Appendix 7A
Name Service Multiplexor API

NSMEnumerateNameSvc

Description

Allows caller to discover the currently registered name service providers. Besides the unique ID of name service provider, this call also returns a copy of the name service provider's description block which provides additional information describing the name service provider.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMEnumberateNameSvc(
        UINT32      *enumHandle,
        UINT32      *nameSvcID,
        NAME_SVC_DESC_BLOCK           *nameSvcDescBlk )
```

Input

enumHandle	Handle to be used to retrieve the next name service provider. This value should initially be set to zero. The output of this service will be the next handle to use on subsequent calls to this function.

Output

nameSvcID	Pointer to receive the unique ID of the next registered name service provider.
nameSvcDescBlk	Pointer to data structure to receive a copy of the description block that this name service provider registered with the name service multiplexor. This parameter can be set to NULL if this information is not needed by caller.

Return values

SUCCESS_CODE
INVALID_PARAMETER
NO_MORE_ENTRIES

Remarks

This service will return **INVALID_PARAMETER** if *enumHandle* is invalid. If no more name service providers are registered (that is, they have all been scanned) then **NO_MORE_ENTRIES** is returned.

See also

NSMRegisterNameSvc
NSMUnregisterNameSvc

NSMGetPreferredName

Description

Returns the configured preferred name for the specified name service provider.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMGetPreferredName(
        UINT32          processGroupID,
        UINT32          processID,
        UINT32          nameSvcID,
        SPECT_DATA      *Data)
```

Input

processGroupID	ID for process group.
processID	ID for process.
nameSvcID	Unique ID of name service provider from which to get preferred name.
Data	Contains the size of the output buffer into which to receive the name.

Output

Data -> Data	Output buffer for name.
Data -> length	Number of bytes copied into the buffer. If the buffer is too small, then this value contains the size of buffer needed.

Return values

```
SUCCESS_CODE
    NAME_SVC_NOT_REGISTERED
    MORE_DATA_ERROR
```

Remarks

Input parameters *processGroupID* and *processID* are used to specify the scope of the preferred name to retrieve.

MORE_DATA_ERROR is returned if the caller's buffer is too small to receive the name.

See also

NSMSetPreferredName

NSMRegisterNameSvc

Description

Allows a name service provider such as BINDERY, NDS, or PNW to register its name service API support with the name service multiplexor.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMRegisterNameSvc(
        UINT32      nameSvcID,
        NAME_SVC_API_SET_TYPE           *apiSet,
        NAME_SVC_DESC_BLOCK                 *nameSvcDescBlk )
```

Input

nameSvcID	Unique ID of name service provider registering its services.
apiSet	Pointer to array of functions that a name service provider must implement to be name-service- compliant.
nameSvcDescBlk	Pointer to name service provider's description block which provides additional information describing this name service provider being registered.

Output

None.

Return values

SUCCESS_CODE
NAME_SVC_ALREADY_REGISTERED

Remarks

Once a name service provider has registered its API support, it may be called by the name service multiplexor in order to resolve name service requests.

See also

NSMUnregisterNameSvc

NSMResolveNameToAddress

Description

Resolves a user-readable name into a computer-usable network address.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMResolveNameToAddress(
        UINT32              processGroupID,
        UINT32              processID,
        CONN_HANDLE             connHandle,
        SPECT_DATA          *objectName,
        SPECT_DATA          *objectType,
        UINT32              transportType,
        UINT32              *nameSvcID,
        VOID                    *nameSvcSpec,
        UINT32              *repSessionSvcID,
        TRAN_ADDR_TYPE              *repTranAddr,
        UINT32              *repTranAddrCount)
```

Input

processGroupID	ID for process group.
processID	ID for process.
connHandle	Connection to use when resolving a name. For example, if the name is a bindery name, then the name service provider will scan the bindery of the given connection for the required address. This value can be NULL if the caller doesn't care which connection the name service providers use to resolve the name. If it is NULL, then the name service provider should use input parameters *processGroupID* and *processID* to see if a preferred name has been configured for that context, and use a connection to that preferred name to resolve name with.
objectName	Name to be resolved. The string must be NULL terminated and a maximum of 512 characters. If this string is Unicode, then the string has a maximum of 1024 bytes.

objectType	Specifies type of service required. For now, the only type that will be defined is "NCP_SERVER". In the future this could be expanded to include print servers, job servers, print queues, and others.
transportType	Specifies the preferred or required transport type. Must be one of the following: TRAN_TYPE_IPX TRAN_TYPE_TCP TRAN_TYPE_WILD TRAN_TYPE_WILD may be ORed with the other values or used alone. When ORed with another value, the wild value indicates an unspecified alternative is acceptable. When used alone it means any transport type is acceptable. Module should set this value to TRAN_TYPE_WILD to be transport independent.
nameSvcSpec	Points to name-service-specific information. See specific name service provider for details. This value should be NULL if input parameter *nameSvcID* is set to NAME_SVC_WILD.
repTranAddrCount	Number of TRAN_ADDR_TYPE array entries made available by output parameter *repTranAddr* for filling out by name service provider.

Output

repsessionSvcID	Unique ID of session protocol module to use to make connection with remote entity.
repTranAddr	Pointer to array of transport addresses that name service provider can fill in with transport addresses. There should be *repTranAddrCount* array entries. On input, parameter *repTranAddrCount* indicates how many array entries are available for the network service provider to fill out.
repTranAddrCount	Actual number of transport addresses being returned by name service provider for resolved name.

Return values

SUCCESS_CODE
NAME_SVC_NOT_REGISTERED
INVALID_PARAMETER
RESOLVE_NAME_FAILED
MORE_DATA_ERROR

Remarks

Name service providers will be enumerated to resolve the given name if the *nameSvcType* field of *name* is set to **NAME_SVC_WILD**; otherwise, the specified name service provider will be called to resolve the name.

It is possible for a name to be resolved to multiple transport addresses of different transport types. The caller must then decide which transport address to use, since this will determine which transport is used for communicating with the server.

MORE_DATA_ERROR is returned if the network service provider could have returned more transport addresses if the *repTranAddr* buffer space had been large enough to accomodate them all.

See also

NSMResoveObjectToID

NSMResolveObjectToID

Description

Resolves a user-readable NetWare name into a computer-usable ID and connection reference handle for use by requester modules.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMResolveObjectToID(
        UINT32              processGroupID,
        UINT32              processID,
        CONN_HANDLE             connHandle,
        SPECT_DATA          *objectName,
        SPECT_DATA          *objectType,
        UINT32              transportType,
        UINT32              *nameSvcType,
        VOID                    *nameSvcSpec,
        UINT32              *repObjectID,
        UINT32              *repSessionSvcID,
        TRAN_ADDR_TYPE              *repTranAddr,
        UINT32              *repTranAddrCount)
```

Input

processGroupID	ID for process group.
processID	ID for process.
connHandle	Connection to use when resolving object name. For example, if the name is a bindery name, then the name service provider will scan the bindery of the given connection for the given object name. This value can be NULL if the caller doesn't care which connection the name service providers use to resolve the name. If the value is NULL, then name service provider should use input parameters *processGroupID* and *processID* to see if a preferred name has been configured for that context, and use a connection to that preferred name to resolve name with.

objectName	Name to be resolved. The string must be NULL terminated and a maximum of 512 characters. If this string is Unicode, then the string has a maximum of 1024 bytes. (See the definition of OBJECT_SPECT_DATA for details.)
objectType	Type of object to resolve. *objectType->name* must point to one of the following strings: "USER" "GROUP" "QUEUE" "NCP_SERVER"
transportType	The preferred or required transport type. Must be one of the following: TRAN_TYPE_IPX TRAN_TYPE_IP TRAN_TYPE_WILD TRAN_TYPE_WILD may be ORed with the other values or used alone. When ORed with another valude, it indicates that an unspecified alternative is acceptable. When used alone it means any transport type is acceptable. To be transport independent, modules should set *transportType* to TRAN_TYPE_WILD.
namesvcID	Type of name being resolved. Must be one of the following: NAME_SVC_BINDERY_ID NAME_SVC_NDS_ID NAME_SVC_NDS_TREE_ID NAME_SVC_PNW_ID NAME_SV_WILD
nameSvcSpec	Points to name-service-specific information. See name service provider specification for details. This value should be NULL if the input parameter *nameSvcID* is set to NAME_SVC_WILD.

	repTranAddrCount	The number of TRAN_ADDR_TYPE array entries made available by output parameter *repTranAddr* for filling out by name service provider.
Output	*repobjectID*	Pointer used to store the identifier used by name service provider to identify object in its name space.
	repSessionSvcID	Unique ID of session protocol module to use to make connection with remote entity.
	repTranAddr	Pointer to array of transport addressses that name service provider can fill in with transport addresses. There should be *repTranAddrCount* array entries. On input, parameter *repTranAddrCount* indicates how many array entries are available for the network service provider to fill out.
	repTranAddrCount	Actual number of transport addressses being returned by name service provider for resolved name.

Return values

SUCCESS_CODE
INVALID_PARAMETER
RESOLVE_OBJECT_FAILED

Remarks

We currently do not support the returning of multiple connection/identifier pairs for an object name that is not unique in the resolved name space of a name service provider.

See also

NSMResolveNameToAddress

NSMSetPreferredName

Description

Sets the preferred name for the specified name service provider using the specified scope.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMSetPreferredName(
        UINT32              processGroupID,
        UINT32              processID,
        UINT32              nameSvcID,
        SPECT_DATA              *Data)
```

Input

processGroupID	ID for process group.
processID	ID for process.
nameSvcID	Unique ID of name service provider to set preferred name for.
Data	Preferred name being set.

Output

None.

Return values

```
SUCCESS_CODE
    NAME_SVC_NOT_REGISTERED
    INVALID_PARAMETER
```

Remarks

Input parameters *processGroupID* and *processID* are used to specify the scope of the preferred name to store.

A return code of **INVALID_PARAMETER** is returned if the name being set is too big for the name service provider it's being set for.

See also

NSMGetPreferredName

NSMUnregisterNameSvc

Description

Allows a name service provider to unregister its services from the name service multiplexor. A name service provider must make this call before being unloaded from the system.

Syntax

```
#include "name_svc.h"
    UINT32
    NSMUnregisterNameSvc(
        UINT32      nameSvcID )
```

Input

nameSvcID	Unique ID of name service provider being unregistered.

Output

None.

Return values

```
SUCCESS_CODE
    NAME_SVC_NOT_REGISTERED
```

See also

NSMRegisterNameSvc