

Data Set Name Functions

NWSM_DATA_SET_NAME	3-3
Get Data Set Name Functions	3-4
NWSMGetFirstName	3-4
NWSMGetNextName	3-5
NWSMCloseName	3-6
NWSMGetDataSetName	3-7
NWSMGetOneName	3-8
Put Data Set Name Functions	3-9
NWSMPutFirstName	3-9
NWSMPutNextName	3-11
NWSMPutOneName	3-13

The data set list functions parse or create a data set name list or selection list. The functions use the following definitions:

```
#define VALID      0x2AAAAAAAAAL
#define INVALID    0x155555555L
```

Context information for each data set name list or selection list is kept in a list handle. The handle is a UINT32 that contains the address of the name list structure, that uses the following data structure:

```
typedef struct
{
    UINT32      valid;
    BUFFERPTR   buffer;
    BUFFERPTR   bufferEnd;
    BUFFERPTR   ptr;
    UINT16      count;
    UINT16      index;
} NWSM_NAME;
```

valid contains two values, VALID or INVALID, that indicates if the structure is still allocated (VALID), or previously released (INVALID).

buffer points to a NWSM_SELECTION_LIST or a NWSM_DATA_SET_NAME_LIST. For more information about these structures, see *Target Service API*.

bufferEnd points to one byte beyond the buffer's end.

ptr points to the next name in the list.

count specifies the number of names in the list.

index indicates the number of names worked on (i.e., parsed or inserted).

NWSM_DATA_SET_NAME

If the data set name or selection list is parsed, information about the name is returned in the following structure:

```
typedef struct
{
    UINT32    nameSpaceType;
    UINT32    selectionType;
    UINT16    count;
    UINT16    HUGE *namePositions;
    UINT16    HUGE *separatorPositions;
    UINT16    nameLength;
    STRING    name;
} NWSM_DATA_SET_NAME;
```

For more information about the position values see *Target Service API*.

Get Data Set Name Functions

CCODE

NWSMGetFirstName

```
( void HUGE *buffer,
  NWSM_DATA_SET_NAME HUGE *name,
  UINT32 HUGE *handle);
```

Parameters

buffer	(INPUT) Points to a NWSM_DATA_SET_NAME_LIST or a NWSM_SELECTION_LIST structure.
name	(OUTPUT) Returns the name information. For more information see <i>Target Service API</i> .
handle	(OUTPUT) Points to a UINT32 variable. This handle is used for all subsequent get calls.

Completion Codes

0x0	Successful
0xFFFBFFFC	NWSMUT_NO_MORE_NAMES
0xFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFBFFF	NWSMUT_OUT_OF_MEMORY

Remarks

NWSMGetFirstName returns the first name contained in *buffer*. If the function fails for any reason or there are no more names in the list, *handle* is stamped invalid. *buffer* could contain a list of terminal path names, full path names, or a single TSA defined object.

Note: If NWSMUT_NO_MORE_NAMES is returned by **NWSMGetFirstName**, do not call **NWSMCloseName**.

Example

```
#include <smslib.h>

CCODE                                ccode;
void HUGE                            *buffer;
NWSM_DATA_SET_NAME                   name;
UINT32                               handle;

ccode = NWSMGetFirstName(buffer, (HUGE *)&name, (HUGE *)&handle);
```

See Also

NWSMGetNextName and NWSMCloseName

CCODE

NWSMGetNextName

```
( UINT32 HUGE *handle,
  NWSM_DATA_SET_NAME HUGE *name);
```

Parameters

handle	(INPUT) Passes the handle set by NWSMGetFirstName .
name	(OUTPUT) Returns the name information. For more information about NWSM_DATA_SET_NAME see <i>Target Service API</i> .

Completion Codes

0x0	Successful
0xFFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFFFBFFF	NWSMUT_INVALID_HANDLE
0xFFFFBFFFC	NWSMUT_NO_MORE_NAMES

Remarks

NWSMGetNextName returns the next name contained in *buffer*. The *handle* information is released and set to zero if NWSMUT_NO_MORE_NAMES is returned. *name* can be a terminal node name, a full path, or other types of names.

Note: If NWSMUT_NO_MORE_NAMES is returned by **NWSMGetNextName**, do not call **NWSMCloseName**.

Example

```
#include <smslib.h>

CCODE          ccode;
UINT32         handle;
NWSM_DATA_SET_NAME name;

...
if ((ccode = NWSMGetFirstName(buffer, &name, (HUGE *)&handle)) == 0)
{
    ...
    while ((ccode = NWSMGetNextName((HUGE *)&handle, (HUGE *)&name)) == 0)
    {
        ...
    }
}
```

See Also

NWSMGetFirstName and NWSMCloseName

CCODE

NWSMCloseName

(UIN32 HUGE *handle);

Parameters

handle	(INPUT) Passes the handle set by NWSMGetFirstName or NWSMPutFirstName .
--------	---

Completion Codes

0x0	Successful
0xFFFFBFFF	NWSMUT_INVALID_HANDLE

Remarks

NWSMCloseName releases the handle information and sets the handle structure to invalid and *handle* to zero.

Note: If NWSMUT_NO_MORE_NAME is returned by **NWSMGetFirstName** or **NWSMGetNextName**, do not call **NWSMCloseName**.

Example

```
#include <smslib.h>

CCODE      ccode;
UIN32      handle;

ccode = NWSMCloseName((UIN32 HUGE *)&handle);
```

See Also

NWSMGetFirstName
 NWSMGetNextName
 NWSMPutFirstName
 NWSMPutNextName

CCODE

NWSMGetDataSetName

```
( void HUGE *buffer,
  UINT32 nameSpaceType,
  NWSM_DATA_SET_NAME HUGE *name);
```

Parameters

buffer	(INPUT) Points to a NWSM_DATA_SET_NAME_LIST or NWSM_SELECTION_LIST structure.
nameSpaceType	(INPUT) Specifies the name space type to return.
name	(OUTPUT) Returns the name information. For more information about NWSM_DATA_SET_NAME see <i>Target Service API</i> .

Completion Codes

0x0	Successful
0xFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFFBFFFB	NWSMUT_OUT_OF_MEMORY
0xFFFBFFFC	NWSMUT_NO_MORE_NAMES

Remarks

NWSMGetDataSetName returns the first name with the specified *nameSpaceType*.

Note: Both structures, NWSM_DATA_SET_NAME_LIST and NWSM_SELECTION_LIST, are essentially the same, but use different names for the fields. The fields of both of these structures have the same offsets.

Example

```
#include <smslib.h>

CCODE          ccode;
void HUGE      *buffer;
UINT32         nameSpaceType;
NWSM_DATA_SET_NAME name;

ccode = NWSMGetDataSetName(buffer, nameSpaceType, (HUGE *) &name);
```

See Also

NWSMGetNextName and NWSMCloseName

CCODE

NWSMGetOneName

```
( void HUGE *buffer,
  NWSM_DATA_SET_NAME HUGE *name);
```

Parameters

buffer	(INPUT) Passes a pointer to NWSM_DATA_SET_NAME_LIST or NWSM_SELECTION_LIST.
name	(OUTPUT) Passes a pointer to a structure and returns the name information. For more information about NWSM_DATA_SET_NAME, see <i>Target Service API</i> .

Completion Codes

0x0	Successful
0xFFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFFFBFFFB	NWSMUT_OUT_OF_MEMORY
0xFFFFBFFFC	NWSMUT_NO_MORE_NAMES

Remarks

NWSMGetOneName returns the first name in the list.
NWSMCloseName should not be called, because no handle is required.

Example

```
#include <smslib.h>

CCODE          ccode;
void HUGE      *buffer;
NWSM_DATA_SET_NAME name;

ccode = NWSMGetOneName (buffer, (HUGE *) &name);
```


Put Data Set Name Functions

CCODE

NWSMPutFirstName

```
( void HUGE **buffer,
  UINT32 nameSpaceType,
  UINT32 selectionType,
  NWBOOLEAN reverseOrder,
  STRING sep1,
  STRING sep2,
  STRING name,
  UINT32 HUGE *handle);
```

Parameters

buffer	(INPUT) Passes a pointer to a NWSM_DATA_SET_NAME_LIST or NWSM_SELECTION_LIST structure. If there is not enough space for the name, the buffer is resized. If <i>buffer</i> is set to NULL, memory is allocated for it.
nameSpaceType	(INPUT) Passes the data set's name space type.
selectionType	(INPUT) Passes the data set's selection type. If a data set name is being passed, set <i>selectionType</i> to zero (0). If <i>name</i> is part of a skipped data set list, <i>selectionType</i> should contain the completion code that describes why the data set was skipped. For more information about <i>selectionType</i> , see <i>Target Service API</i> .
reverseOrder	(INPUT) Passes the flag returned by NWSMTSGetNameSpaceTypeInfo (For more information about reverse order, see <i>Target Service API</i>).
sep1	(INPUT) Passes the first or volume separator. If the name space is Mac or other ¹ , then it is ":" or "." respectively. This information is returned by NWSMTSGetNameSpaceTypeInfo (see <i>Target Service API</i>).
sep2	(INPUT) Passes the second or non-terminal/terminal separator. If the name space is Mac or other, then it is "/" or "\" respectively. This information is returned by NWSMTSGetNameSpaceTypeInfo (see <i>Target Service API</i>).
name	(INPUT) Passes the data set's name (e.g., a full or terminal path). <i>name</i> 's usage (i.e., used a exclude or include list) is determined by <i>selectionType</i> (for more information about NWSM_DATA_SET_NAME, see <i>Target Service API</i>).
handle	(OUTPUT) Passes a pointer to a UINT32 variable and returns a handle to the list. The handle is used for all subsequent calls.

¹ DOS, Unix, FTAM, etc.

Completion Codes

0x0	Successful
0xFFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFFFBFFF	NWSMUT_OUT_OF_MEMORY

Remarks

NWSMPutFirstName puts the name information into the beginning of a `NWSM_DATA_SET_NAME_LIST` or a `NWSM_SELECTION_LIST` structure, sets the buffer size of the list, and sets the number of list entries to one. If there is not enough room in the buffer for it, the buffer's size is increased. This function allocates a memory for buffer, if it is set to `NULL`.

The information for *nameSpaceType*, *reverseOrder*, *sep1*, and *sep2* can be retrieved with the TS API function **NWSMTSGetNameSpaceTypeInfo**.

Note: Both `NWSM_SELECTION_LIST` and `NWSM_DATA_SET_NAME_LIST` are essentially the same. That is why there is no need to indicate the list's type.

NWSMPutFirstName can build a selection list for **NWSMTSScanDataSetBegin**. If an `NWSM_DATA_SET_NAME_LIST` is being built, *selectionType* is put into the reserved field of the list structure. If a list of skipped data is being built, skipped because of an error, *selectionType* should contain the error value.

Example

```
#include <smslib.h>

CCODE      ccode;
void HUGE  *buffer;
UINT32     nameSpaceType;
UINT32     selectionType;
STRING     sep1;
STRING     sep2;
STRING     name;
UINT32     handle;

ccode = NWSMPutFirstName(&buffer, nameSpaceType, selectionType, revserOrder,
    sep1, sep2, name, (HUGE *)&handle);
```

See Also

`NWSMPutNextName`
`NWSMCloseName`

CCODE

NWSMPutNextName

```
( void HUGE **buffer,
  UINT32 HUGE *handle,
  UINT32 nameSpaceType,
  UINT32 selectionType,
  NWBOOLEAN reverseOrder,
  STRING sep1,
  STRING sep2,
  STRING name);
```

Parameters

buffer	(INPUT) Passes a pointer to a NWSM_DATA_SET_NAME_LIST or NWSM_SELECTION_LIST structure. If there is not enough space for the name, <i>buffer</i> is resized.
handle	(INPUT) Passes the handle returned by NWSMPutFirstName .
nameSpaceType	(INPUT) Passes the name space type of <i>name</i> .
selectionType	(INPUT) Passes the data set's selection type. For more information about <i>selectionType</i> , see <i>Target Service API</i> .
reverseOrder	(INPUT) Passes the flag returned by NWSMTSGetNameSpaceTypeInfo (For more information about reverse order, see <i>Target Service API</i>).
sep1	(INPUT) Passes the first or volume separator. If the name space is Mac or other ² , then it is "::" or ":" respectively.
sep2	(INPUT) Passes the second or non-terminal/terminal separator. If the name space is Mac or other, then it is "." or "\" respectively.
name	(INPUT) Passes the data set's name. For more information about NWSM_DATA_SET_NAME, see <i>Target Service API</i> .

Completion Codes

0x0	Successful
0xFFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFFFBFFF	NWSMUT_OUT_OF_MEMORY

Remarks

NWSMPutNextName appends the name information to a NWSM_DATA_SET_NAME_LIST or a NWSM_SELECTION_LIST and increments the number of list entries by one. If there is not enough room in the buffer for it, *buffer* is resized.

² DOS, Unix, FTAM, etc.

Example

```
#include <smslib.h>

CCODE      ccode;
void HUGE  *buffer;
UINT32     handle;
UINT32     nameSpaceType;
UINT32     selectionType;
STRING     sep1;
STRING     sep2;
STRING     name;

ccode = NWSMPutNextName(&buffer, (HUGE*)&handle, nameSpaceType, selectionType,
                        reverseOrder, sep1, sep2, name);
```

See Also

NWSMPutFirstName
NWSMCloseName

CCODE

NWSMPutOneName

```
( void HUGE **buffer,
  UINT32 nameSpaceType,
  UINT32 selectionType,
  NWSMBOOLEAN reverseOrder,
  STRING sep1,
  STRING sep2,
  STRING name);
```

Parameters

buffer	(INPUT) Passes a pointer to a NWSM_DATA_SET_NAME_LIST or NWSM_SELECTION_LIST structure. If there is not enough space for <i>name</i> , <i>buffer</i> is resized. If <i>buffer</i> is NULL, memory is allocated for it.
nameSpaceType	(INPUT) Passes the data set's name space type.
selectionType	(INPUT) Passes the desired selection.
reverseOrder	(INPUT) Passes the flag returned by NWSMTSGetNameSpaceTypeInfo (For more information about reverse order, see <i>Target Service API</i>).
sep1	(INPUT) Passes the first or volume separator. If the name space is Mac or other, then it is ":" or "." respectively. This information is returned by NWSMTSGetNameSpaceTypeInfo (see <i>Target Service API</i>).
sep2	(INPUT) Passes the second or non-terminal/terminal separator. If the name space is Mac or other, then it is ":" or "\" respectively. This information is returned by NWSMTSGetNameSpaceTypeInfo (see <i>Target Service API</i>).
name	(INPUT) Passes the data set's name. For more information about NWSM_DATA_SET_NAME, see <i>Target Service API</i> .

Completion Codes

0x0	Successful
0xFFFBFFFB	NWSMUT_OUT_OF_MEMORY
0xFFFBFFFD	NWSMUT_INVALID_PARAMETER
0xFFBFFFFF	NWSMUT_INVALID_HANDLE

Remarks

NWSMPutOneName appends one name to an NWSM_SELECTION_LIST or an NWSM_DATA_SET_NAME_LIST, and increments the number of list entries by one. Do not call **NWSMCloseName**, because no handle is required. This function can build the *resourceName* parameter for **NWSMTSScanDataSetBegin** (see *Target Service API*).

The information for *nameSpaceType*, *reverseOrder*, *sep1*, and *sep2* can be retrieved with the TS API function **NWSMTSGetNameSpaceTypeInfo**.

Set *nameSpaceType* to **NWSM_TSA_DEFINED_RESOURCE** only if *name* was returned by **NWSMTSListTSResources**.

Example

```
#include <smslib.h>

CCODE      ccode;
void HUGE  *buffer;
UINT32     nameSpaceType;
UINT32     selectionType;
STRING     sep1;
STRING     sep2;
STRING     name;

ccode = NWSMPutOneName(&buffer, nameSpaceType, selectionType, reverseOrder, sep1,
                      sep2, name);
```