Accounting Services APIs

Introduction to Accounting Services

The four accounting service calls enable developers to create servers that can charge for their services. For example, a database server can charge for the number of records viewed, the number of requests serviced or the amount of time connected. A print server can charge for the number of pages printed.

To charge for services, a server must be a member of the ACCOUNT_SERVERS property of the file server. To use accounting services, you must be familiar with the NetWare file server bindery.

NWGetAccountStatus

This function returns the account status of a bindery object.

Synopsis

```
#include ªnwapi.hº

int             ccode;
uint16          serverConnID;
uint16          objectType;
char            objectName[NWMAX_OBJECT_NAME_LENGTH];
int32           balance;
int32           limit;
NWHoldInfo_t    holds[NWMAX_NUMBER_OF_HOLDS];

ccode=NWGetAccountStatus( serverConnID, objectType, objectName,
&balance, &limit, holds );
```

Input

serverConnID passes the file server connection ID.

objectType passes the type of bindery object for which the request is being made. (See Appendix A, ªBindery Object Types.º)

objectName passes a pointer to the string containing the object name for which the account status request is being made.

Output

balance receives the number of value units available to the object to buy services on the network.

limit receives the value of the lowest level the object's account balance can reach before the object can no longer buy services on the network.

holds returns a list of objectIDs and holdAmounts that have been placed on the account (maximum = 16). (See Appendix A, ªNWHoldInfo_t Structure.º)

Return Values

0	Successful
-1	Unsuccessful

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

Description

This function queries a file server's bindery for the current account status of a specified bindery object by passing the bindery object name and type. The function returns the object's balance, limit and holds parameters.

The value in the balance parameter represents the object's account balance, usually in some established monetary unit such as cents.

The holds parameter lists servers that have issued NWSubmitAccountHold calls against the object and the amount reserved by each value-added server. The holds parameter also lists the object ID number of a value-added server that has issued a NWSubmitAccountHold call against the object. Up to 16 servers can place holds on the account at one time. Multiple holds from the same server are combined. Each server hold is made up of two fields: (1) the object ID of the server that placed the hold, and (2) the amount of that server's hold.

See Also

NWSubmitAccountHold

NWSubmitAccountCharge

This function updates the account of a bindery object by charging for a service and updating the audit record.

Synopsis

```
#include ªnwapi.hº

int           ccode;
uint16        serverConnID;
uint16        objectType;
char          objectName[NWMAX_OBJECT_NAME_LENGTH];
uint16        serviceType;
int32         chargeAmount;
int32         cancelHoldAmount;
uint16        commentType;
char          comment[NWMAX_COMMENT_LENGTH];

ccode=NWSubmitAccountCharge( serverConnID, objectType, objectName, serviceType, chargeAmount,
cancelHoldAmount, commentType, comment );
```

Input

serverConnID passes the file server connection ID.

objectType passes the type of bindery object for which the request is being made. (See Appendix A, ªBindery Object Types.º)

objectName passes a pointer to the name of the object for which the account status request is being made.

serviceType passes the type of service for which the request is being made (usually the object type of the charging account server).

chargeAmount passes the amount of account server charges for the service it provides.

cancelHoldAmount passes the amount to be subtracted from the total amount of all holds previously placed by the server. If no NWSubmitAccountHold calls were made prior to providing the service, this value should be zero.

commentType passes the type of comment written to the audit report.

comment passes a pointer to a comment associated with the object's account charge.

Output

None.

Return Values

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

0x94	No Write Privileges
0xA2	I/O Lock Error

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

Description

This function charges an object's account balance and relinquishes a hold against the object's account balance. The function can also write a note about the transaction in an audit record (optional). The charge and hold amounts do not have to be the same.

The objectType and objectName parameters must uniquely specify the bindery object and cannot contain wildcard characters.

The serviceType parameter usually contains the object type of the charging account server. The common server object types are listed below:

Object	**Type**
Archive Server	NWOT_ARCHIVE_SERVER
Job Server	NWOT_JOB_SERVER
Print Server	NWOT_PRINT_SERVER

The commentType parameter contains the number of the comment type in the comment parameter. Comment types are administered by Novell and are listed below:

Comment	**Description**
1	Connect time charge
2	Disk storage charge
3	Log in note
4	Log out note
5	Account locked note
6	Server time modified note

Developers should contact Novell for unique comment types. Comment types greater than 8000h are reserved for experimental purposes.

Notes

The comment parameter is the entry that the value-added server makes in an audit record. This audit record is contained in the
SYS:SYSTEM\NET$ACCT.DAT file.

See Also

NWSubmitAccountNote

NWSubmitAccountHold

This function reserves a specified amount of an object's account balance pending a NWSubmitAccountCharge call.

Synopsis

```
#include ªnwapi.hº

int          ccode;
uint16       serverConnID;
uint16       objectType;
char         objectName[NWMAX_OBJECT_NAME_LENGTH];
int32        reserveAmount;

ccode=NWSubmitAccountHold( serverConnID, objectType, objectName, reserveAmount );
```

Input

serverConnID passes the file server connection ID.

objectType passes the type of bindery object for which the request is being made. (See Appendix A, ªBindery Object Types.º)

objectName passes a pointer to the name of the bindery object for which the account status request is being made.

reserveAmount passes the hold amount to be placed against the client's account pending service.

Output

None.

Return Values

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

0x94	No Write Privileges
0xA2	I/O Lock Error
0xC1	No Account Balance
0xC3	Too Many Holds

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

Description

This function reserves a specified amount of an object's account balance before that object receives and is charged for a service on the network.

The objectType and objectName parameters must uniquely identify the bindery object and may not contain wildcard characters.

The reserveAmount parameter gets the amount that the server expects to charge for the service it is about to provide to the object.

Notes

No more than 16 servers can reserve amounts of an object's account balance at one time. Multiple holds from the same server are combined.

NWSubmitAccountNote

This function adds a note about an object's account to an audit record. This API does not charge for the service.

Synopsis

```
#include ªnwapi.hº

int             ccode;
uint16          serverConnID;
uint16          objectType;
char            objectName[NWMAX_OBJECT_NAME_LENGTH];
uint16          serviceType;
uint16          commentType;
char            comment[NWMAX_COMMENT_LENGTH];

ccode=NWSubmitAccountNote( serverConnID, objectType, objectName,
serviceType, commentType, comment );
```

Input

serverConnID passes the file server connection ID.

objectType passes the type of bindery object for which the request is being made. (See Appendix A, ªBindery Object Types.º)

objectName passes a pointer to the object name for which the account status request is being made.

serviceType passes the type of service for which the request is being made (usually the object type of the charging account server).

commentType passes the type of comment in the comment parameter. (See Appendix A, ªComment Types.º)

comment passes a pointer to the comment associated with the object's account.

Return Values

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

0xEA	No Such Member
0xEB	Not Set Property
0xEC	No Such Set
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

This function adds a note about an accounting transaction to an audit record.

The objectType and objectName parameters must uniquely identify the bindery object and may not contain wildcard characters.

The serviceType parameter usually contains the object type of the charging account server. The common server object types are listed below:

Object	**Type**

Archive Server	NWOT_ARCHIVE_SERVER
Job Server	NWOT_JOB_SERVER
Print Server	NWOT_PRINT_SERVER

The commentType parameter contains the number of the comment type in the comment parameter. Comment types are administered by Novell and are listed below:

Comment	**Description**
1	Connect time charge
2	Disk storage charge
3	Log in note
4	Log out note
5	Account locked note
6	Server time modified note

Developers should contact Novell for unique comment types. Comment types greater than 8000h are reserved for experimental purposes.

Notes

The comment parameter contains the entry that the server makes in the audit record. The audit record is contained in the SYS:SYSTEM\NET$ACCT.DAT file.