

# Appendix 9B

## NDS Structures and Definitions

Manifest Constants .....	2
Name-Resolution-Specific Tag .....	2
Structure Definitions .....	2
AUTHEN_CACHE .....	2
AuthService .....	2
DNCSERVICE .....	2
NDS_RESOLVE_INFO .....	3
PreferredService .....	3
SECURE_NCP .....	3
TreeService .....	3
NDS VLM Compatability Structures .....	4
CLIENT32.H Structures Used by NDS .....	5
TRAN_ADDR_TYPE .....	5
SPECT_DATA .....	6

## Manifest Constants

### Name-Resolution-Specific Tag

```
#define CLIENT32_NAME_SVC_V1_00 0x00FE0001
```

## Structure Definitions

### AUTHEN\_CACHE

```
typedef struct _AUTHEN_CACHE_  
{  
    UINT16  totalSize;  
    UINT16  publicKeyLength;  
    UINT16  credentialLength;  
    UINT16  signatureLength;  
} AUTHEN_CACHE;
```

### AuthService

```
typedef struct  
{  
    Resource  rl;  
    UINT32   PG_Scope;  
    UINT32   P_Scope;  
    UINT32   treeHandle;  
    semHandle authenSem;  
    CONN_HANDLE monitorConn;  
    SECURE_NCP TDSSecureData;  
    AUTHEN_CACHE TDSSize;  
    VOID      *TDSData;  
    UINT32    monHandle;  
  
} AuthService;
```

### DNCSERVICE

```
typedef struct  
{  
    Resource  rl;  
    UINT32   PG_Scope;  
    UINT32   P_Scope;  
    UINT32   treeHandle;  
    UINT32   defNCType;  
    UINT32   defNCLength;  
    UINT8    defNC[MAX_DN_CHARS * 2];  
} DNCSERVICE;
```

**NDS\_RESOLVE\_INFO**

```
typedef struct _NDS_RESOLVE_INFO
{
    UINT32  tag;
    UINT32  flags;
    UINT32  reqFlags;
    UINT32  reqScope;
    UINT32  repResolveType;
    UINT32  repFlags;
    UINT32  resolvedOffset;
    UINT32  derefNameLength;
    UNICODE *derefName;

} NDS_RESOLVE_INFO;
```

**PreferredService**

```
typedef struct
{
    Resource rl;
    UINT32  PG_Scope;
    UINT32  P_Scope;
    UINT32  treeHandle;
} PreferredService;
```

**SECURE\_NCP**

```
typedef struct _SECURE_NCP
{
    UINT8  sessionKey[SESSION_KEY_SIZE];
    UINT8  challengeKey[CHALLENGE_KEY_SIZE];
} SECURE_NCP;
```

**TreeService**

```
typedef struct
{
    Resource rl;
    UINT8  treeName[NW_TREE_NAME_LENGTH];
    UINT32  LRUTimeStamp;
    UINT16  inUseCountHard;
    UINT16  inUseCountTask;
} TreeService;
```



## NDS VLM Compatability Structures

```
#define VLMS_LOGGING_IN      0x00000001
#define VLMS_LOGGING_OUT    0x00000002
#define VLMS_LOGGED_IN      0x00000004

typedef struct
{
    Resource    rl;
    UINT32     PG_Scope;
    UINT32     P_Scope;
    UINT32     status;
    UINT32     rwTDSCount;
    SECURE_NCP tdsSecureData;
    CONN_HANDLE monitorConn;
    UINT32     authHandle;
    UINT32     muxHandle;
} MonitorService;
```

## CLIENT32.H Structures Used by NDS

### TRAN\_ADDR\_TYPE

Describes the data structure for a transport address used by the name service APIs NDSResolveNameToAddress() and NDSResolveObjectToId().

```
typedef struct
{
    UINT32    transportType;
    UINT32    transportLen;
    UINT8     transportAddr [MAX_TRANSPORT_ADDRESS_LEN];
} TRAN_ADDR_TYPE;
```

#### Fields

<i>transportType</i>	Type of transport address. Currently defined values are:  TRAN_TYPE_IPX TRAN_TYPE_TCP TRAN_TYPE_WILD
<i>transportLen</i>	Length of returned transport address.
<i>transportAddr</i>	Buffer that contains transport address.

## **SPECT\_DATA**

Describes the data structure for specifying a string in either UNICODE or in local code page.

```
typedef struct SpectDataTag {
    UINT32      Length;
    UINT8 DIST  *Data;
    UINT32      DataType;
    UINT16      LocalCodePage;
    UINT16      CountryCode;
} SPECT_DATA
```