E

Notes For SMS Developers

SMS Overview

SMS divides the storage management tasks into three key components: the Target Service Agent (TSA), the Storage Management Engine (SME), and the Storage Device Interface (SDI). The TSA is a target-specific process that understands the target's data structures and is designed to isolate the knowledge of a target's data structures to one module. It performs I/O on the target's data and format/deformats the data according to SIDF.

The SME controls the session and tells the TSA kind(s) of data a user wants to back up or restore. Its main task is two-fold. The SME first receives formatted data from the TSA, does additional processing, and sends it to the SDI for storage onto the media. The second is to receive data from SDI and send it to the TSA when restoring the data.

The SDI takes care of managing the devices and media, ensuring the session is SIDF compatible, and handling requests from the SME.

TSA, SME, and SDI's Roles

The following text provides responsibility clarification for SMS developers. Responsibility of the three levels (media, transfer buffer, and data set level) is distributed among the TSA, the Storage Management Engine (SME), and the Storage Device Interface (SDI).

TSA

The TSA's main responsibility is to format\deformat the target's data set data according to SIDF's specifications at the data set level (see the read and write data set functions in *Target Service API*).

The TSA must be case sensitive if the name space type requires it. It checks all incoming strings from the SME and other sources for case sensitivity where needed.

Sections or FIDs that are not recognizable by the TSA are discarded. That is, if a session containing DOS and Macintosh data sets are restored to a target that supports only DOS data sets, the TSA discards the Macintosh data sets, except its DOS equivalent (i.e., the data fork is kept, but the resource fork is discarded).

Also, if a TSA receives an older data set that does not have all the features of the newer data set (e.g., a new attribute was added), the TSA creates missing information.

The data set level defines how the TSA formats the target's data. For more information about the role of the data set level, see "Data Set Level" in Chapter 1. This is the only SIDF level the TSA is concerned about.

SME

The SME is responsible for at least four items:

- Querying the TSA for the data set information and formatting it according to SIDF's specifications.
- Providing to SDI the developer specific information (sections and FIDs not defined in this document). Again, this is formatted according to SIDF's specifications (see the transfer buffer section for more information).

- Putting the TSA formatted data and SME formatted data set information into data set\subdata sets and transfer buffers. For more information about the transfer buffer, see *Storage Device API*.

- Logging the serviced data sets and their location.

SIDF allows the SME to use many media types (e.g., tape, Magneto Optical, Write Once Read Many [WORM]) without requiring knowledge about what it is writing to.

SDI

The SDI is responsible for the media level and parts of the transfer buffer level, and is not responsible for the data set level. "Section Responsibilities" describes in detail what fields SDI is responsible for.

Note: SDI writes all file marks and set marks.

The transfer buffer, defined by SIDF, is the same transfer buffer used by SDI to transfer data between the SME, itself, and the media. The SDI establishes the maximum transfer buffer size. During a write session, the SME may give to SDI any transfer buffer size up to the maximum size. However, during a read session, the SME must give to SDI a transfer buffer of "maximum size" bytes or data may be lost.

The smallest unit of data that SDI can accept from the SME is a logical sector. **The SDI function, NWSMSDWriteTransferBuffer**, returns the location of the transfer buffer.

Formatting Data

SMS developers do not need to know the information presented in "Data Representation" to format data according to SIDF's specifications for data sets defined in this document. SMS provides the routines to format and deformat the data sets (see **NWSMPutNextField** and **NWSMGetNextField** in *Storage Management Services Utilities Library)*. Other functions can be used to help format/deformat information (see "Section Responsibility"). For those defining new data sets, "Data Representation" is critical to designing your sections.

Section Responsibility

The headings below describes the responsibilities of each module for each field in a section. For a complete description of each section see "Sections."

Blank Space

SME

Soft Media Mark

SDI

Media Header

The SME must fill in the following fields:

Media set create date and time
Media set label
Media number

See **NWSMSetMediaHeaderInfo** and **NWSMGetMediaHeaderInfo** to create the above fields (see *Storage Management Services Utilities Library*). To write the header to the media, call SDI's **NWSMSDLabelMedia**.

SDI must fill out the rest of the fields.

Session Header

The SME must fill the following fields:

Session Date and Time
Session Description
Software Name
Software Type
Software version
Source Name
Source Type
Source version

SMS provides two routines that easily set and get the above information. See **NWSMSetSessionHeaderInfo** and **NWSMGetSessionHeaderInfo** in *Storage Management Services Utilities Library* for more information.

SDI must fill the rest of the fields. SDI receives the session description from the user via the SME and determines the maximum size of the transfer buffer during the backup session (see **NWSMSDOpenSessionForWriting**).

The source fields contain information about which target the data was retrieved from. SMS developers can get this information from **NWSMGetTargetServiceType** (see Target Service API). The TSA defines Source type.

Software Type contains a null-terminated string that describes the software type. All SMS type engines should put "SMS Engine."

Session Trailer

For a write session, SDI fills the fields above from the information copied from the media header.

Session Index

The SME is responsible for building this entire section and writing it out to the media. To build this section, the SME copies data from the Data Set Information section (this section can be built as the data sets are retrieved from the TSA).

The location information can be found as follows. The location of each transfer buffer is returned by SDI in the control block when the data set is written to the media. The offset to each data set is known by the SME.

> **Note:** To write the information to the media, call **NWSMSDWriteSessionData** with session data type set to session index (see *Storage Device API*).

Media Index

The media index is a log that records where each session exists on a medium. SDI creates and maintains the media index.

Media Trailer

SDI creates this section.

Transfer Buffer Header

SDI builds this section; however, the SME must set Transfer Buffer Sequence number (see **NWSMSDWriteSessionData**). SDI copies the following fields from the session header:

Session Date and Time
Session Description
Software Name

Software Type
Software Version
Source Name
Source Type
Source Version

The SME can write developer specific information into the transfer buffer through SDI (see **NWSMSDOpenSessionForWriting** and **NWSMSDWriteSessionData** in *Storage Device API*).

Unused in this Buffer. SDI calculates the unused area from information contained in NWSMSD_CONTROL_BLOCK

Session Description. SDI repeats this information from the data it was given during the **NWSMSDOpenSessionForWriting** call.

Transfer Buffer Sequence contains the transfer buffer's position within a session. During a write session, the SME sets this to one and increments it after passing it to SDI. This value is set via **NWSMSDWriteSessionData**.

Sector Physical Address - Contains the address of the transfer buffer. SDI fills the data portion of this field during **NWSMSDWriteSessionData**.

Revision Level - Contains a non-null-terminated string that indicates the revision level of the TSA servicing the data. For SMS, this contains "REV 1.00".

CRC Type: This field contains a string identifying the CRC type. For SMS, the data is "SMS".

Data Set Header

The SME is responsible for creating this section. Two routines described in *Storage Management Services Utilities Library* are used to create and update this section. These are the **NWSMSetRecordHeader** and **NWSMUpdateRecordHeader** functions.

The SME is required to encapsulate the data set information and data set data into a data set. The TSA formats the data set data; however, the SME must format the data set information according to the SIDF specifications. The data set information is returned by the TS API scanning functions.

Subdata Set Header

The SME creates this section. If a data set overflows a transfer buffer the SME should call **NWSMUpdateRecordHeader** to change the data set's size. Two routines, **NWSMSetRecordHeader** and **NWSMUpdateRecordHeader**, can be used to create and update the subdata set header (see *Storage Management Services Utilities Library*).

Data Set Information

The SME builds this section. The TS API scanning functions returns the scan information and data set name list. These can be formatted by **NWSMSetRecordHeader** and **NWSMUpdateRecordHeader** (see *Storage Management Services Utilities Library*). See *Target Service API* for more information on

NWSMTSScanDataSetBegin
NWSMTSScanNextDataSet
NWSMTSScanDataSetEnd

See "NWSM_SCAN_INFORMATION" and "NWSM_DATA_SET_NAME_LIST" in *Target Service API*.

Creator ID. The TSA defines the ID.

Transaction Set Header

Transaction Set Type. The TSA defines this.

Transaction Set Name. The TSA defines this..

Full Paths

The full path is required if the data set is a parent or if *returnChildTerminalNodeNameOnly* of the associated NWSM_SCAN_CONTROL structure is FALSE. For more information about this structure, see *Data Requestor TS API*.