will rarely be needed outside of network management applications.

The NetWare watchdog will also clear a client's connection when the client does not respond to the watchdog packets. (The watchdog is handled by the kernel and does not interact with the APIs.) If the watchdog has not heard from a client in 5 minutes, the watchdog sends a watchdog packet to the client. If the client does not respond, the watchdog starts sending watchdog packets at 1-minute intervals for up to 10 minutes. If the client does not respond within 15 minutes, the watchdog assumes that the client is no longer connected to the file server and clears the client's connection.

Once a client has been detached, the client must then re-attach before the client can log in again.

Logging In and Logging Out

After attaching to a file server, an application must log in to the file server by using NWLoginToServerPlatform. Logging in allows the client to gain the rights and privileges necessary to manage and manipulate network resources. To log in to a file server, an application must provide a bindery object's name, type and password. When an object logs in to a file server, the file server puts the object's ID number in the file server's Password table and in the Connection Information Table.

When an object logs out of a file server, the file server removes the object's ID from the file server's Password table. However, the object's ID is not removed from the Connection Information Table until the object detaches from the file server.

Connection Tables

Two connection tables are kept, one by a module in the kernel and the other by the file server.

Client Connection Table

A copy of the Client Connection table is kept by the APIs for each client attached to the file server. Various APIs allow the client to query the table for the information listed below. The other fields in the table are for internal use only and contain information such as packet size and the number of packet retries.

This table contains the following information that APIs can query.

File server connection number (serverConnID). This field contains the file server connection number. A module in the kernel assigns this number
(0 through *n*) as the client attaches to file servers. The first file server, or the client's default file server, is assigned 0; the next file server, 1; and so forth. If the client has already attached to the maximum number of file servers supported by the kernel configuration, the attempt to attach to a ninth file server fails.

File server name. This field contains the name of the file server that the client is attached or logged in to.

NetWare version. This field contains the version of the operating system that the file server is running.

Client connection number (clientConnID). This field contains the client connection number. The file server assigns this number (0 through 250) as the file server grants a connection slot to a client. If all of the file server's connection slots have been assigned to other clients, the attempt to attach fails. The number of slots varies with the version of the operating system.

Connection Information Table

The file server maintains the Connection Information Table. This table contains all clients attached or logged in to the file server. The APIs use the client's
clientConnID and serverConnID to query the table for the following information.

Object Name (clientObjectName). This field contains the object's name. Each object must be created on the file server before the object can log in. (For example, you can create users with SYSCON or print queues with PCONSOLE.) Each object must be given a unique name for its type. This name becomes the clientObjectName that must be passed by the API for a client to log in.

Object ID (clientObjectID). This field contains the object ID. As each object is created, the file server assigns the object a unique object ID. This number is used to identify the object without specifying the object's name and type.

Object Type (clientObjectType). This field contains the object type. As each object is created, the object must be assigned a bindery type. When a user is created with SYSCON, SYSCON assigns type User to the object. When a group is created with SYSCON, SYSCON assigns type Group to the object.

Login Time (clientLoginTime). This field contains the date and time that the object logged in to the file server.

Network address and Node address (internetAddress). This field contains the unique address of the client. This address is made up of the client's network address and node address. The network address is the address that has been assigned to the cabling system the client is physically cabled to. The node address is the address of the network board installed in the client's workstation.

Using Client Connection IDs

The clientConnID (or client connection number) indicates the number allocated by the file server to a particular client. It is important not to confuse clientConnID with serverConnID. These numbers can provide an easy way to identify objects logged in on the network and obtain additional information about them. The API call NWGetConnectionInformation uses both numbers (serverConnId and
clientConnID) to return the object name, type, ID and login time about the object that has made the connection. Likewise, the API call NWGetInternetAddress uses both numbers (serverConnID and clientConnID) to return the

network and node address a connection is using.

A list of all the connections for a particular bindery object can be obtained with NWGetObjectClientConnIDs. To make this API call, an application must pass in the object's name and type. A client can obtain its own connection number with the default file server with NWGetClientConnID.