

# Target Service API

April 27, 1993  
Rev. 2.0

Novell®, Inc.  
122 East 1700 South  
P.O. Box 5900  
Provo, Utah 84606

# Copyright

©February 1993 Novell, Inc. All Rights Reserved

This work is subject to U.S. and international copyright laws and treaties. No part of this work may be used, practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## Trademarks

**Novell** is a registered trademark of Novell, Inc.

**NetWare** is a registered trademark of Novell, Inc.

# Table Of Contents

Introduction	1-1
Term Definitions	1-3
TSA Options	1-3
TSA-Specific Resources	1-4
TSA-Specific Options	1-5
Scan Type Options	1-5
Selection Type Options	1-7
Open Mode Options	1-8
Scanning	1-9
Scanning Order	1-10
Scanning Precedence	1-11
Call Sequence	1-11
Connection Establishment	1-11
Back Up Options	1-12
Back Up	1-13
Restore Options	1-14
Restore	1-15
Connection Termination	1-16
Miscellaneous Functions	1-16
Call Summary	1-17
Function Descriptions	2-1
Connectivity Functions	2-2
NWSMListTSAs - Data Requestor Fn	2-2
NWSMConnectToTSA - Data Requestor Fn	2-4
NWSMTSScanTargetServiceName	2-5
NWSMTSListTargetServices	2-6
NWSMTSGetTargetServiceAddress	2-8
NWSMTSGetTargetServiceType	2-9
NWSMTSConnectToTargetService	2-10
TSA Option Functions	2-12
NWSMTSGetUnsupportedOptions	2-12
NWSMTSScanTargetServiceResource	2-14
NWSMTSListTSResources	2-16
NWSMTSBuildResourceList	2-18
NWSMTSGetTargetResourceInfo	2-19
NWSMTSGetTargetScanTypeString	2-21
NWSMTSGetTargetSelectionTypeStr	2-23
NWSMTSScanSupportedNameSpaces	2-25
NWSMTSListSupportedNameSpaces	2-27
NWSMTSGetNameSpaceTypeInfo	2-29
NWSMTSGetOpenModeOptionString	2-31
Back Up Functions	2-34
NWSMTSScanDataSetBegin	2-34
NWSMTSOpenDataSetForBackup	2-41
NWSMTSReadDataSet	2-44
NWSMTSScanNextDataSet	2-46
NWSMTSScanDataSetEnd	2-49
NWSMTSReturnToParent	2-51

NWSMTRSrenameDataSet . . . . .	2-53
NWSMTRSdeleteDataSet . . . . .	2-55
NWSMTSSetArchiveStatus . . . . .	2-56
Restore Functions . . . . .	2-57
NWSMTSSetRestoreOptions . . . . .	2-57
NWSMTRIsDataSetExcluded . . . . .	2-59
NWSMTROpenDataSetForRestore . . . . .	2-61
NWSMTRWriteDataSet . . . . .	2-69
Connection Termination Functions . . . . .	2-72
NWSMTRReleaseTargetService . . . . .	2-72
NWSMTRReleaseTSA . . . . .	2-73
Miscellaneous Functions . . . . .	2-74
NWSMTRCloseDataSet . . . . .	2-74
NWSMTRFreeNameList - A Utility Fn . . . . .	2-75
NWSMTRCatDataSetName . . . . .	2-76
NWSMTRParseDataSetName . . . . .	2-77
NWSMTRSeparateDataSetName . . . . .	2-79
Other Documents . . . . .	A-1
SMS Documents . . . . .	A-2
Miscellaneous Novell Document . . . . .	A-2
Data Structure Description . . . . .	B-1
NWSM_NAME_LIST . . . . .	B-2
Scanning Structures . . . . .	B-2
NWSM_SCAN_CONTROL . . . . .	B-3
Zero Date and Time Fields . . . . .	B-6
Scanning Settings . . . . .	B-6
NWSM_SELECTION_LIST . . . . .	B-11
NWSM_SCAN_INFORMATION . . . . .	B-18
NWSM_DATA_SET_NAME_LIST . . . . .	B-19
NWSM_VERSION_INFORMATION . . . . .	B-23
Completion Codes . . . . .	C-1
Alphabetized . . . . .	C-2
Numerical Ordered . . . . .	C-7
Revision History . . . . .	D-1
Glossary . . . . .	GL-1
Index . . . . .	Index-1

# Introduction

Term Definitions .....	1-3
TSA Options .....	1-3
TSA-Specific Resources .....	1-4
TSA-Specific Options .....	1-5
Scan Type Options .....	1-5
Selection Type Options .....	1-7
Open Mode Options .....	1-8
Scanning .....	1-9
Scanning Order .....	1-10
Scanning Precedence .....	1-11
Call Sequence .....	1-11
Connection Establishment .....	1-11
Back Up Options .....	1-12
Back Up .....	1-13
Restore Options .....	1-14
Restore .....	1-15
Connection Termination .....	1-16
Miscellaneous Functions .....	1-16
Call Summary .....	1-17

The Target Service (TS) API document is for those writing to the API and those writing a TSA. The document assumes the reader is familiar with the System Independent Data Format (SIDF [see *System Independent Data Format* for more information]). It presents:

- Detailed concept and data flows
- An explanation of the API required by the TSA
- TSA's responsibilities for System Independent Data Format (SIDF).

This document groups the functions according to functionality (e.g., options, back up, etc.) and within these groups, present the calls in order of usage (e.g., open data set, close data set, etc.). Other APIs used with the TS API are the Data Requestor API and SMS Utilities API.

The document uses the following conventions:

**ALL CAPITALS** All references to "#defines" and structures are capitalized.

**bold** All function names within sentences are bold.

**Fn** Abbreviation for function.

*italics* All parameter names within sentences are italicized.

**NWSM** All DR API calls and structures in this document have this prefix. It means NetWare Storage Management.

**NWSMDR** All errors returned by the SMDR have a NWSMDR prefix.

**NWSMTS** All TSA function calls, structures, and errors in this document have this prefix. It means NetWare Storage Management Target Service.

**NWSMUT** All errors returned by the utility functions have a NWSMUT prefix.

**passes** All parameters described in the API description chapter are said to pass a value if an application transfers data to

---

	an API function.
return	All parameters described in the API description chapter are said to return a value if the API function returns a value(s).
<u>underline</u>	All references to SMS terms are underlined.

## Term Definitions

Engine	The SMS engine (e.g., SBackup).
Data Set Data	Contains the data set's data, path information, attributes, etc. Under SMS, this is the target's data formatted by the TSA. See <i>System Independent Data Format</i> for more information.
Data Set Information	Information needed to help an application (e.g., SME). For instance, this information can be the data set's name, creation date, etc. An application can use this to display the data set that is being serviced. Under SMS, this is the scan and scan control information. See <i>System Independent Data Format</i> for more information.
Exclude	To skip a data set(s) from a back up or restore session.
Include	To admit a data set(s) into a back up or restore session.
Parent(s)	An object that can have siblings. For example, directories and subdirectories are parents.
Child	An object that cannot have siblings. For example, all files are children, however, subdirectories are not children.
Sibling	A parent or child.

## TSA Options

The Target Service Agent (TSA) is a target-specific process that reads from and writes to the target. The Target Service API (TS API), and TSA-specific-user-selectable options and

resources drive the TSA. Since all target's are different, the options and resources allowed by each TSA differs. This difference is the focus of this section.

Targets differ in the type of data sets they contain and the method used to scan (search) for them. For example, one target has a flat file system, while another has a hierarchical file system. When scanning for data sets on these targets, the option to exclude files under a directory does not exist under a flat file system.

Targets also differ in resources. Resources are volume, drives, directories, etc. For example NetWare's resources include file servers, volumes, and drives, while DOS' resources include drives (for more information about resources, see "TSA-Specific Resources" in this chapter).

Since the target's TSA reflects these previously mentioned differences, each TSA will provides different options (e.g., traverse the file system, exclude all data bases, or backup a file service). Since the engine should not know these differences, SMS provides a way for each TSA to express the option and resources available to the user. Each TSA defines a list of strings that describe its option and resources. The SME retrieves these strings, presents them to the user for selection, and then notifies the TSA of the user's selections when scanning is initiated. Sections "TSA-Specific Resources" and "TSA-Specific Options" further discusses these resources and options. Other options that are common to most TSAs are discussed in "NWSM\_SCAN\_CONTROL" of Appendix B.

## **TSA-Specific Resources**

The TSA views its target's data as primary resources and secondary resources. The primary resource is like the root(s) of the target. It serves as a starting point to get at the data sets (under NetWare there are, in a sense, two root levels: the file server and its volumes). The secondary resources are the siblings of the primary resource(s). Table 1-1 shows some examples of primary and secondary resources.



**Table 1-1.** Resources

<b>Target</b>	<b>Primary Resource(s)</b>	<b>Secondary Resources</b>
NetWare	File Servers, Volumes, and Bindery	Directories and files
DOS Workstation	Drives	Directories and files
Btrieve	Database	None

Notice that although Btrieve's database may consist of many files, they are backed up as one data set, and therefore, it has no siblings. When asked for a list of primary resources, the TSA may return the following:

```
"File Server"
"Bindery"
"Sys"
"Vol 1"
```

Usually, the primary resource list is presented to the user for selection. Selection of the first resource tells the TSA to service the entire target. Selection of other resources tells the TSA to service only that part of the target. To get the strings that represent the TSA's primary resources, call **NWSMTSScanTargetServiceResource** or **NWSMTSListTSResources**. Secondary resources are returned while scanning the target for data sets (i.e., when calling **NWSMTSScanDataSetBegin**, **NWSMTSScanNextDataSet**, and **NWSMTSScanDataSetEnd**). To specify the resource to scan call **NWSMTSScanDataSetBegin**.

## TSA-Specific Options

TSA options are like filters that specify which data sets to retrieve or leave alone, and are user selectable. There are three TSA-specific option groups: the scan type, the selection type, and the open mode. These are further discussed below.

### Scan Type Options

A scan type is a TSA defined action used against a scan to filter in/out particular TSA-defined data sets throughout a session. For example, one action is to exclude hidden data sets from a back up. The action is "exclude from back up" and the TSA defined data set type is "hidden." The following list shows the scan types used by the NetWare 4.0 TSA:

- Do not traverse
- Exclude archived children
- Exclude hidden parents
- Exclude system children
- Exclude system parents
- Exclude child trustees
  
- Exclude parent trustees
- Exclude access database
- Exclude volume restrictions
- Exclude disk space restrictions
- Exclude extended attributes
- Exclude data streams

If an option is chosen from the preceding list, it means to exclude that type of data set. If the same option is not chosen, it means to include that type of data set. There are two scan types: the predefined and the TSA-specific scan types. Predefined scan types are common to most TSAs, but a TSA does not have to support it. However, every TSA must acknowledge whether it supports a predefined scan type or not (see **NWSMTSGetTargetScanTypeString**). The preceding list is a group of predefined scan types. TSA-specific scan types are those defined by the TSA developer.

**Note:** There is a maximum of 32 scan types per TSA (predefined-TSA options and TSA-specific options).

As mentioned before, strings represent a TSA's options. The call that retrieves the strings that represent the TSA's scan types is **NWSMTSGetTargetScanTypeString**. The order of these strings corresponds to a scan type bit map. That is, string one represents bit one, string two represents bit two, etc. When called, the function returns the strings representing the predefined scan types in the order shown in the preceding list. The strings representing the TSA-specific scan type follow these strings.

To notify the TSA of the user's choice of scan types, set or unset the appropriate bits of the bit map and send it to the TSA. The bit map is *scanType* of the `NWSM_SCAN_CONTROL` structure. To send the bit map call **NWSMTSScanDataSetBegin**.

### Selection Type Options

A selection type is a TSA defined action used against a scan to filter in/out particular user-specified data sets throughout a session. For example, one action is to exclude all "\*.exe" files. The action is "exclude" and the user-specified data sets are "\*.exe." Selection types used by the NetWare 4.0 TSA are:

**Table 1-2.** Selection Types

Exclude TSA defined resource
Include TSA defined resource
Exclude parent
Include parent
Exclude child
Include child
Exclude child by full name
Include child by full name

There are two selection types: the predefined and the TSA-specific selection types. Predefined selection types are common to most TSAs, but a TSA does not have to support it. However, every TSA must acknowledge whether it supports a predefined-selection type or not. The TSA developer defines the TSA-specific selection types.

As mentioned before, strings represent the TSA's options. The call that retrieves these strings is **NWSMTSGetTargetSelectionTypeStr**. Unlike the scan type, where each string corresponds to a bit in a bit map, one selection type string represents two bits: bit zero and bit x (where x is from 1 through 31). Bit zero represents one TSA action when the bit is off and another when the bit is on (the TSA defines the action). Bits 1 through 31 represent the type of data sets to apply the action against. Therefore, there is a maximum of 62 selection types.

**Note:** Selection types cannot be ORed together.

For example, in the NetWare 4.0 TSA, bit zero represents "exclude a data set" when turned off and "include a data set" when turned on. Predefined bits one through four represent TSA defined resources (TSA-specific resources), parents, children, and children by full name respectively. When bit zero and bits one through four are combined, they produce Table 1-2.

To notify the TSA of the user's selection(s), set or unset the corresponding bits of *selectionType* in the `NWSM_SELECTION_LIST` structure and pass the user-specified data sets into an `NWSM_SELECTION_LIST` structure when starting the scan (see **NWSMTSScanDataSetBegin**).

Now to complete the example given at the beginning of this section. To exclude all executable files, clear bit zero and set bit 3 of *selectionType*, and place "\*.exe" in the data set name list of the `NWSM_SELECTION_LIST` structure. To exclude all executable in a particular directory, the user includes the path to the directory in the string "\*.exe", however bit 4 instead of 3 must be set.

### Open Mode Options

Open mode options determine how a data set is opened during a back up or restore session. Unlike the other two option groups where they are applied to all data sets, the open mode options are applied to one data set. There are three groups of open mode options as shown below:

- numeric open mode options
- non-numeric open mode options
- TSA-specific non-numeric open mode options.

The numeric options have numerical values that are sequenced (as opposed to bit mapped) such as 0, 1, 2, 3, and 4; hence the name numeric. These options cannot be ORed with another numeric option, consequently only one option from this group can be chosen. However, they can be ORed with one or more non-numeric options. The following lists show the numeric open mode options for back up and restore:

Back up numeric open modes:

<code>NWSM_USE_LOCK_MODE_IF_DW_FAILS</code>	0x1
<code>NWSM_NO_LOCK_NO_PROTECTION</code>	0x2
<code>NWSM_OPEN_READ_ONLY</code>	0x3

Restore numeric open modes:

<code>NWSM_OVERWRITE_DATA_SET</code>	0x1
<code>NWSM_DO_NOT_OVERWRITE_DATA_SET</code>	0x2
<code>NWSM_CREATE_PARENT_HANDLE</code>	0x3
<code>NWSM_UPDATE_DATA_SET</code>	0x4

Non-numeric open mode options are bit mapped (i.e., a single bit represents the option), consequently, one or more options can be chosen from this group by ORing them together. The following lists show the non-numeric open mode options for

back up and restore:

Restore non-numeric open modes:

NWSM_CLEAR_MODIFY_FLAG_RESTORE	0x40
NWSM_RESTORE_MODIFY_FLAG	0x80

TSA-specific non-numeric open mode options:

NWSM_NO_DATA_STREAMS	0x100
NWSM_NO_EXTENDED_ATTRIBUTES	0x200
NWSM_NO_PARENT_TRUSTEES	0x400
NWSM_NO_CHILD_TRUSTEES	0x800
NWSM_NO_VOLUME_RESTRICTIONS	0x1000
NWSM_NO_DISK_SPACE_RESTRICTIONS	0x2000

**Note:** At times, the back up session and restore session may not share the same TSA-specific non-numeric options.

Three calls deal with the open mode options:

**NWSMTSGetOpenModeOptionString**,  
**NWSMTSOpenDataSetForBackup**, and  
**NWSMTSOpenDataSetForRestore** (see these functions for more information).

## Scanning

This section covers the scanning process, scanning order, and scanning precedence. Other options affecting the scanning process are discussed under "Scanning Settings" in Appendix B.

After the scan control, selection list, data set name lists structures are set (see "TSA Options" of this chapter), the engine can start the scanning (searching) process to find the data sets to back up. The following functions are used to scan the target:

NWSMTSScanDataSetBegin  
 NWSMTSScanNextDataSet  
 NWSMTSScanDataSetEnd

**NWSMTSScanDataSetBegin** starts the scanning process. The starting point for the scan is the resource name given to this function. The function returns either the parent of a data set or a data set (see "Scanning Settings" in Appendix B.) that was specified by the scan type options and selection type options.

**NWSMTSScanNextDataSet** continues the scanning process. This function is called repeatedly until all data sets are found or an error occurs. **NWSMTSScanDataSetEnd** terminates

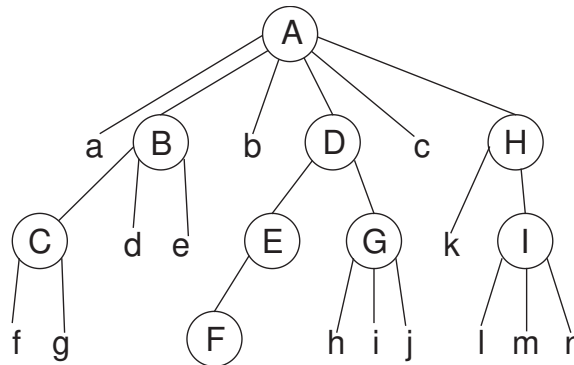
the scan started by **NWSMTSScanDataSetBegin**.

Multiple scans can be initiated by calling **NWSMTSScanDataSetBegin** once for each resource name. If multiple scans are used, make sure that the data area covered by a resource does not overlap the area covered by another resource. Resources that overlap each other will cause a duplication of effort. Each scan started must be terminated by a scanning error or by calling **NWSMTSScanDataSetEnd**.

**Note:** The scanning functions can also be used to outside the context of a back up session. They can also be used to see if a data set exists.

## Scanning Order

The scanning order followed by the scan functions is as follows. Scan all the children of parent1, then if parent1 contains a parent(s), get a parent and scan it. All the children of the next parent are scanned first, then the parent(s) it contains. If there are no parents, the scan walks up one level of the file system and starts scanning the next parent in parent1. This is pattern is followed recursively until all data sets under a resource are scanned. Figure 1-2 shows an example of the scanning order.



Scanning Order: AabcBdeCfgDEFGhijHkllmn

**Figure 1-2.** Scanning Order

Traversal of the target's file system can be modified by the exclusion of data sets specified in the scan type and selection type options (see "Scan Type Options" and "Selection Type Options" in this chapter). `NWSM_SCAN_CONTROL` provides an option to prevent traversal of the target's file system (See "NWSM\_SCAN\_CONTROL" in Appendix B).

## Scanning Precedence

The algorithm used by the scan functions and algorithms for the restore functions follow the precedence below (the upper action has a higher precedence than the lower action):

- Exclude a data set
- Include a data set
- Other reading or writing options (e.g.,  
NWSM\_DO\_NOT\_OVERWRITE\_DATA\_SET<sup>1</sup>)

For example, during a restore session, if a data set is included for restoration, and the same data set is marked as NWSM\_DO\_NOT\_OVERWRITE\_DATA\_SET, the TSA will restore the data set even if it does exist. To prevent the data set from being overwritten, do not include the data set.

## Call Sequence

This section shows a simplified example of how the TS functions are used. Notice that there are no local or remote sensitive contexts. This is handled automatically by the Novell provided Storage Management Data Requestor (SMDR). We will assume that a connection to SDI already exists (see *Storage Device API* for more information).

## Connection Establishment

Connecting the SME to a target involves finding and connecting to the TSA and the target. The following steps illustrate this.

1. The SME first asks SMS for a list of SMDRs by using:

NWSMListSMDRs

Using this call first and then the call below, can be faster than just calling NWSMListTSAs.

**Note:** See *Storage Management Services Utilities Library* for a description of this call.

2. Next, the SME chooses a SMDR and asks for TSAs of a particular type (e.g., DOS, NetWare, etc.) by using:

NWSMListTSAs

---

<sup>1</sup> A restore option

3. Upon selecting a TSA, the SME connects to it through:

NWSMConnectToTSA

4. All targets available through the TSA are found by calling:

NWSMTSListTargetServices

5. After selecting a target, connect to it by using:

NWSMTSConnectToTargetService

6. Optionally, you may want information about the target service by calling:

NWSMTSGetTargetServiceType

## Back Up Options

The next phase involves building the scan control options, gathering the TSA-specific options and resources, allowing the user to choose one or more of these options, and notifying the TSA of the selected options (see "TSA Issues" in this chapter for more information). The following steps illustrate this (there is no set order for gathering the options):

1. Before building the scan control options list, see "NWSM\_SCAN\_CONTROL" in Appendix "Data Structures Description" to see what can be in this list.

2. To build the scan type options list use:

NWSMTSGetTargetScanTypeString

3. To build the selection list options, use:

NWSMTSGetTargetSelectionTypeStr

4. To get the resources (i.e., root[s] of a target) use:

NWSMTSScanTargetServiceResource

**Note:** For more information about resources, see "TSA-Specific Resources" in this chapter.

During this step, you may get the target's supported name spaces and information about them by using:

NWSMTSScanSupportedNameSpaces



and

NWSMTSGetNameSpaceTypeInfo

5. To build the open mode options list use:

NWSMTSGetOpenModeOptionString

6. Allow the user to select or modify the options.
7. Put the user's selection(s) and specifications into the following data structures:

NWSM\_SELECTION\_LIST

NWSM\_SCAN\_CONTROL

NWSMTS\_DATA\_SET\_NAME\_LIST

## Back Up

During the back up phase, the SME scans for the specified data sets, puts it and other information into SIDF records, and sends the records to SDI for storage. The following steps illustrate this:

1. Before scanning for data sets, the application resets its view of the target (e.g., a volume may have been mounted since the back up session started) by calling:

NWSMTSBuildResourceList

2. The engine notifies the TSA of the user's selections and finds the first data set by passing the converted user selections (see step 7 from "Back Up Options") to:

NWSMTSScanDataSetBegin

3. Open the data set by calling:

NWSMTSOpenDatasetForBackup

Through this function each data set can be opened in various modes (e.g., open read only).

4. Get the data set data by using:

NWSMTSReadDataSet

**Note:** This call formats the data set data according to SIDF.

5. Format the scan information according to SIF, put it and the data set data into an SIF record, and put the record into an SIF transfer buffer. The following functions can be used to format the scan information and place the result and the data set data into the transfer buffer:

NWSMSetRecordHeader  
NWSMUpdateRecordHeader

See *Storage Management Services Library* for more information about these functions.

6. If the transfer buffer is full, update the record header and write it to the media via SDI.
7. Close the data set by using:

NWSMTSCloseDataSet

8. Get the next data set by calling:

NWSMTSScanNextDataSet

9. Repeat steps 3 through 8 until all data sets are found.

## Restore Options

The restore options phase involves gathering the TSA-specific options and resources, allowing the user to choose one or more of these options, and notifying the TSA of the selected options (see "TSA Issues" in this chapter for more information). The following steps illustrates this (there is no set order for gathering the options):

1. To build the selection list options, use:

NWSMTSGetTargetSelectionTypeStr

2. To get the resources (i.e., root or roots of a target) use:

NWSMTSScanTargetServiceResource

**Note:** For more information about resources, see "TSA-Specific Resources" in this chapter.

During this step, you may get the target's supported name spaces and information about them by using:

NWSMTSScanSupportedNameSpaces

---

and

NWSMTSGetNameSpaceTypeInfo

3. To build the open mode options list use:

NWSMTSGetOpenModeOptionString

4. Allow the user to select or modify the options.
5. Put the user's selection(s) and specifications into the following data structures:

NWSM\_SELECTION\_LIST

NWSMTS\_DATA\_SET\_NAME\_LIST

## Restore

During the restore phase, the application sets the restore options, scans for the specified data sets, unpacks the scan information and data set, and sends it to the TSA for writing. The following steps illustrate this:

**Note:** The restore method presented here is very slow. See NWSMTSSetRestoreOptions for a faster method.

1. Set the restore options by calling:

NWSMTSSetRestoreOptions

2. Get a transfer buffer from SDI.
3. Get a record from the transfer buffer by using:

NWSMGetRecordHeader

We are assuming that each record contains an entire data set. For more information about NWSMGetRecordHeader see *Storage Management Services Library*.

4. Get the data set's name from the data set information (scan information) by using:

NWSMGetOneName

**Note:** See *Storage Management Services Utilities Library* for this function.

5. Restore the data set by calling:

### NWSMTSOpenDataSetForRestore

**Note:** This call tells you if the user excluded the data set.

If the user excluded the data set, go back to step 2 or 3.

6. Restore the data set by calling:

### NWSMTSWriteDataSet

**Note:** This function deformats the data set data.

7. Close the data set by calling:

### NWSMTSCloseDataSet

8. Repeat steps 2 through 7 until all data sets are restored.

## Connection Termination

To disconnect the application from the target service and TSA call:

### NWSMTRSReleaseTargetService

and

### NWSMReleaseTSA

## Miscellaneous Functions

Listed below are other functions that can be used to help the SME programmer.

### NWSMTSGetUnsupportedOptions

### NWSMFreeNameList

### NWSMTSCatDataSetName

### NWSMTSGetNameSpaceTypeInfo

### NWSMTSParseDataSetName

### NWSMTSSeparateDataSetName

### NWSMTSGetTargetServiceType

Also see *Storage Management Services Utilities Library* for additional functions.

## Call Summary

<b>Connections</b>	
NWSMListTSAs NWSMConnectToTSA NWSMTSListTargetServices NWSMTSConnectToTargetService	
<b>Options</b>	
NWSMTSGetTargetScanTypeString NWSMTSGetTargetSelectionTypeStr NWSMTSScanTargetServiceResource NWSMTSGetOpenModeOptionString	
<b>Back Up</b>	<b>Restore</b>
NWSMTSScanDataSetBegin NWSMTSOpenDataSetForBackup NWSMTSReadDataSet NWSMSetRecordHeader NWSMUpdateRecordHeader NWSMTSCloseDataSet NWSMTSScanNextDataSet	NWSMTSSetRestoreOptions NWSMGetRecordHeader NWSMGetOneName NWSMTSOpenDataSetForRestore NWSMTSWriteDataSet NWSMTSCloseDataSet
<b>Connection Termination</b>	
NWSMTRSReleaseTargetService NWSMReleaseTSA	

