

# Server Platform Services APIs

## NWDisableServerPlatformLogin

This function disables the ability of clients to log in to the specified server.

### Synopsis

```
#include "nwapi.h"

int          ccode;
uint16       serverConnID;

ccode=NWDisableServerPlatformLogin( 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: |
|    | 0xC6          | No Console Operator Rights                             |
|    | 0x96          | Server Out Of Memory                                   |
|    | 0xFB          | Invalid Parameters                                     |
|    | 0xFF          | No Response From Server                                |

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

### Description

This call is useful for working with the server after all clients have been logged out (for example, during backup procedures).

### Notes

IMPORTANT: You must remember to enable server login before logging out of the server or no one, including the client that made this call, will be able to access the server again.

### See Also

NWEnableServerPlatformLogin

## NWDownServerPlatform

This function will down the specified server.

### Synopsis

```
#include "nwapi.h"
```

```
int          ccode;
uint16       serverConnID;
uint8        forceFlag;

ccode=NWDownServerPlatform( serverConnID, forceFlag );
```

**Input**

*serverConnID* passes the file server connection ID of the server that will be shut down.

*forceFlag* passes one of the following flags:

- |      |   |
|------|---|
| 0x00 | If any files are open, the call returns 0xFF and the server will not go down. |
| 0xFF | Server shuts down after automatically closing all open files.                 |

**Output**

None.

**Return Values**

- |      |  |
|------|--|
| 0    | Successful.  |
| -1   | Unsuccessful. One of the following error codes is placed in NWErrno: |
| 0xC6 | No Console Operator Rights   |
| 0x96 | Server Out Of Memory   |
| 0xFB | Invalid Parameters   |
| 0xFF | No Response From Server  |

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

**Description**

This server will check for any open files and will not go down unless you set the force flag to 0xFF, in which case all unfinished transactions will be backed out and all open files will be closed. The operating system will then be shut down and must be re-booted or brought up from the command line.

**NWEnableServerPlatformLogin**

This function enables clients to log in to the specified server.

**Synopsis**

```
#include "nwapi.h"

int          ccode;
uint16       serverConnID;

ccode=NWEnableServerPlatformLogin( 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: |
| 0xC6 | No Console Operator Rights   |
| 0x96 | Server Out Of Memory   |
| 0xFB | Invalid Parameters   |
| 0xFF | No Response From Server  |

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

## Description

This function enables client login. If clients are already logged in when this function call is made, there will be no interruption of service to those clients.

## See Also

NWDisableServerPlatformLogin

## NWGetDiskUtilization

This function call returns information about the total disk usage of a specified client.

## Synopsis

```
#include "nwapi.h"
```

```
int          ccode;  
uint16      serverConnID;  
uint8       volNumber;  
uint32      trusteeID;  
uint16      dirCount;  
uint16      fileCount;  
uint16      blockCount;
```

```
ccode=NWGetDiskUtilization( serverConnID, volNumber, trusteeID, &dirCount, &fileCount, &blockCount );
```

## Input

*serverConnID* passes the file server connection ID.

*volNumber* passes the number of the volume being checked.

*trusteeID* passes the client trustee ID (objectID of the trustee).

*dirCount* passes a pointer to the space allocated for the directory count.

*fileCount* passes a pointer to the space allocated for the file count.

*blockCount* passes a pointer to the space allocated for the cluster count.

## Output

*dirCount* receives the number of directories owned by the trustee in the specified volume.

*fileCount* receives the number of files in the specified volume owned by the trustee.

*blockCount* receives the number of blocks used in the specified volume by the trustee.

## Return Values

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

|      |                            |
|------|----------------------------|
| 0xC6 | No Console Operator Rights |
| 0x96 | Server Out Of Memory       |
| 0xFB | Invalid Parameters         |
| 0xFF | No Response From Server    |

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

### Description

This function returns disk utilization information about a specified client (or trustee) based on their object ID number. NetWare blocks are usually 4K, but can be configured otherwise when the file server is initially configured. The NWGetVolInfoWithHandle (in File Services) returns exact information on the actual size of blocks on the given file server.

## Notes

This call is not valid for NetWare for UNIX servers. NetWare for UNIX servers do not maintain this information on a per-client (objectID) basis.

## See Also

## NWGetVolInfoWithHandle

## NWGetServerPlatformDateAndTime

This function returns the network date and time maintained on the specified file server.

## Synopsis

```
#include "nwapi.h"

int                                     ccode;
uint16                                serverConnID;
NWServerPlatformDateAndTime_t         dateAndTime;

ccode=NWGetServerPlatformDateAndTime( serverConnID,
&dateAndTime );
```

## Input

*serverConnID* passes the file server connection ID.

*dateAndTime* passes a pointer to the structure allocated for the network date and time. (See "Description" on the next page and Appendix A ,  
<sup>a</sup>NWServerPlatformDateAndTime t Structure.°)

## Output

*dateAndTime* receives network date and time. (See "Description" on the next page and Appendix A, <sup>a</sup>NWServerPlatformDateAndTime t Structure.)

**Return Values**

|    |               |  |
|----|---------------|--|
| 0  | Successful.   |  |
| -1 | Unsuccessful. | One of the following error codes is placed in NWErrno: |
|    | 0xC6          | No Console Operator Rights                             |
|    | 0x96          | Server Out Of Memory                                   |
|    | 0xFB          | Invalid Parameters                                     |
|    | 0xFF          | No Response From Server                                |

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

**Description**

The NWServerPlatformDateAndTime\_t structure is as follows:

```
typedef struct {
uint8 year      (0 through 99; for example: 82=1982; )
uint8 month     (1 through 12)
uint8 day       (1 through 31)
uint8 hour      (0 through 23)
uint8 minute    (0 through 59)
uint8 second    (0 through 59)
uint8 dayOfWeek (0 through 6, 0 = Sunday)
} NWServerPlatformDateAndTime_t;
```

**Notes**

Date and time are not automatically synchronized across an internetwork.

If the year is less than 82, the year is considered to be in the 21st century.

**See Also**

NWSetServerPlatformDateAndTime

**NWGetServerPlatformDescriptionStrings**

This function returns descriptive information about a file server.

**Synopsis**

```
#include "nwapi.h"

int          ccode;
uint16      serverConnID;
NWDescriptionStrings_t strings;

ccode=NWGetServerPlatformDescriptionStrings( serverConnID, &strings );
```

**Input**

*serverConnID* passes the file server connection ID.

*strings* passes a pointer to the structure allocated for the file server description strings. (See "Description" below and Appendix A, <sup>a</sup>NWDescriptionStrings\_t Structure.)

**Output**

*strings* receives the file server description strings. (See "Description" below and Appendix A, <sup>a</sup>NWDescriptionStrings\_t Structure.)

**Return Values**

- 0           Successful.
- 1          Unsuccessful.   One of the following error codes is placed in NWErrno:
- 0xC6           No Console Operator Rights

0x96           Server Out Of Memory

0xFB           Invalid Parameters

0xFF           No Response From Server

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

**Description**

The NWDescriptionStrings\_t structure is as follows:

```
typedef struct {
    char   companyName[ NWMAX_COMPANY_NAME_LENGTH ];
    char   revisionDescription[ NWMAX_DESCRIPTION_LENGTH ];
    char   revisionDate[ NWMAX_DATE_LENGTH ];
    char   copyrightNotice[ NWMAX_COPYRIGHT_NOTICE_LENGTH ];
} NWDescriptionStrings_t;
```

*companyName* receives the name of the company that is providing this version of NetWare.

*revisionDescription* receives the NetWare version and revision description string.

*revisionDate* receives the revision date in the form 02/15/1988.

*copyrightNotice* passes a pointer to the string allocated for the copyright notice.

**Notes**

The requesting workstation must be attached to the file server. Console operator rights are not required to perform this function.

**NWGetServerPlatformInformation**

This function obtains information about the specified file server.

**Synopsis**

```
#include "nwapi.h"

int                                   ccode;
uint16                               serverConnID;
NWServerPlatformInfo_t               serverInfo;

ccode=NWGetServerPlatformInformation( serverConnID,  &serverInfo );
```

**Input**

*serverConnID* passes the file server connection ID.

*serverInfo* passes a pointer to the structure allocated for the server information. (See "Description" below and Appendix A, <sup>a</sup>NWServerPlatformInfo\_t Structure.)

## Output

*serverInfo* receives the server information. (See "Description" below and Appendix A, "NWServerPlatformInfo\_t Structure.")

## Return Values

|      |  |
|------|--|
| 0    | Successful.  |
| -1   | Unsuccessful. One of the following error codes is placed in NWErrno: |
| 0xC6 | No Console Operator Rights   |
| 0x96 | Server Out Of Memory   |
| 0xFB | Invalid Parameters   |
| 0xFF | No Response From Server  |

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

## Description

This routine optionally returns several items of information about a file server based on a specified file server connection ID. If a certain item is not wanted, a NULL parameter can be substituted in place of the unwanted parameter. However, all parameter positions must be occupied.

The NWServerPlatformInfo\_t structure is as follows:

```
typedef struct {
    uint16    majorVersion;
    uint16    minorVersion;
    uint16    revision;
    uint16    SFTLevel;
    uint16    TTSLevel;
    uint16    accountingVersion;
    uint16    VAPVersion;
    uint16    queueingVersion;
    uint16    printServerVersion;
    uint16    virtualConsoleVersion;
    uint16    securityRestrictionLevel;
    uint16    internetBridgeSupport;
    uint16    maxClientConnSupported;
    uint16    clientConnInUse;
    uint16    peakClientConnUsed;
    uint16    maxVolumes;
    char      serverName[NWMAX_SERVER_NAME_LENGTH];
} NWServerPlatformInfo_t;
```

## See Also

NWGetServerPlatformDescriptionStrings

## NWGetServerPlatformLoginStatus

This function gets the status of the server login-enabled or disabled.

## Synopsis

```
#include "nwapi.h"

int      ccode;
uint16   serverConnID;
```

```
uint8          userLoginAllowed;

ccode=NWGetServerPlatformLoginStatus( serverConnID,
&userLoginAllowed );
```

## Input

*serverConnID* passes the file server connection ID.

*userLoginAllowed* passes a pointer to the space allocated for the status of user login.

## Output

*userLoginAllowed* receives the status of user login:

0 = User Login Disabled  
Non-zero = User Login Enabled

## Return Values

|      |  |
|------|--|
| 0    | Successful.  |
| -1   | Unsuccessful. One of the following error codes is placed in NWErrno: |
| 0xC6 | No Console Operator Rights   |
| 0x96 | Server Out Of Memory   |
| 0xFB | Invalid Parameters   |
| 0xFF | No Response From Server  |

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

## Description

The userLoginAllowed parameter receives a non-zero value if user login is enabled, and a 0 if user login is disabled. You must be a supervisor or supervisor equivalent to make this call.

## See Also

NWEnableServerPlatformLogin  
NWDisableServerPlatformLogin

## NWGetServerPlatformName

This function gets the name of the server platform with the connection ID.

## Synopsis

```
#include "nwapi.h"

int          ccode;
uint16       serverConnID;
char         serverPlatformName[NWMAX_SERVER_
NAME_LENGTH];

ccode=NWGetServerPlatformName( serverConnID, serverPlatformName );
```

## Input

*serverConnID* passes the file server connection ID.

*serverPlatformName* passes a pointer to the string allocated for the server name.



**Output**

*serverPlatformName* receives the server name.

**Return Values**

- |    |               |  |
|----|---------------|--|
| 0  | Successful.   |  |
| -1 | Unsuccessful. | One of the following error codes is placed in NWErrno: |
|    | 0xFB          | Invalid Parameters                                     |
|    | 0xF8          | Not Attached To Server                                 |

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

**Description**

This call accesses the connection tables maintained by the APIs and returns the name of the server attached to with NWAttachToServerPlatform.

**See Also**

NWAttachToServerPlatform

**NWIsNetWare386**

This function checks whether a connected file server is running NetWare v3.x.

**Synopsis**

```
#define "nwapi.h"

NWBoolean_ts      ccode;
uint16            serverConnID;

ccode=NWIsNetWare386( serverConnID );
```

**Input**

*serverConnID* passes the file server connection ID.

**Output**

None.

**Return Values**

- |   |                                    |
|---|------------------------------------|
| 1 | Server is running NetWare v3.x     |
| 0 | Server is not running NetWare v3.x |

**Description**

NetWare for UNIX corresponds to NetWare v3.x.

**NWSetServerPlatformDateAndTime**

This function sets the network date and time on the specified file server.

**Synopsis**

```
#include "nwapi.h"
```

```
int                                     ccode;
uint16                                serverConnID;
NWServerPlatformDateAndTime_t         dateAndTime;

ccode=NWSetServerPlatformDateAndTime( serverConnID, &dateAndTime );
```

**Input**

*serverConnID* passes the file server connection ID.

*dateAndTime* passes a pointer to the structure allocated for the network date and time. (See <sup>a</sup>Description<sup>o</sup> on the next page and Appendix A, <sup>a</sup>NWServerPlatformDateAndTime\_t Structure.<sup>o</sup>)

**Output**

None.

**Return Values**

|    |               |  |
|----|---------------|--|
| 0  | Successful.   |  |
| -1 | Unsuccessful. | One of the following error codes is placed in NWErrno: |
|    | 0xC6          | No Console Operator Rights                             |
|    | 0x96          | Server Out Of Memory                                   |
|    | 0xFB          | Invalid Parameters                                     |
|    | 0xFF          | No Response From Server                                |

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

**Description**

The NWServerPlatformDateAndTime\_t structure is as follows:

```
typedef struct {
    uint8 year      (0 through 99; for example: 82=1982; )
    uint8 month     (1 through 12)
    uint8 day       (1 through 31)
    uint8 hour      (0 through 23)
    uint8 minute    (0 through 59)
    uint8 second    (0 through 59)
    uint8 dayOfWeek (0 through 6, 0 = Sunday)
} NWServerPlatformDateAndTime_t;
```

**Notes**

Date and time are not automatically synchronized across an internetwork.

If the year is less than 82, the year is considered to be in the 21st century.

This call is not valid for NetWare for UNIX servers. If the system date and time need to be changed, the host administrator should be contacted.

**See Also**

NWGetServerPlatformDateAndTime