Appendix 4B
ConnMan Structures, Constants, and Definitions

Structures

CONN_RENAME

Used in the NESL EVENT_CONN_RENAMED event.

```
typedef struct _CONN_RENAME_ {
   CONN_HANDLE    currentHandle;
   CONN_HANDLE    oldHandle;
} CONN_RENAME;
```

CONN_ENTRY_RETURN_ALL

This structure is used to return (or set) connection information when **CONNGetStructure** or **CONNSetStructure** is called. It contains all the connection-related parameters that can be set or read.

```
typedef struct _CONN_INFO_TYPE_ {
    UINT32          connInfoVersion;
    UINT32          connReference;
    UINT32          connMaxDomainNameLen;
    SPECT_DATA      connDomainName;
    UINT32          connNameSvcId;
    UINT32          connSecurity;
    UINT32          connServerConnNum;
    UINT32          connAuthUserId;
    UINT32          connAuthState;
    UINT32          connMaxServerNameLen;
    SPECT_DATA      connServerName;
    TRAN_ADDR_TYPE      connTranAddr;
    UINT32          connMaxIo;
    UINT32          connLicense;
    UINT32          connMaxServiceNameLen;
    SPECT_DATA      connServiceName;
    UINT32          connRoundTrip;
    UINT32          connServerVersion;
} CONN_INFO_TYPE;
```

Definitions

```
#define  CONN_HANDLE    UINT32
#define  AUTH_HANDLE    UINT32
```

Provider IDs

Following are all defined IDs for name service providers, authentication service providers, transport providers, and session protocol providers. Note that each of these types includes a wildcard type. Defined IDs can be ORed with a wildcard or used alone. Two defined types (non-wildcard) may not be ORed. If ORing is used, the defined name is tried first, then all other providers are queried.

```
enum  {  NAME_SVC_NDS_ID      = 1,
         NAME_SVC_BINDERY_ID,
         NAME_SVC_PNW_ID,
         NAME_SVC_WILDCARD    = 0x8000 };

enum  {  AUTH_SVC_NONE        = 0,
         AUTH_SVC_BINDERY_ID,
         AUTH_SVC_NDS_ID,
         AUTH_SVC_PNW_ID,
         AUTH_SVC_WILDCARD    = 0x8000 };

enum  {  INVALID_SESSION_ID   = 0,
         NCP_SESSION_ID,
         SMB_SESSION_ID,
         WILD_SESSION_ID      = 0x8000 };

enum  {  TRAN_ID_IPX = 1,
         TRAN_ID_UDP,
         TRAN_ID_DDP,
         TRAN_ID_ASP,
         TRAN_ID_WILDCARD     = 0x8000 };
```

License States

These are the values used if Conn_entry_license is being set or retrieved via a **CONNSetValue** or **CONNGetValue.**

```
enum  {  LICENSE_STATE_OFF    = 0,
         LICENSE_STATE_ON,
         LICENSE_STATE_ON_FOR_HANDLE };

enum  {  NDS_NOT_CAPABLE      = 0,
         NDS_CAPABLE          };
```

Broadcast States

```
enum  {  BCAST_PERMIT_ALL     = 0,
         BCAST_PERMIT_SYSTEM,
         BCAST_PERMIT_NONE    };
```

NESL Events

```
#define EVENT_CONN_AUTHENTICATED "CONNECTION_AUTHENTICATED"
#define EVENT_CONN_CREATED       "CONNECTION_CREATED"
#define EVENT_CONN_DESTROYED     "CONNECTION_DESTROYED"
#define EVENT_CONN_LOGGED_OUT    "CONNECTION_LOGGED_OUT"
#define EVENT_CONN_PRE_CREATED   "CONN_PRE_CONNECTION_CREATED"
#define EVENT_CONN_PRE_DESTROYED "CONN_PRE_CONNECTION_DESTROYED"
#define EVENT_CONN_RECONNECTED   "CONNECTION_RECONNECTED"
#define EVENT_CONN_RENAMED       "CONNECTION_RENAMED"
#define EVENT_CONN_UNAUTHENTICATED  "CONNECTION_UNAUTHENTICATED"
```

Connection Handle Lookup Types

```
#define  MATCH_EQUALS      0x0000   // Lookup for equivalent info
#define  MATCH_NOT_EQUALS  0x0001   // Lookup for non-equivalent
                                      info
```

Connection Opening Flags

```
#define  SHORT_LIVED_CONNECTION  0x0000
#define  LONG_LIVED_CONNECTION   0x0001
```

Connection Validation Flags

```
#define  CONN_VALIDATE_HANDLE    0x0000   // Validate only the
                                             connHandle
#define  CONN_VALIDATE_SESSION   0x0001   // Validate to the
                                             session level
```

Connection Password Flags

```
#define  CONN_PASSWD_PROMPT_NONE       0x00000000
#define  CONN_PASSWD_PROMPT            0x00000001
#define  CONN_PASSWD_PROMPT_OLD        CONN_PASSWD_PROMPT
#define  CONN_PASSWD_PROMPT_NEW        0x00000002
#define  CONN_PASSWD_PROMPT_BOTH       (CONN_PASSWD_PROMPT |
                                       CONN_PASSWD_PROMPT_OLD)
```

Connection Security Flags

```
#define  SECURITY_SIGNING_NOT_IN_USE   0x00000000
#define  SECURITY_SIGNING_IN_USE       0x00000001
#define  SECURITY_LEVEL_CHECKSUM       0x00000100
```

```
#define  SECURITY_LEVEL_SIGN_HEADERS   0x00000200
#define  SECURITY_LEVEL_SIGN_ALL       0x00000400
#define  SECURITY_LEVEL_ENCRYPT        0x00000800
```

Object Type Defines

These defines are used by **CONNOpenByName** to define what the object type is.

```
#define  OBJECT_TYPE_NCP_SERVER              "File Server"
#define  OBJECT_TYPE_FILE_SERVER             "File Server"
#define  OBJECT_TYPE_AFP_SERVER              "AFP Server"
#define  OBJECT_TYPE_ALIAS                   "Alias"
#define  OBJECT_TYPE_BINDERY_OBJECT          "Bindery Object"
#define  OBJECT_TYPE_BINDERY_QUEUE           "Bindery Queue"
#define  OBJECT_TYPE_COMPUTER                "Computer"
#define  OBJECT_TYPE_COUNTRY                 "Country"
#define  OBJECT_TYPE_DEVICE                  "Device"
#define  OBJECT_TYPE_DIRECTORY_MAP           "Directory Map"
#define  OBJECT_TYPE_EXTERNAL_ENTITY         "External Entity"
#define  OBJECT_TYPE_GROUP                   "Group"
#define  OBJECT_TYPE_LIST                    "List"
#define  OBJECT_TYPE_LOCALITY                "Locality"
#define  OBJECT_TYPE_MESSAGE_ROUTING_GROUP   "Message Routing
                                             Group"
#define  OBJECT_TYPE_MESSAGING_SERVER        "Messaging Server"
#define  OBJECT_TYPE_ORGANIZATION            "Organization"
#define  OBJECT_TYPE_ORGANIZATIONAL_PERSON   "Organizational
                                             Person"
#define  OBJECT_TYPE_ORGANIZATIONAL_ROLE     "Organizational Role"
#define  OBJECT_TYPE_ORGANIZATIONAL_UNIT     "Organizational Unit"
#define  OBJECT_TYPE_PARTITION               "Partition"
#define  OBJECT_TYPE_PERSON                  "Person"
#define  OBJECT_TYPE_PRINT_SERVER            "Print Server"
#define  OBJECT_TYPE_PRINTER                 "Printer"
#define  OBJECT_TYPE_PROFILE                 "Profile"
#define  OBJECT_TYPE_QUEUE                   "Queue"
#define  OBJECT_TYPE_RESOURCE                "Resource"
#define  OBJECT_TYPE_SERVER                  "Server"
#define  OBJECT_TYPE_TOP                     "Top"
#define  OBJECT_TYPE_UNKNOWN                 "Unknown"
#define  OBJECT_TYPE_USER                    "User"
#define  OBJECT_TYPE_VOLUME                  "Volume"
```

Connection Entry Instance Equates

When a user calls **CONNGetStructure** or **CONNGetValue** (or the equivalent Set functions), these are the types that may be requested or set. This table shows what is returned in each case, and which NLMs may read and write these values.

			Return	Who Reads	Who Writes
#define	CONN_ENTRY_RETURN_NONE	0			
#define	CONN_ENTRY_VERSION	1	Value	global	Never
#define	CONN_ENTRY_AUTH_SVC_ID	2	AUTH_SVC_*	global	Auth Mux
#define	CONN_ENTRY_BROADCAST_STATE	3	BCAST_*	global	global
#define	CONN_ENTRY_REFERENCE	4	Value	global	Never
#define	CONN_ENTRY_DOMAIN_NAME	5	Struct	global	AuthMux
#define	CONN_ENTRY_WORKGROUP_ID	6	Struct	global	AuthMux
#define	CONN_ENTRY_SECURITY	7	Value	global	global
#define	CONN_ENTRY_SERVER_CONN_NUM	8	Value	global	SessMux
#define	CONN_ENTRY_AUTH_USER_ID	9	Value	global	AuthMux
#define	CONN_ENTRY_SERVER_NAME	10	Struct	global	SessMux
#define	CONN_ENTRY_TRAN_ADDR	11	Struct	global	SessMux
#define	CONN_ENTRY_NDS_ABILITY	12	NDS_*	global	SessMux
#define	CONN_ENTRY_MAX_IO	13	Value	global	SessMux
#define	CONN_ENTRY_LICENSE	14	LICENSE_STATE	global	global
#define	CONN_ENTRY_PUBLIC_STATE	15	Value	global	Never
#define	CONN_ENTRY_NAME_SVC_ID	16	NAME_SVC_*	global	SesMux NsMux
#define	CONN_ENTRY_ROUND_TRIP	17	Value	global	SessMux
#define	CONN_ENTRY_SERVER_VERSION	18	Value	global	SessMux
#define	CONN_ENTRY_TRAN_ADDR_OBJ	19	Value	global	SessMux
#define	CONN_ENTRY_SFT_LEVEL	20	Value	global	global
#define	CONN_ENTRY_TTS_LEVEL	21	Value	global	global
#define	CONN_ENTRY_SERVICE_NAME	22	Struct	global	AuthMux
#define	CONN_ENTRY_PERM	23	Flag	global	ConnMan sets
#define	CONN_ENTRY_AUTH	24	Flag	global	AuthMux sets
#define	CONN_ENTRY_ANCHOR	25	Flag	global	ConnMan sets
#define	CONN_ENTRY_SUSPENDED	26	Flag	global	ConnMan sets
#define	CONN_ENTRY_RESOURCE_COUNT	27	Value	none	global inc
#define	CONN_ENTRY_TRAN_SVC_ID	28	TRAN_SVC_*	global	SessMux
#define	CONN_ENTRY_AUTH_HANDLE	29	Value	global	AuthMux
#define	CONN_ENTRY_AUTH_SPEC_PTR	30	Value	AuthMux	AuthMux
#define	CONN_ENTRY_SESS_SVC_ID	31	Value	global	SessMux
#define	CONN_ENTRY_SESS_SPEC_PTR	32	Value	SessMux	SessMux
#define	CONN_ENTRY_ORDER_NUM	33	Value	global	SessMux
#define	CONN_ENTRY_MAX_RW_IO	34	Value	global	SessMux
#define	CONN_ENTRY_RETURN_ALL	65535	CONN_INFO_TYPE	global	none
#define	CONN_ENTRY_END_OF_TABLE	CONN_ENTRY_MAX_RW_IO		MAX VALUE	