

# Legato<sup>®</sup> Storage Manager

Command Reference Guide

Release 9.0.1 for Windows NT and Windows 2000

June 2001

Part No. A90175-01

This *Legato Storage Manager Command Reference Guide* contains complete information on the commands to use with Legato Storage Manager (LSM) for Oracle data storage management. It describes the commands for setting up and configuring the LSM software and resources. It also provides details on the commands for managing the Oracle backup data, devices, media, and the LSM online index files. See the *Legato Storage Manager Administrator's Guide* for complete information on LSM configuration.

ORACLE<sup>®</sup>

---

Legato Storage Manager Command Reference Guide, Release 9.0.1 for Windows NT and Windows 2000

Part No. A90175-01

Copyright © 2001, Oracle Corporation. All rights reserved.

Primary Author: Anita Puronto

Graphic Designer: Valarie Moore

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle7, Oracle8, and Oracle8i are trademarks or registered trademarks of Oracle Corporation. Legato is a registered trademark of Legato Systems, Inc. Other names may be trademarks of their respective owners.

---

---

# Contents

|   |             |
|---|-------------|
| <b>Send Us Your Comments .....</b>      | <b>vii</b>  |
| <b>Preface.....</b>                     | <b>ix</b>   |
| Audience .....                          | x           |
| Organization.....                       | x           |
| Related Documentation .....             | xi          |
| Conventions.....                        | xii         |
| Documentation Accessibility .....       | xiii        |
| <br>                                    |             |
| <b>1 Introduction</b>                   |             |
| <b>Categories of LSM Commands .....</b> | <b>1-2</b>  |
| <br>                                    |             |
| <b>2 LSM Commands</b>                   |             |
| <b>mm_data(5).....</b>                  | <b>2-1</b>  |
| <b>mminfo(8).....</b>                   | <b>2-6</b>  |
| <b>mmlocate(8).....</b>                 | <b>2-20</b> |
| <b>mmpool(8).....</b>                   | <b>2-23</b> |
| <b>mmrecov(8).....</b>                  | <b>2-25</b> |
| <b>nsr(5).....</b>                      | <b>2-31</b> |
| <b>nsr(8).....</b>                      | <b>2-35</b> |
| <b>nsr_archive_client(5).....</b>       | <b>2-43</b> |
| <b>nsr_archive_request(5).....</b>      | <b>2-46</b> |
| <b>nsr_client(5).....</b>               | <b>2-50</b> |
| <b>nsr_crash(8).....</b>                | <b>2-58</b> |

|                                     |       |
|-------------------------------------|-------|
| <b>nsr_data(5)</b> .....            | 2-61  |
| <b>nsr_device(5)</b> .....          | 2-65  |
| <b>nsr_directive(5)</b> .....       | 2-76  |
| <b>nsr_getdate(3)</b> .....         | 2-78  |
| <b>nsr_group(5)</b> .....           | 2-80  |
| <b>nsr_label(5)</b> .....           | 2-86  |
| <b>nsr_layout(5)</b> .....          | 2-89  |
| <b>nsr_license(5)</b> .....         | 2-91  |
| <b>nsr_migration(5)</b> .....       | 2-94  |
| <b>nsr_notification(5)</b> .....    | 2-99  |
| <b>nsr_policy(5)</b> .....          | 2-102 |
| <b>nsr_pool(5)</b> .....            | 2-104 |
| <b>nsr_regexp(5)</b> .....          | 2-108 |
| <b>nsr_repack_schedule(5)</b> ..... | 2-110 |
| <b>nsr_resource(5)</b> .....        | 2-113 |
| <b>nsr_schedule(5)</b> .....        | 2-118 |
| <b>nsr_service(5)</b> .....         | 2-121 |
| <b>nsr_shutdown(8)</b> .....        | 2-130 |
| <b>nsr_stage(5)</b> .....           | 2-132 |
| <b>nsr_storage_node(5)</b> .....    | 2-136 |
| <b>nsradmin(8)</b> .....            | 2-140 |
| <b>nsralist(8)</b> .....            | 2-146 |
| <b>nsrarchive(8)</b> .....          | 2-147 |
| <b>nsrcap(8)</b> .....              | 2-150 |
| <b>nsrcat(8)</b> .....              | 2-153 |
| <b>nsrck(8)</b> .....               | 2-154 |
| <b>nsrclone(8)</b> .....            | 2-158 |
| <b>nsrd(8)</b> .....                | 2-164 |
| <b>nsrexec(8)</b> .....             | 2-166 |
| <b>nsrexecd(8)</b> .....            | 2-167 |
| <b>nsrfile(8)</b> .....             | 2-169 |
| <b>nsrhsmck(8)</b> .....            | 2-172 |
| <b>nsrib(8)</b> .....               | 2-174 |
| <b>nsrim(8)</b> .....               | 2-179 |
| <b>nsrindexasm(8)</b> .....         | 2-186 |

|                              |       |
|------------------------------|-------|
| <b>nsrindexd(8)</b> .....    | 2-188 |
| <b>nsrinfo(8)</b> .....      | 2-190 |
| <b>nsrlic(8)</b> .....       | 2-195 |
| <b>nsrls(8)</b> .....        | 2-198 |
| <b>nsrmig(8)</b> .....       | 2-200 |
| <b>nsrmm(8)</b> .....        | 2-202 |
| <b>nsrmmmd(8)</b> .....      | 2-210 |
| <b>nsrmmdbasm(8)</b> .....   | 2-211 |
| <b>nsrmmdbd(8)</b> .....     | 2-213 |
| <b>nsrmon(8)</b> .....       | 2-216 |
| <b>nsrpmig(8)</b> .....      | 2-217 |
| <b>nsrports(8)</b> .....     | 2-221 |
| <b>nsrrepack(8)</b> .....    | 2-223 |
| <b>nsrretrieve(8)</b> .....  | 2-230 |
| <b>nsrssc(8)</b> .....       | 2-232 |
| <b>nsrstage(8)</b> .....     | 2-236 |
| <b>nsrwatch(8)</b> .....     | 2-240 |
| <b>nwadmin(8)</b> .....      | 2-242 |
| <b>nwarchive(8)</b> .....    | 2-244 |
| <b>nwbackup(8)</b> .....     | 2-246 |
| <b>nwrecover(8)</b> .....    | 2-248 |
| <b>nwretrieve(8)</b> .....   | 2-251 |
| <b>preclntsave(8)</b> .....  | 2-253 |
| <b>pstclntsave(8)</b> .....  | 2-254 |
| <b>recover(8)</b> .....      | 2-256 |
| <b>save(8)</b> .....         | 2-263 |
| <b>savefs(8)</b> .....       | 2-268 |
| <b>savegrp(8)</b> .....      | 2-273 |
| <b>scanner(8)</b> .....      | 2-291 |
| <b>tapeexercise(8)</b> ..... | 2-298 |
| <b>uasm(8)</b> .....         | 2-300 |



---

---

# Send Us Your Comments

## **Legato Storage Manager Command Reference Guide, Release 9.0.1 for Windows NT and Windows 2000**

**Part No. A90175-01**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- E-mail: [ntdoc\\_us@oracle.com](mailto:ntdoc_us@oracle.com)
- FAX - (650) 506-7365 Attn: Oracle Database for Windows Documentation
- Postal service:  
Oracle Corporation  
Oracle Database for Windows Documentation Manager  
500 Oracle Parkway, Mailstop 1op6  
Redwood Shores, CA 94065  
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services. Contact information for Oracle Support Services is available at this Web site:

<http://www.oracle.com/support/>

---

---

# Preface

This *Legato Storage Manager Command Reference Guide* contains complete information on the commands to use with Legato Storage Manager (LSM) for Oracle data storage management. It describes the commands for setting up and configuring the LSM software and resources. It also provides details on the commands for managing the Oracle backup data, devices, media, and the LSM online index files.

This preface contains these topics:

- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

## Audience

The information in this guide is for system administrators and Oracle database administrators (DBAs) who are responsible for maintaining Oracle Server backup and recovery systems using LSM. It includes information on configuring an Oracle data storage management system using LSM for Oracle9i level servers on Windows 2000 and Windows NT.

You need a working knowledge of Oracle concepts and terminology, especially as related to Oracle database backup and recovery, to use this guide effectively. This guide does *not* include LSM installation instructions. For information on installing LSM, refer to the *Oracle Server Installation Guide* for your particular system. For more information on configuring LSM, refer to the *Legato Storage Manager Administrator's Guide*.

## Organization

This document contains:

### **Chapter 1, "Introduction"**

This chapter introduces you to the LSM commands provided as part of the LSM Server software. It describes the six different categories of LSM commands and lists the manual pages in Chapter 2 that provide detailed descriptions of each LSM command.

### **Chapter 2, "LSM Commands"**

This chapter includes a set of manual pages describing each available LSM command. Each manual page provides details on the correct LSM command syntax, the LSM resources that you can configure, and specific examples of how to use the LSM commands to configure and manage the LSM Server and perform backup and restore operations.

## Related Documentation

This document provides Legato Storage Manager usage and configuration information. For more information on installation, configuration, and tuning recommendations for a production database system, refer to the following manuals:

- *Oracle9i Database installation guide for Windows*
- *Oracle9i Backup and Recovery Concepts*
- *Oracle9i Recovery Manager User's Guide*
- *Oracle9i Recovery Manager Reference*
- *Oracle9i User-Managed Backup and Recovery Guide*

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://technet.oracle.com/membership/index.htm>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://technet.oracle.com/docs/index.htm>

## Conventions

Because UNIX is case-sensitive, conventions in this document may differ slightly from those used in Oracle product documentation.

The following conventions apply to text in this guide:

|                        |   |
|------------------------|---|
| <code>monospace</code> | Monospace type indicates Windows command-line commands, directory names, path names, and file names.  |
| brackets [ ]           | Words enclosed in brackets indicate key names (for example, Press [Return]). Note that brackets have a different meaning when used in command syntax. |
| <i>italics</i>         | Italic type indicates a variable, including variable portions of filenames, document titles, or emphasis.   |
| UPPERCASE              | Uppercase letters indicate Structured Query Language (SQL) statements, initialization parameters, or environment variables.                           |

Command syntax is shown in `monospace` font in this guide. The following conventions apply to command syntax:

|                        |  |
|------------------------|--|
| backslash \            | A backslash indicates a command that is too long to fit on a single line. Enter the line as printed (with a backslash) or enter it as a single line without a backslash:<br><pre>dd if=/dev/rdskc0t1d0s6 of=/dev/rst0 bs=10b \ count=10000</pre> |
| braces { }             | Braces indicate required items: <code>.DEFINE {macro1}</code>  |
| brackets [ ]           | Brackets indicate optional items: <code>cvtrct termname [outfile]</code><br>Note that brackets have a different meaning when used in regular text.   |
| ellipses ...           | Ellipses indicate an arbitrary number of similar items:<br><code>CHKVAL fieldname value1 value2 ... valueN</code>  |
| <i>italics</i>         | Italic type indicates a variable. Substitute a value for the variable:<br><code>library_name</code>  |
| <code>monospace</code> | Monospace type indicates commands or text you type:<br><code>connect</code>  |

|                  |   |
|------------------|---|
| <i>monospace</i> | Monospace italic type indicates commands or text you type and                               |
| <i>italic</i>    | need to substitute a variable: <code>nwadmin -s server-name</code>                          |
| vertical line    | A vertical line indicates a choice within braces or brackets:<br><i>SIZE filesize [K M]</i> |

## Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.



---

---

# Introduction

This *Legato Storage Manager Command Reference Guide* describes the LSM commands and LSM Server resources including:

- Correct usage of each LSM command
- Descriptions of the LSM Server resources that you can configure
- Example usage of each LSM command

---

---

**Note:** This guide is based on the corresponding guide for the Legato NetWorker upgrade product for UNIX. As a result, several features documented in [Chapter 2, "LSM Commands"](#) are not supported by LSM--including ASMs, registration, archiving, ClientPaks, cloning, and using autochangers (or jukeboxes) and silos. Be sure to convert any filenames that appear in UNIX format to the corresponding Windows filename format.

---

---

## Categories of LSM Commands

The LSM commands described in this guide are provided as part of the LSM Server software. The LSM commands can be grouped into the following six categories:

- [Client and Media Index Management Commands](#)
- [Data Management Commands](#)
- [Device and Media Management Commands](#)
- [LSM Resource Commands](#)
- [LSM Server Configuration Commands](#)
- [Other LSM Commands](#)

The remaining sections of this chapter describe the six categories of LSM commands. The manual pages in [Chapter 2, "LSM Commands"](#) provide detailed descriptions of each individual LSM command.

---

---

**Note:** The manual page [nsr\(8\)](#) in Chapter 2 provides a general overview of LSM features and functionality and the correct usage of the different LSM commands.

---

---

### Client and Media Index Management Commands

This category of LSM commands provides information on how to manage the online client and media index files on the LSM Server. See the following manual pages for details on the LSM commands in this category:

- [mminfo\(8\)](#)
- [mmlocate\(8\)](#)
- [mmpool\(8\)](#)
- [mmrecov\(8\)](#)
- [nsrck\(8\)](#)
- [nsrib\(8\)](#)
- [nsrim\(8\)](#)
- [nsrindexd\(8\)](#)

- [nsrinfo\(8\)](#)
- [nsrls\(8\)](#)

## Data Management Commands

This category of LSM commands provides information on how to manage the Oracle data backed up by the LSM Server. See the following manual pages for details on the LSM commands in this category:

- [nsralist\(8\)](#)
- [nsrarchive\(8\)](#)
- [nsrclone\(8\)](#)
- [nsrhmck\(8\)](#)
- [nsrmig\(8\)](#)
- [nsrmmd\(8\)](#)
- [nsrmmdbd\(8\)](#)
- [nsrpmig\(8\)](#)
- [nsrretrieve\(8\)](#)
- [nsrssc\(8\)](#)
- [nsrstage\(8\)](#)
- [preclntsave\(8\)](#)
- [pstclntsave\(8\)](#)
- [recover\(8\)](#)
- [save\(8\)](#)
- [savefs\(8\)](#)
- [savegrp\(8\)](#)
- [scanner\(8\)](#)
- [uasm\(8\)](#)

## Device and Media Management Commands

This category of LSM commands provides information on how to manage the Oracle backup devices and media used by the LSM Server. See the following manual pages for details on the LSM commands in this category:

- [mm\\_data\(5\)](#)
- [nsrmm\(8\)](#)
- [nsrrepack\(8\)](#)
- [tapeexercise\(8\)](#)

## LSM Resource Commands

This category of LSM commands provides information on each of the LSM resources that you can set up and configure on the LSM Server. See the following manual pages for details on the LSM commands in this category:

- [nsr\\_archive\\_request\(5\)](#)
- [nsr\\_client\(5\)](#)
- [nsr\\_data\(5\)](#)
- [nsr\\_device\(5\)](#)
- [nsr\\_directive\(5\)](#)
- [nsr\\_group\(5\)](#)
- [nsr\\_label\(5\)](#)
- [nsr\\_migration\(5\)](#)
- [nsr\\_notification\(5\)](#)
- [nsr\\_policy\(5\)](#)
- [nsr\\_pool\(5\)](#)
- [nsr\\_repack\\_schedule\(5\)](#)
- [nsr\\_resource\(5\)](#)
- [nsr\\_schedule\(5\)](#)
- [nsr\\_service\(5\)](#)

- [nsr\\_stage\(5\)](#)
- [nsrports\(8\)](#)

## LSM Server Configuration Commands

This category of LSM commands provides information on each of the programs that you can use to set up and configure the LSM Server. See the following manual pages for details on the LSM commands in this category:

- [nsradmin\(8\)](#)
- [nsrexec\(8\)](#)
- [nsrexecd\(8\)](#)
- [nsrwatch\(8\)](#)
- [nwadmin\(8\)](#)
- [nwarchive\(8\)](#)
- [nwbackup\(8\)](#)
- [nwrecover\(8\)](#)
- [nwretrieve\(8\)](#)

## Other LSM Commands

This category of LSM commands provides information on configuration and usage of LSM that is not included in the other categories. See the following manual pages for details on the LSM commands in this category:

- [nsr\(5\)](#)
- [nsr\(8\)](#)
- [nsr\\_archive\\_client\(5\)](#)
- [nsr\\_crash\(8\)](#)
- [nsr\\_getdate\(3\)](#)
- [nsr\\_layout\(5\)](#)
- [nsr\\_license\(5\)](#)
- [nsr\\_regexp\(5\)](#)
- [nsr\\_shutdown\(8\)](#)

- **nsr\_storage\_node(5)**
- **nsrcap(8)**
- **nsrcat(8)**
- **nsrd(8)**
- **nsrfile(8)**
- **nsrindexasm(8)**
- **nsrlic(8)**
- **nsrmmdbasm(8)**
- **nsrmon(8)**

---

---

## LSM Commands

### mm\_data(5)

#### Name

**MM data** – Legato Storage Manager media multiplexor data (tape and disk) format

#### Description

This documents the data format that the Legato Storage Manager media multiplexor daemon, **nsrmmd(8)**, writes to long term storage media such as tapes and optical disks. See **nsr\_device(5)** and **nsrmm(8)** for a discussion of supported device families and types. The format described here applies to any fixed record device, such as raw disks, or fixed record tape devices with file marks. Legato Storage Manager uses the *eXternal Data Representation* (XDR) standard to write media which can be interchanged among a wide variety of computers. Only the mechanism used to multiplex save set streams onto the storage media is described here; the formats of save set streams depend on the type of Legato Storage Manager client, and are described in **nsr\_data(5)**.

A *volume* is one physical piece of media such as a tape reel or disk cartridge. A tape volume is made up of multiple media *files*, and each media file may contain several media *records*. These media files and records should not be confused with a client's (for example UNIX or DOS) user files or records; the two do not necessarily correspond. For example, a given media file or even a single media record may contain many small client user files. On the other hand, a single large client file may be split across several media files, and even across several volumes. Media files do not span volume boundaries. Save sets may span media files and even volumes.

On most tapes, media files can be skipped very quickly by the device's hardware or associated device driver software, and the hardware can detect when an end of a file has been reached. On some tapes, records can also be quickly skipped forward. Otherwise, access to the media is sequential.

Media records are described by the **mrecord** structure. Label records are fixed in size, **MINMRECSIZE** bytes. Other records can potentially be a larger size that must be some constant for the rest of the volume. Legato Storage Manager always writes, reads and skips data in units of full-sized media records. Each **mrecord** contains zero or more **mchunks** of data from either one or more client save sessions, or used internally for synchronization and labels. The XDR format of a media file's **mrecords** and **mchunks** are as follows:

```
const MINMRECSIZE = 32768;           /* minimum media record size */
const MMAXCHK = 2048;                /* max number of chunks in record */
const MHNDLEN = 124;                 /* private area len for handlers */
typedef unsigned long ssid_t;        /* save set id */
typedef unsigned long ssoff_t;       /* an offset in a save set */
typedef unsigned long volid_t;       /* key for the volume database */
struct mchunk {
    ssid_t mc_ssid;                   /* owning save set id */
    ssoff_t mc_low;                   /* 1st byte, relative to save stream */
    opaque mc_data<MINMRECSIZE>;     /* chunk's data */
};
struct mrecord {
    opaque mr_handler[MHNDLEN];       /* private to media handler*/
    u_long mr_orec;                   /* record size */
    volid_t mr_volid;                 /* encompassing volume's id*/
    u_long mr_fn;                     /* encompassing file number*/
    u_long mr_rn;                     /* record number within the file */
    u_long mr_len;                    /* record byte length */
    mchunk mr_chunk<MMAXCHK>;        /* chunks of save streams */
};
```

The first field of an **mrecord**, **mr\_handler**, is reserved for media-specific data (currently it is not used by any implementation). The **mr\_orec** field is the size of the current record. A media record's header fields, **mr\_volid**, **mr\_fn**, and **mr\_rn**, are used to check the tape position and the data read from the record. The file numbers and record numbers start at zero and increment sequentially. The record number is reset each time the file number is incremented. On disks, file numbers are always zero. The **mr\_len** field is the actual number of valid bytes in this record, as opposed to the size of the device's read or write request.

If file or record skipping is unreliable, Legato Storage Manager can still recover from isolated errors, at worst by rewinding and reading the tape from the start. If a volume can be physically unmounted or mounted without notice to the media management daemon, then the volume identifier in each record provides a quick way of verifying when this happens, without the need for a full rewind and reading of the label in most cases.

The **mchunks** within an **mrecord** contain client data from one or more save sessions. The **mc\_ssid** and **mc\_low** values are used to reconstruct the save streams from the chunks within the records. The **mc\_data** field holds the actual data of each chunk. For a given save set, **mc\_low** plus the length of **mc\_data** should equal the following chunk's value for **mc\_low**. Save sets may be intermingled arbitrarily within media records.

The first chunk of the first record of the first media file on the volume encapsulates the **volume label** information; for some media, the second chunk contains additional volume information, for example the media pool the volume belongs to; subsequent data in the first file is reserved for future expansion. The label may be duplicated in a second file for redundancy, in case the first copy of the label gets accidentally overwritten. The formats of the volume label and additional label information are described by the following XDR data structures:

```

const MVOLMAGIC = 0x070460;           /* volume magic number */
const NSR_LENGTH = 64;               /* length of several strings */
const RAP_MAXNAMELEN = 64;          /* max length of attribute name */
struct mvollabel {
    u_long mvl_magic;                /* medium volume verification number */
    u_long mvl_createtime;           /* time at which volume labeled */
    u_long mvl_expiretime;          /* time for volume to expire */
    u_long mvl_resize;              /* expected size of mrecords */
    volid_t mvl_volid;              /* medium volume id */
    string mvl_volname<NSR_LENGTH>; /* medium volume name */
};
struct vallist {
    vallist *next;
    string value<>;                 /* attribute value */
};
struct attrlist {
    attrlist *next;
    string name<RAP_MAXNAMELEN>;    /* attribute name */
    vallist *values;                /* attribute values */
};
/*
 * Additional information may includes the following attributes (listed by the

```

```
    * name they are stored with): "volume pool" : the media pool
    */
struct mvolinfo {
    struct attrlist *mvi_attributes; /* any other information */
};
```

The **mvl\_magic** field must be equal to **MVOLMAGIC** in order for the chunk to represent a valid volume label. If the volume label changes in the future, the new format will have another “magic” number, but the format described here must still be allowed. The **mvl\_volid** is an internal identifier assigned and managed by the media manager. The **mvl\_volname** is the volume name that is assigned when the media is first labeled. The time fields are in UST format – the number of seconds elapsed since 00:00 GMT, January 1, 1970. The **mvl\_recsiz** is the size of all subsequent media records found on the tape.

The **mvp\_pool** is the pool name that is assigned when the media is first labeled. Different media pools allow administrators to segregate their data onto sets of volumes. Media cannot be reassigned from one media pool to another. Pool names are a maximum of **NSR\_LENGTH** characters long.

Synchronization marks, called **schunks**, are also written periodically to the media for each save set. Synchronization chunks are used by **scanner(8)** when verifying or extracting directly from a volume. They are also used by **nsrmm** when trying to recover from media errors during file recovery. The following XDR data structure describes a synchronization chunk:

```
struct schunk {
    opaque ssi_host[NSR_LENGTH]; /* save set host */
    opaque ssi_name[NSR_LENGTH]; /* symbolic name */
    u_long ssi_time; /* save time */
    u_long ssi_expiry; /* expiration date */
    u_long ssi_size; /* actual size saved */
    u_long ssi_nfiles; /* number of files */
    ssid_t ssi_ssid; /* ssid for this save set */
    u_long ssi_flag; /* various flags, see below*/
    u_long ssi_info; /* volid or ssid, see below*/
};

#define SSI_START 1 /* start of a save set */
#define SSI_SYNC 2 /* synchronization point */
#define SSI_CONT 3 /* continued from another volume */
#define SSI_END 4 /* end of this save set */
#define SSI_SSMASK 0x0000000f /* save set sync chunk type*/
#define SSI_LBIAS 0x10000000 /* the level is included in the flags */
#define SSI_IMASK 0xff000000 /* mask to cover bits for level */
#define SSI_LSHIFT 24 /* shift amount for the level */
```

```
#define SSI_INCOMPLETE 0x00010000 /* not finished (aborted) */
#define SSI_CONTINUED 0x00800000 /* continued save set series */
```

The **ssi\_host** is the name of the index which contains this save set. Traditionally this is the host name of the client where the save set originated. The **ssi\_name** is the save set name to be presented to the user. These are both null-terminated strings, even though the fields are fixed length. The **ssi\_time** field contains the create time of the save set in UST. The **ssi\_expiry** field is the expiration date, in UST for this save set. This field is zero if an explicit save set expiration time was not specified when the save set was created. The **ssi\_size** and **ssi\_nfiles** are the number of bytes and number of files saved so far for this save set. The **ssi\_ssid** is the save set identifier of this save set.

The **ssi\_flag** indicates the type of this synchronization chunk, the level of save set, and other information about the save set. There are four basic types of synchronization marks that can be found from examining **ssi\_flag & SSI\_SSMASK**. **SSI\_START** is used to mark the beginning of a save set. **SSI\_SYNC** marks a periodic synchronization point and is only written at an exact file boundary in the save set. **SSI\_CONT** indicates that this is the continuation of a save set that started on a different volume. When **ssi\_flag & SSI\_SSMASK** is **SSI\_CONT**, **ssi\_info** contains the volume identifier for the save set's preceding volume. These synchronization chunks are used when a save set spans a volume boundary. **SSI\_END** marks the end of a save set.

If the **SSI\_LBIAS** bit is set then **ssi\_flag & SSI\_LMASK** shifted right by **SSI\_LSHIFT** specifies the level of the save set. The **SSI\_INCOMPLETE** bit indicates that this save set did not finish properly. This could be caused by a user interrupting an in progress save.

The **SSI\_CONTINUED** bit indicates that this save set is logically continued to or from another save set. These continued save sets are used to handle very large save sets. If the **SSI\_CONTINUED** bit is set and **ssi\_flag & SSI\_SSMASK** is **SSI\_START**, then **ssi\_info** gives the previous save set id that this save set was continued from. If the **SSI\_CONTINUED** bit is set and **ssi\_flag & SSI\_SSMASK** is **SSI\_END**, then **ssi\_info** gives the next save set id that this save set is continued to.

**See Also:** [nsr\\_device\(5\)](#), [nsr\\_data\(5\)](#), [nsrmm\(8\)](#), [nsrmm\(8\)](#), [nsrmmdbd\(8\)](#), [nsr\(8\)](#), [scanner\(8\)](#). RFC 1014 XDR: External Data Representation Specification

## mminfo(8)

### Name

**mminfo** – Legato Storage Manager media database reporting command

### Synopsis

**mminfo** [ **-avV** ] [ **-o order** ] [ **-s server** ] [ *report* ] [ *query* ] [ *volname...* ]

*report:*

[ **-m** | **-B** | **-S** | **-X** | **-r reportspec** ]

*query:*

[ **-c client** ] [ **-N name** ] [ **-t time** ] [ **-q queryspec** ]

### Description

The **mminfo** command reports information about Legato Storage Managers media and save sets. The **mminfo** command can produce several different reports depending on the flags specified. Several built-in reports can be specified using short-hand flags. Custom reports can also be specified. The default report, along with the built-in reports printed by the use of the **-v**, **-V**, **-m**, **-S**, **-B**, and **-X** flags, are described first as follows. The custom query and report generators, using the **-q queryspec** and **-r reportspec** options, are described in the **Custom Queries and Reports** section. Other options are described in the **Options** section.

Without any options, **mminfo** displays information about the save sets that completed properly during the last twenty four hours, and are still contained in an on-line file index (browsable save sets). The following information is printed for each save set: the containing volume name, the client's name, the creation date, the size saved on that volume, the save set level, and the save set name. The size field is displayed in Bytes (B), Kilobytes (KB), Megabytes (MB) or Terabytes (TB). The save set level will display 'full', 'incr', 'migration' or 1 through 9, for full, incremental, migration save sets, level 1 through 9, respectively. The level is only kept for scheduled saves and file migration; save sets generated by explicitly running the **save(8)** command (called *ad hoc* saves) do not have an associated level.

Specifying the **-v** flag causes three additional fields to be displayed: the creation time, the internal save set identifier (ssid), and two flags. One character is used for each flag.

The first flag indicates which part of the save set is on the volume. When the save is completely contained on the volume, a **c** is displayed. An **h** is displayed when the save set spans volumes and the **head** is contained on this volume. The remaining sections will be on other volumes. An **m** is displayed when the save set spans volumes and a **middle** section is contained on this volume. The head and tail sections will be on different volumes. There may be more than one middle section. A **t** is displayed when the **tail** section of a spanning save set is contained on this volume. Again, the other sections will be on other volumes.

The second flag indicates the status of the save set. A **b** indicates that the save set is in the on-line index and is **browsable** using the **recover(8)** command. An **r** indicates that the save set is not in the on-line index and is **recoverable** using the **scanner(8)** command. An **E** indicates that the save set has been marked eligible for recycling and may be over-written at any time. An **S** denotes that the save set was scanned in (or rolled in). Rolled in save sets are not subject to the standard index management procedures and will remain in the file index until the user manually purges the save set. An **a** indicates that the save was aborted before completion. Aborted save sets are removed from the on-line file index by **nsrck(8)**. An **i** indicates that the save is still in progress. The **-v** flag prints aborted, purged, and incomplete save sets in addition to the complete, browsable save sets printed by default.

The **-V** flag displays even more detail than the **-v** flag, and is generally used for debugging. This format also displays information (that is, media file number and record number) that can be used to speed the operation of the **scanner(8)** command. Rather than displaying one line for each save set for each volume, two lines are displayed each time a section of a save set occurs within a file on a volume. A single save set will have multiple index entries if it starts in one file on a volume and ends in another. The first line will contain the fields similar to those described previously for the default output, with two differences. First, the size field will now list the number of bytes that are contained in the section, rather than the total amount of the save set on this volume. Second, both the creation date and the time are shown. The second line contains the following fields: the save time in seconds since 00:00:00 GMT, Jan 1, 1970, the internal save set identifier (ssid), the offset of the first and last bytes of the save set contained within section, the media file number, the first record within the media file containing data for this save set, the internal volume identifier (volid), the total size of the save set, and the flags, described previously in the **-v** paragraph, indicating which part of the save set is contained in this media file (**c**, **h**, **m**, or **t**) and the save set's status (**b**, **r**, **a**, or **i**).

The **-m** flag causes **mminfo** to display the name of each volume in the media database, the number of bytes written to it, the percent of space used (or the word 'full' indicating that the volume is filled to capacity), the number of bytes read, the expiration date, the number of times the volume has been mounted, and the

volume's capacity. Volumes that are recyclable (see **nsrim(8)**) are flagged by an **E** in the 1st column (meaning **E**ligible for recycling). If a volume has been marked as manually-recyclable, an **M** is displayed instead of the **E**. If a volume is both manually-recyclable and eligible for recycling, an **X** will be displayed. Archive and migration volumes are flagged by an **A**, also in the 1st column. If the volume is not an archive or migration volume, and is not recyclable, no flag appears.

Specifying the **-v** flag with the **-m** flag causes three additional fields to be displayed: the internal volume identifier (**valid**), the number of the next file to be written, and the type of media.

Using a **-V** flag with the **-m** adds a column of flags to the output displayed. There are currently two possible flags. The **d** flag is set if the volume is currently being written (**dirty**). The **r** flag is set if the volume is marked as read-only. If neither condition is present, the flags column will be empty.

The **-S** flag displays a long, multi-line save set report, which is used for debugging. The number of lines varies for each save set. Due to the length, there are no column headers. Instead, each attribute of the save set is displayed in a 'name=value' manner, except the client and save set name, which are displayed as 'client:name'. The first line of each multi-line group starts on the left margin and includes the save set identifier (**ssid**), save time as both a date/time string and seconds since 00:00:00 GMT, Jan 1, 1970, and the client and save set names. Subsequent lines for this save set are indented. If the save set is part of a save set series (a 'continued save set') and is not the first in the series, the save set identifier of the previous save set in the series is shown on the second line by itself. The next line displays the level, the save set flags (in 'ssflags' format, as described in the table in the **Custom Queries and Reports** section), the save set size in bytes, the number of files in the save set, and the save set expiration date ('undef' means the expiration date is determined by the current policies for the client). If the save set has extended attributes, they are printed next, at most one attribute for each line. The clones or instances of the save set are shown last (every save set has at least once instance). The first line of each clone shows the clone identifier, the date and time the instance was created, and the flags for each clone (in 'clflags' format from the **Custom Queries and Reports** table). For each instance, each section of that instance is shown as a fragment line. The fragment line shows the offset of that fragment from the beginning of the save set, the volume identifier (**valid**) containing the fragment, the media file and record numbers of start of the fragment, an absolute positioning identifier (unused by existing servers), and the date of last access of the fragment. The **-v** and **-V** options have no effect on this report.

The **-X** flag prepares a save set summary report instead of one or more lines for each save set. Note that the entire media database must be examined to resolve this

query, making it very slow and expensive. The summary lists the total number of save sets, and breaks the total down into several overlapping categories summarizing the save set types. The recent save set usage, if appropriate to the query, is also printed. The categories are the number of fulls, the number of incrementals, The number of other non-full, non-incremental saves, the number of ad hoc, archive, migration, empty and purged save sets, the number of index save sets, and finally, the number of incomplete save sets. For recent usage, the number of save sets for each day are shown, up to a week ago, along with a summary of the week's save sets and, if applicable, a summary of the month's save sets. For each line, the number of files (saved in the time interval specified), number of save sets, total size, and average size for each save set, and average size for each file, are listed. The percentage of the amount saved for incrementals v.s. fulls and the percentage of browsable files are also printed, when appropriate. The **-v** and **-V** options have no effect on the summary report.

The **-B** flag performs a canned query to output, in a convenient format, the list of bootstraps generated in the previous five weeks. In this format, there is one line of output displayed for each matched save set. Each line contains the save date and time, save level, save set identifier (ssid), starting file number, starting record number, and the volume. The equivalent query is described in the following Examples. The **-v** and **-V** options have no effect on the bootstrap display.

## Options

### **-a**

Causes queries to apply to all complete, browsable save sets, not just those in the last 24 hours. This option is implied by the **-c**, **-N**, **-q**, **-m**, and **-o** options, described in the following sections. When combined with a media-only report (**-m** or a custom report showing only media information), **-a** applies to all volumes, not just those with complete and browsable save sets.

### **-c** *client*

Restricts the reported output to the media and/or save sets pertaining to the specified *client*.

### **-m**

Display a media report instead of the default save set report (in other words, a report about the media containing save sets, not the save sets themselves).

### **-N** *name*

Restricts the reported output to the media and/or save sets pertaining to the specified save set *name*.

**-o** *order*

Sort the output displayed in the specified order. Before displaying the save sets, they are sorted by various fields. Numeric fields are sorted least to greatest, other fields are sorted alphabetically. *order* may be any combination of the letters **celmontR**, representing client, expiration date, length, media name, name of save set, offset on media (file and record number), time, and Reverse, respectively. The default sorting order for save set reports is **mocntl**. The offset fields (file and record) are only considered when the **-V** option has been selected and for custom reports that show save set section (fragment) information. When applied to **-m** media-only reports, the length is the amount used on the volume, the expiration date applies to the volume, the time is the last time the media was accessed, and the other order flags are ignored.

**-q** *queryspec*

Adds the given query constraint to the list of constraints on the current query. Multiple **-q** options may be given. See the **Custom Queries and Reports** section for the syntax of the *queryspec*.

**-r** *reportspec*

Appends the given report specification to the list of attributes to be displayed for the current query. Multiple **-r** options may be given. See the **Custom Queries and Reports** section for the syntax of the *reportspec*.

**-s** *server*

Display volume and save set information from the Legato Storage Manager system on *server*. See **nsr(8)** for a description of server selection. The default is the current system.

**-t** *time*

Restricts the reported information to the media and/or save sets pertaining to the save sets created on or after *time*. See **nsr\_getdate(3)** for a description of the recognized time formats. The default is 'yesterday'.

**-v**

Turn on the verbose display reports, described previously.

**-B**

Run the canned query to report bootstraps which have been generated in the past five weeks, as described previously. This option is used by **savegrp(8)** when saving the server's index and bootstrap.

**-S**

Displays a long, multi-line save set report, as described previously.

**-V**

Displays additional verbose report output, as described previously.

**-X**

Prepare a summary report, as described previously.

## Custom Queries and Reports

The custom query and report options of **mminfo** allow one to generate media and save set reports matching complex constraints without resorting to pipelines and scripts. This section describes the syntax of custom query and report specifications, and gives some simple examples. Additional examples are shown in the following **Examples** section.

The custom query option, **-q *queryspec***, is an extension to the short-hand query options, such as **-c *client***, which allow you to make queries based on almost any media or save set attribute in the database, and allow various comparisons in addition to the simple equality comparison provided by the short-hand options. The format of a *queryspec* is

```
[!] name [ comp value ] [ , ... ]
```

where *name* is the name of a database attribute, listed in the following table, *comp* is a valid comparator for the attribute, from the set '>', '>=', '=', '<=', '<', and *value* is the value being compared. Leading and trailing spaces can be used to separate the individual components of the specification. The comparator and value must be specified for all but flag attributes. Generally numeric attributes allow all five comparators, and character string attributes generally only allow equality. When comparing flags, whose values are normally 'true' and 'false', one may alternatively use the '[! ] *name*' syntax. The '!*name*' form is equivalent to '*name=false*', and '*name*' by itself is equivalent to '*name=true*'. The comparisons in the specification are separated by commas. If a time or a string contains commas, you must quote the value with single or double quotation marks. Quotation marks are escaped within a string by repeating them. The following is a valid string comparison:

```
name="Joe's daily, ""hot"" Save Set"
```

Note that command line shells also interpret quotation marks, so you will need to enclose the entire query within quotation marks, and quote the single value inside the query, possibly with a different kind of quote, depending on the shell. Except for multiple character string values, explained in the following sections, all of the specified constraints must match a given save set and/or media volume before a line will be printed in the report. Multiple **-q** options may be specified, and may be

combined with the short-hand query constraints `-c`, `-N` and `-t`. The order of the query constraints is unimportant.

Numeric constraints, except for identifiers (volume, save set and clone identifiers) allow ranges to be specified, and all character string constraints allow multiple possible values to be specified. Note that times and levels are considered to be numeric values, not character strings. The upper and lower bounds of a numeric range are specified as two separate constraints. For example,

```
%used>20,%used<80
```

matches volumes that are between 20% and 80% used. Each possible value of a given character string attribute is specified as a separate equality constraint. For example,

```
client=pegasus,client=avalon
```

matches save sets from the client 'pegasus' or the client 'avalon'.

The custom report option, `-r reportspec`, allows one to specify exactly which media and save set attributes should be shown in the report, the order of the columns, the column widths, and where line breaks should be placed. The format of a *reportspec* is

```
name [ (width) ] [ , name [ (width) ] ... ]
```

where *name* is the name of a database attribute, listed as follows, and the optional *width*, enclosed in parentheses, specifies how wide the column should be. Leading and trailing spaces are ignored. The default column width depends on the attribute; default widths are also shown in the following table. Multiple `-r` options may be specified. The order of the columns in the report will be left to right, and correspond to the order of the attribute names specified. Each line of information displayed will contain all of the data requested (you can cause line breaks within a logical line by using the *newline* attribute name). If a value does not fit in the requested column width, subsequent values in the line will be shifted to the right (values are truncated at 256 characters).

The following table lists all of the recognized attribute names, their valid range of query values (or 'NA' for attributes that are only valid for report specifications), their default column width in characters (or 'NA' for flag attributes that are only valid for query specifications), and a short description. Numeric attributes (shown as *number* in the valid range column of the table) can be specified using any of the comparators listed previously, and can be used in range comparisons. The *=number* attributes have the same characteristics as *number* attributes, with the exception that only numeric equality comparisons are allowed. *Flag* attributes have the values 'true' or 'false', only apply as query constraints, and have corresponding flag

summary strings for report specifications. *Time* attributes are specified in **nsr\_getdate(3)** format and are otherwise treated as numeric attributes (note that you will need to quote times that contain commas). The special time ‘forever’, when used as an expiration date, means a save set or volume will never expire. The special time ‘undef’ is displayed when the time is undefined. Undefined expiration dates mean a save set will expire depending on its client’s retention and browsability policies. When Times are displayed as *MM/DD/YY HH:MM:SS* for numeric month, day year (last two digits), hours, minutes, and seconds, respectively. If the column is very narrow (less than 14 characters), only the date is shown, columns less than 17 characters wide drop the seconds, columns 17 or more characters wide print the full date. *Size* and *kbsize* attributes may have a scale factor appended to them, ‘KB’ for kilobytes, ‘MB’ for Megabytes, ‘GB’ for Gigabytes and ‘TB’ for Terabytes. The default scale (when no scale is explicitly specified) on query constraints for *size* attributes is bytes; the default for *kbsize* attributes is kilobytes. The scale varies in reports, depending on the actual value. *String* attributes may be any arbitrary character string, enclosed in quotation marks if necessary, as described previously in the query syntax paragraph.

| attribute   | value               | width | description  |
|-------------|---------------------|-------|--|
| <b>name</b> | <b>range</b>        |       |  |
| space       | NA                  | 1     | White space before the next column.  |
| newline     | NA                  | 1     | Line break(s) within a logical line.Width is actually the number of newlines desired.  |
| volume      | string              | 15    | The volume name.   |
| valid       | =number             | 11    | The unique volume identifier.  |
| barcode     | string              | 15    | The volume barcode, when set.  |
| family      | string              | 4     | The media family (eg. tape, disk).   |
| type        | string              | 7     | The media type (eg. 8mm, optical).   |
| volflags    | NA                  | 5     | Volume summary flags, d and r,for dirty (in use) and read-only.  |
| state       | NA                  | 3     | Volume state summary, <b>E</b> , <b>M</b> , <b>X</b> and <b>A</b> , meaning eligible for recycling, manually-recyclable, both, and archive or migration volumes, respectively. |
| full        | flag                | NA    | Matches full volumes.  |
| inuse       | flag                | NA    | Matches in-use (dirty) volumes.  |
| volrecycle  | flag                | NA    | Matches recycleable volumes.   |
| readonly    | flag                | NA    | Matches read-only volumes.   |
| manual      | flag                | NA    | Matches manually-recyclable volumes.   |
| pool        | string              | 15    | The pool containing the volume.  |
| location    | string              | 15    | The volume’s location.   |
| capacity    | size                | 8     | The volume’s estimated capacity.   |
| written     | kbsize              | 7     | Kbytes written to volume.  |
| %used       | number<br>or ‘full’ | 5     | Estimated percentage used, or ‘full’ for volumes marked as full.   |

|             |                                   |    |  |
|-------------|-----------------------------------|----|--|
| read        | kbsize                            | 8  | Kbytes read (recovered) from the volume.   |
| next        | number                            | 5  | Next media file for writing.   |
| nrec        | number                            | 5  | Next media record for writing.   |
| volaccess   | time                              | 9  | Last time volume was accessed (written to or labeled). Old servers do not provide this value reliably.   |
| volexp      | time                              | 9  | Volume expiration date.  |
| olabel      | time                              | 9  | The first time the volume was labeled.   |
| labeled     | time                              | 9  | The most recent time the media volume was (re)labeled.   |
| mounts      | number                            | 6  | Number of times the volume was mounted explicitly (reboots do not count).  |
| recycled    | number                            | 4  | Number of times the volume was relabeled.  |
| avail       | NA                                | 3  | Summary of volume availability, current valid values, <b>n</b> meaning nearline (ie. in a jukebox), and <b>ov</b> meaning the volume is being managed by SmartMedia.   |
| near        | flag                              | NA | Matches nearline volumes.  |
| smartmedia  | flag                              | NA | Matches volumes managed by SmartMedia.   |
| metric      | number                            | 6  | Volume speed and desirability metric (unused by existing servers).   |
| savesets    | NA                                | 6  | Number of save sets on a volume.   |
| name        | string                            | 31 | The save set name.   |
| savetime    | time                              | 9  | The save time.   |
| nsavetime   | NA                                | 11 | The save time, printed as seconds since 00:00:00 GMT, Jan 1, 1970.   |
| ssid        | =number                           | 11 | The unique save set identifier.  |
| level       | 0..9,<br>full, incr,<br>or manual | 5  | The backup level. Manual backups are printed as blank column migration values in reports.  |
| client      | string                            | 11 | The client name for the save set.  |
| attrs       | NA                                | 31 | The extended save set attributes.  |
| psid        | =number                           | 11 | When part of a save set series, the previous save set identifier in the series, zero for the first or only save set in a series.   |
| ssflags     | NA                                | 7  | The save set flags summary, one or more characters in the set <b>CvrSENriIF</b> , for continued, valid, purged (recoverable), scanned-in (rolled-in), eligible for recycling, NDMP generated, raw, incomplete,in-progress and finished (ended),respectively. |
| continued   | flag                              | NA | Matches continued save sets.   |
| recoverable | flag                              | NA | Matches recoverable (purged) save sets.  |
| ssrecycle   | flag                              | NA | Matches recyclable save sets.  |
| incomplete  | flag                              | NA | Matches incomplete save sets.  |

|            |         |    |  |
|------------|---------|----|--|
| rolledin   | flag    | NA | Matches rolled-in save sets.   |
| NDMP       | flag    | NA | Matches NDMP save sets.  |
| raw        | flag    | NA | Matches raw save sets.   |
| valid      | flag    | NA | Matches valid save sets. All save sets are marked 'valid' by current servers.                                    |
| sumflags   | NA      | 3  | Per-volume save set summary flags, as described for the <b>-v</b> report.  |
| fragflags  | NA      | 3  | Per-section save set summary flags, as described for the <b>-V</b> report.                                       |
| totalsize  | number  | 11 | The total save set size.   |
| nfiles     | number  | 5  | The number of the client's files in the save set.  |
| ssexp      | time    | 9  | The save set's expiration time, 'undef' means it depends on client policies.                                     |
| copies     | number  | 6  | The number of copies (instances or clones) of the save set, all with the same save time and save set identifier. |
| cloneid    | =number | 11 | The clone identifier of one copy.  |
| clonetime  | time    | 9  | The time a copy was made.  |
| clflags    | NA      | 5  | The clone flags summary, from the set <b>ais</b> for aborted, incomplete and suspect (read error), respectively. |
| suspect    | flag    | NA | Matches suspect save set copies, copies that had errors during file recovery.                                    |
| annotation | string  | 31 | Matches the string to the save set's annotation.   |
| first      | number  | 11 | The offset of the first byte of the save set contained within the section.                                       |
| last       | NA      | 11 | The calculated offset of the last byte of the save set contained within the current section.                     |
| fragsize   | NA      | 7  | The calculated size of the current section of the save set.  |
| sumsize    | NA      | 7  | The calculated total size of all of the sections of the save set on this volume.                                 |
| mediafile  | number  | 5  | The media file number containing the current section of the save set.  |
| mediarec   | number  | 5  | The media record number where the first bytes of the save set are found within the current media file.           |
| mediamark  | number  | 5  | The absolute positioning data for the current section (not used by existing servers).                            |
| ssaccess   | time    | 9  | The last time this section of the save set was access (for backup or recover).                                   |

## Examples

In the following examples, the equivalent short-hand and custom versions of the report are shown, when a short-hand option exists for a given report or query.

Display all bootstraps generated in the previous five weeks, as reported by **savegrp(8)**:

```
mminfo -B  
mminfo -N bootstrap -t '5 weeks ago' -ot  
-r 'savetime(17),space,level(6),ssid'  
-r 'mediafile(6),mediarec(8),space(3),volume'
```

Display information about all of the volumes:

```
mminfo -m  
mminfo -a -r 'state,volume,written,%used,read,space,volexp'  
-r 'mounts(5),space(2),capacity'
```

Display media information from volumes **mars.001** and **mars.002**:

```
mminfo -m mars.001 mars.002  
mminfo -m -q 'volume=mars.001,volume=mars.002'
```

Display all save sets named **/usr**:

```
mminfo -N /usr  
mminfo -q name=/usr
```

Display save sets named **/usr**, generated by client **venus**, in the past week:

```
mminfo -N /usr -c venus  
mminfo -q 'name=/usr,client=venus'
```

Display save sets named **/usr**, generated by client **venus**, on volume **mars.001**:

```
mminfo -N /usr -c venus mars.001  
mminfo -q 'name=/usr,client=venus,volume=mars.001'
```

Display a media report of all volumes written on in the past week:

```
mminfo -m -t 'last week'  
mminfo -m -q 'savetime>=last week'
```

Display a media report of all non-full volumes, showing the percent-used, pool and location of each volume:

**mminfo -a -r 'volume,%used,pool,location' -q '!full'**

Display a media report similar to the **-m** report but showing the barcode instead of the volume label:

**mminfo -a -r 'state,barcode,written,%used,read,space,volexp'  
-r 'mounts(5),space(2),capacity'**

Display a verbose list of the instances of all save sets with more than one copy, sorted by save time and client name:

**mminfo -otc -v -q 'copies>1'**

Display all archive save sets with an annotation of "project data" for the past four months.

**mminfo -q 'annotation=project data'  
-r "volume,client,savetime,sumsize,ssid,name,annotation"  
-t 'four months ago'**

## Files

**/nsr/mm/mmvolume**

The save set and media volume databases (actually accessed by **nsrmmdbd(8)**).

**See Also:** [nsr\\_getdate\(3\)](#), [nsr\\_layout\(5\)](#), [nsradmin\(8\)](#),  
[nsrmmdbd\(8\)](#), [recover\(8\)](#), [savegrp\(8\)](#), [scanner\(8\)](#)

## Diagnostics

**no matches found for the query**

No save sets or volumes were found in the database that matched all of the constraints of the query.

**invalid volume name 'volname'**

The volume name given is not in a valid format. Note that volume names may not begin with a dash. Queries that match no volumes will return the error 'no matches found for the query'.

**only one of -m, -B, -S, -X or -r may be specified**

Only one report can be generated at a time. Use separate runs of **mminfo** to obtain multiple reports.

**invalid sorting order specifier, choose from ‘*celmnotR*’**

Only letters from *celmnotR* may be used with the `-o` option.

**only one `-o` allowed**

Only one sorting order may be specified.

**only one `-s` allowed**

Only one server can be queried at one time. Use multiple runs of `mminfo` to obtain reports from multiple servers.

**Out of Memory**

The query exhausted available memory. Try issuing it again, using the sorting order `-om`, or make the query more restrictive (for example, list specific volumes, clients, and/or save set names).

**invalid value specified for ‘*attribute*’**

The value specified is either out of range (for example, a negative number for a value that can only take positive numbers), the wrong type (an alphabetic string value specified for a numeric attribute), or just poorly formatted (for example, non-blank characters between a close quote and the next comma or a missing close quote).

**value of ‘*attribute*’ is too long**

The value specified for *attribute* is longer than the maximum accepted value. Query attributes must have values less than 65 characters long.

**non-overlapping range specified for ‘*attribute*’**

The range specified for *attribute* is a non-overlapping numeric range, and cannot possibly match any save set or volume in the database.

**unknown query constraint: *attribute***

The given query *attribute* is not valid. See the **Custom Queries and Reports** table for a list of all valid attribute names.

**need a value for query constraint ‘*attribute*’**

The *attribute* is not a flag, and must be specified in the ‘*name comparator value*’ format.

**constraint ‘*attribute*’ is only valid for reports**

The *attribute* specified for a query may only be used in report (`-r`) specifications. Calculated values, flag summaries, save set extended attributes, and formatting tools (**space** and **newline**) may not be used in queries.

**invalid comparator for query constraint ‘*attribute*’**

The comparator used is not valid for the given *attribute*. See the **Custom Queries and Reports** section for a list of the valid comparators for *attribute*.

**query constraint 'attribute' specified more than once**

The given attribute was specified more than once with the same comparator, and is not a string attribute (string attributes can match one of several specific values).

**unknown report constraint: attribute**

The given report *attribute* is not valid, see the **Custom Queries and Reports** table for a list of all valid attribute names.

**constraint 'attribute' is only valid for queries**

The *attribute* specified for a report is a flag matching attribute and may only be used in query (-q) specifications. See the **Custom Queries and Reports** table for the appropriate flag summary attribute that one may use in reports of a given flag.

**column width of 'attribute' is invalid**

The width specified for *attribute* is out of range. Column widths must be positive numbers less than 256.

**missing close parenthesis after report constraint 'attribute'**

The width of *attribute* is missing a close parenthesis.

**missing comma after report constraint 'attribute'**

There are non-blank characters after the width specification for *attribute* without any comma preceding them.

**No data requested, no report generated**

The given report specification contains only formatting, no data attribute names.

## Limitations

You cannot specify save set extended attributes as query constraints.

You cannot list several possible equality matches for numbers, only for strings.

Some queries, namely those that are not highly selective (few query constraints) and use a sorting order where the volume name is not the primary sort key, still require **mminfo** to retrieve the entire database before printing any of it. Such queries use up large amounts of memory in **mminfo**, but not, as was the case with older versions, in **nsrmmdbd**.

You cannot make a report that shows save set or media instances and a summary without running **mminfo** at least twice.

You cannot specify query constraints that compare database attributes with each other.

You cannot make a report that uses -B flag with -c flag.

## mmlocate(8)

### Name

**mmlocate** – Legato Storage Manager media location reporting command

### Synopsis

```
mmlocate [ -s server ] [ -l { -n volname | -i valid | location } ] [ -L ] [ -d location ]  
[ -c { -n volname | -i valid } ] [ -u { -n volname / -i valid / location } ]
```

### Description

The **mmlocate** command is used to access and manage the volume location information contained in the media database. The information contained in a volume's location field is meant to give the user an idea of where the volume can physically be found. Other Legato Storage Manager commands will display the location along with the volume name (see the *versions* subcommand of **recover**(8)). Any user can use this command with the **-l** (default) or **-L** options. The **-c**, **-d** and **-u** options are limited to Legato Storage Manager administrators (see **nsr**(8)). **-l** is assumed by **mmlocate** if a **-L**, **-c**, **-d** or **-u** option is not specified.

Running **mmlocate** without any arguments lists all volumes and their locations for the specified server (if you do not specify a server, the current host is used).

Note that each time **nsrjb0** moves a piece of media inside a jukebox, the location of a volume is set to the name of the jukebox. When using storage nodes, the name of the jukebox is used to indicate on which node the volume can be mounted. Hence, the first portion of this field containing the jukebox name should not be changed. When using volumes on a storage node that are not contained within a jukebox, this field can be used to indicate on which node a volume should be mounted, by giving it a value of any remote device on that node. See **nsr\_storage\_node**(5) for additional detail on storage nodes.

### Options

**-c**

Clears the location field for the specified volume.

**-d** *location*

Deletes all volumes associated with the specified location. A confirmation prompt appears prior to the deletion of each volume.

**-i** *valid*

Restricts the **mmlocate** operation to the specified volume ID (*valid*).

**-l**

Lists entries. Performs a database query using the supplied *volume name*, *volume ID* or *location*.

If a volume name or volume id is given, then only the volume's location information is displayed. If a location is provided, then only volumes in that location are displayed. When the **-l** option is used without specifying any other options (volume name, volume id, or location), volumes without a set location are displayed.

**-L**

Lists all *locations* found in the database.

**-n** *volname*

Restricts the operation to the volume name (*volname*) listed.

**-s** *server*

Accesses the *server*'s media database.

**-u**

Updates the location for a volume. Locations are limited to a maximum length of 64 characters. The **-n** *volname* or **-i** *valid* and *location* options must be used in conjunction with the **-u** option.

## Examples

Update the media database to show that volume Offsite.011 is now at location 'Media Vault'

```
"mmlocate -u -n Offsite.011 'Media Vault'
```

Delete volumes at location 'Media Shelf 6'

```
"mmlocate -d 'Media Shelf 6'
```

Delete location information for volume NonFull.001

```
"mmlocate -c -n NonFull.001
```

List the location of volume NonFull.001

```
"mmlocate -n NonFull.001
```

List all volumes stored in the location 'Media Vault'

**"mmlocate 'Media Vault'**

## Files

**/nsr/mm/mmvolume**

The media database.

**See Also:** [nsrmm\(8\)](#), [mminfo\(8\)](#), [nsr\(8\)](#), [recover\(8\)](#), [nsr\\_storage\\_node\(5\)](#)

## Diagnostics

**Server *server* does not support remote update operations...**

If you are running **mmlocate** against an old server, you are not allowed to use the **-u** or **-c** options. You must login to that server and run the **mmlocate** program there.

# mmpool(8)

## Name

**mmpool** – Legato Storage Manager media pool reporting command

## Synopsis

```
mmpool [ -s server ] [ volume... ]  
[ -d pool ] [ -l pool ] [ -L ]
```

## Description

The **mmpool** command is used to access pool information stored in the Legato Storage Manager server's media database. This command can also be used to delete all the volumes in a particular pool. If you specify one or more volume names with the **mmpool** command, the report shows the pool to that each named volume belongs to. By default, all volumes and their pools are displayed.

You cannot change the pool to which a volume belongs, without relabeling the volume, which destroys all data stored on the volume. Pools are configured through a Legato Storage Manager administration tool, such as **nwadmin**(8) or **nsradmin**(8). These tools are used to create and modify unique *pool* (see **nsr\_pool**(5)) resources.

## Options

**-d pool**

Deletes all volumes for the given pool. The user will be prompted for deletion of each volume.

**-l pool**

Lists all volumes and the pools to which they belong. If a **pool** is specified, **mmpool** only list the volumes belonging to that pool.

**-L**

Lists the names of all pool resources configured on the server.

**-s server**

Specifies the Legato Storage Manager server to act upon. See **nsr**(8) for a description of server selection.

## Files

**/nsr/mm/mmvolume (UNIX)**

The media database on the server.

**See Also:** [nsr\(8\)](#), [nsr\\_device\(5\)](#), [nsr\\_pool\(5\)](#), [nsradmin\(8\)](#),  
[nsrmm\(8\)](#), [nwadmin\(8\)](#)

## mmrecov(8)

### Name

**mmrecov** – Recover a Legato Storage Manager server’s on-line index and media index

### Synopsis

**mmrecov** [ **-q** | **-v** ]

### Description

The **mmrecov** command is used in recovering from the loss of a Legato Storage Manager server’s critical files including the server’s index, the media index, and the server’s resource files. Typical events causing such disasters are accidental removal of these files by a user or a serve disk problem on the Legato Storage Manager server itself. The **mmrecov** command is used to recover the indexes and other files. See **nsr\_crash(8)** for a discussion of general issues and procedures for Legato Storage Manager client and server recovery.

**Mmrecov** is used to recover the Legato Storage Manager server’s on-line file and media index from the media (backup tapes or disks) when either of the server’s online file or media index has been lost or damaged. Note that this command *overwrites* the server’s existing online file and media index. The **mmrecov** command is not used to recover Legato Storage Manager clients’ online indexes; you can use normal recover procedures for this purpose.

The Legato Storage Manager system must be fully installed and correctly configured prior to using this command. If any of the Legato Storage Manager software is lost, re-install Legato Storage Manager from the distribution files before you run **mmrecov**. Use the same release of Legato Storage Manager, and install it in the same location as it was before the software was lost.

**Mmrecov** program works in two phases. First, it extracts the contents of a *bootstrap* save set, which contains the media and online file indexes. The online file index only contains one entry, for the online file index itself. In the second phase, **mmrecov** will run **recover(8)** to completely recover the server’s online file index. This last phase is performed in the background, so the operator can respond to subsequent media mount requests.

When **mmrecov** is started, it will ask for the device from which the bootstrap save set will be extracted. Next, it will ask for the *bootstrap* save set identifier. This number is found in the fourth column (labeled *ssid*) of the last line of the bootstrap information sheet printed by **savegrp**, as the following example shows:

```
Jun 17 22:21 1999 mars's Legato Storage Manager bootstrap information Page 1
```

| date    | time     | level | ssid     | file | record | volume        |
|---------|----------|-------|----------|------|--------|---------------|
| 6/14/99 | 23:46:13 | full  | 17826163 | 48   | 0      | mars.1        |
| 6/15/99 | 22:45:15 | 9     | 17836325 | 87   | 0      | mars.2        |
| 6/16/99 | 22:50:34 | 9     | 17846505 | 134  | 0      | mars.2 mars.3 |
| 6/17/99 | 22:20:25 | 9     | 17851237 | 52   | 0      | mars.3        |

In the previous example, the *ssid* of the most recent bootstrap save set is **'17851237'**. If you are cloning save sets, your bootstrap save set is also cloned, and you need to use the *second* to last save set. See the **Recovering from Clone Media** section for an example of bootstrap information with cloned save sets. Next, **mmrecov** prompts for the file and record location of the bootstrap save set. Both values may default to zero if they are not known. The file and record locations are the fifth and sixth columns of the bootstrap information sheet. In the previous example, the values for the file and record locations are 52 and 0, respectively. Finally, **mmrecov** will ask that the volume (**'mars.3'** in the previous example) containing the selected *bootstrap* save set be inserted into the specified device. All of the *ssid*, file location, record location, and the physical volume must be determined by the user from the printed sheet, since **mmrecov** has no way of determining this information. On the other hand, if the volume containing the bootstrap is not known, the **-B** option of **scanner(8)** can be used to determine the file and record locations.

If the *bootstrap* save set spans more than one volume, multiple volume names are printed. The order printed is the order required by **mmrecov**. In the previous example, the third save set produced on 6/16/92 begins on volume **'mars.2'** and spans to volume **'mars.3'**. If a bootstrap save set spans volumes, **mmrecov** will ask for the name of the device where the next volume has been loaded when an end-of-volume occurs. The volume is then scanned, and the bootstrap save set extracted.

After the volume scan completes, **mmrecov** will complete. A **recover** will be running in the background, reconstructing a complete index from the save sets generated by the server's save schedule. Since the save sets may be spread across multiple volumes, **nwadmin(8)** or **nsrwatch(8)** should be run, and the volumes mounted as they are requested.

When the **recover** completes, the message **"The index is now fully recovered"** is displayed. **recover** can now be used in its normal interactive browsing mode, to

recover other server files, including other Legato Storage Manager client on-line file indexes. Once a Legato Storage Manager client's index is recovered, that client can start recovering its files.

As stated earlier, the Legato Storage Manager resource files are saved as part of the bootstrap save set. If your resource files were also deleted, you may quickly replace them by copying or moving them from `/nsr/res.R` to `/nsr/res`. Before restoring them to `/nsr/res`, the daemons must be shut down (see `nsr_shutdown(8)`).

Sometimes, it is necessary to recover the Legato Storage Manager server onto a new computer, for example, after a major hardware failure. When this occurs, the Legato Storage Manager Licensing software will detect the move. Once the Legato Storage Manager server has been moved to a new computer, it must be re-registered with Customer Support within 15 days of the move, or the server will disable itself. After disabling itself, you will only be able to recover files; new backups cannot be performed until the server is re-registered. Notifications will be sent by the NSR Registration notification, warning of the need to re-register the product.

## Recovering from Clone Media

If you are running `mmrecov` with clone media only, for example, at a remote site, you will need to perform the recovery using a slightly different method. When selecting the bootstrap identifier, make sure that you are using the information associated with the cloned save set: the last save set listed in the bootstrap information. Consider the following list of save sets:

Jun 17 22:21 1999 mars's Legato Storage Manager bootstrap information Page 1

| date    | time     | level | ssid     | file | record | volume   |
|---------|----------|-------|----------|------|--------|----------|
| 6/14/99 | 23:46:13 | full  | 17826163 | 48   | 0      | mars.1   |
| 6/14/99 | 23:46:13 | full  | 17826163 | 12   | 0      | mars_c.1 |
| 6/15/99 | 22:45:15 | 9     | 17836325 | 87   | 0      | mars.2   |
| 6/15/99 | 22:45:15 | 9     | 17836325 | 24   | 0      | mars_c.2 |
| 6/17/99 | 22:20:25 | 9     | 17851237 | 52   | 0      | mars.3   |
| 6/17/99 | 22:20:25 | 9     | 17851237 | 6    | 0      | mars_c.3 |

In the previous example, the `ssid` of the most recent bootstrap save set is `'17851237'`. The cloned save set resides on `mars_c.3` and the values for the file and record locations are 6 and 0, respectively.

Once **mmrecov** recovers the bootstrap save set, it will start pursuing other pieces of the server's client index to complete the recovery. The cloned bootstrap knows about the original and cloned volumes. If all clone volumes needed are online when the index recovery proceeds, **mmrecov** will complete on its own.

If some of the volumes are not online, then **mmrecov** will attempt to recover the index from the original volume it was backed up to, and therefore request the original media. In the previous example bootstrap information, *mars\_c.1* and *mars\_c.3* would all need to be online. If volume *mars\_c.3* was the only volume online, then **mmrecov** would also request *mars.1*. To finish recovering the server's index in this case, you need to perform the following steps:

1. Note what volumes are needed for recovery and delete them from the media database. **nwadmin(8)** or **nsrwatch(8)** lists out the volumes needed for recovery in the *Pending* messages panel. Use **nwadmin(8)** or **nsrmm(8)** to delete the volumes from the media database.

Given the scenario in the previous example where only *mars\_c.3* was mounted, we would have to delete *mars.1* from the media database, using the command **nsrmm -d mars.1**.

2. Restart the server to stop the index recovery in progress.

Use **nsr\_shutdown(8)** to bring the server down. Run **nsrd(8)** to start the server again.

3. Recover the server's index using recover by save set.

Use **mminfo(8)** to determine the save set id of the last index backup, for example:

```
mminfo -s server -c server -N /nsr/index/server -v -o t
```

Note that the **-N** option should be given the full path to the index if it is located in a different location. For example, if */nsr/index/server* is a symlink to */disk1/nsr/index/server*, you should specify **-N /disk1/nsr/index/server**. The save set id to use for recover by save set follows the size field in the **mminfo(8)** output. In the following example output from **mminfo**, the save set id is **'18196'**

```
mars_c.002   mars      12/27/96 14:06:32 138 KB    18196 cb    9 /nsr/index/mars
```

Once you have the save set id, you can run **recover(8)**. Given this information, one would run the following command:

```
recover -s server -S 18196
```

When the **recover** completes, the message **"The index is now fully recovered"** is displayed.

## Options

- q**  
Quiet. Display only error messages.
- v**  
Verbose. Generates debugging information.

## Files

- /nsr**  
If this was a symbolic link when the bootstrap save set was created, it is recovered (recreated) unconditionally by **mmrecov**.
- /nsr/res**  
This directory and its contents are saved as part of the bootstrap save set. **Mmrecov** restores this directory, and then renames it to **/nsr/res.R**. The original directory is temporarily renamed to **/nsr/res.org** while the bootstrap save set is being recovered.
- /nsr/mm/mmvolume**  
The Legato Storage Manager server's media index saved as part of the bootstrap save set, and unconditionally recovered by **mmrecov**.
- /nsr/index/servername/db**  
The Legato Storage Manager server's on-line file index saved as part of the bootstrap save set, and unconditionally recovered by **mmrecov**.

## Bugs

The name **mmrecov** is misleading, as a result mmrecov is often used when it is not needed. A name like "recover\_server\_index\_or\_media\_index\_when\_either\_is\_missing" is more descriptive. If the server's on-line index and media index are intact, then any part of the bootstrap save set contents are recoverable using normal recover procedures.

To recover files that are not in the on-line file index (for example, files saved after the last run of **savegrp**), **scanner** must be used to rebuild the media and on-line file indexes from the contents of the volumes generated between the time of the last run of **savegrp** and the loss of the original index.

**See Also:** [mminfo\(8\)](#), [nsr\\_crash\(8\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nsr\\_client\(5\)](#), [nsr\\_schedule\(5\)](#), [nsr\\_shutdown\(8\)](#), [recover\(8\)](#), [save\(8\)](#), [savefs\(8\)](#), [savegrp\(8\)](#), [scanner\(8\)](#), [nsrindexasm\(8\)](#), [nsrmm\(8\)](#), [nsrmmdbasm\(8\)](#), [nwadmin\(8\)](#), [nsrwatch\(8\)](#), [nsr\\_getdate\(3\)](#)

## Diagnostics

### **"The index for *server* was NOT fully recovered"**

There was an error recovering all or part of the *server*'s index. Look back through the information displayed to find out what was not recovered (one or more additional error messages will be mixed in with other **mmrecov** status messages). You may need to retry the **mmrecov**, for example, in the case of a temporary resource limitation or media error, or you may need to resort to recovering an older version of the index, in the case a permanent media error.

# nsr(5)

## Name

**nsr** – Legato Storage Manager directive file format

## Description

This man page describes the format of directive files. These files are interpreted by **save(8)** and Application Specific Module (ASM) programs, during Legato Storage Manager backup processes. This format is also used in the *directive* attribute of the **nsr\_directive(5)** resource.

Directives control how particular files are to be backed-up, how descendent directories are searched, and how subsequent directives are processed. For each file backed-up, any ASM information required to recover that file is also backed-up. This enables **recover(8)**, or any ASM directly invoked, to recover a file correctly, even if the current directives have changed since the file was backed-up. See **uasm(8)** for a general description of the various ASMs.

The directive file in each directory is parsed before anything in that directory is backed up, unless Legato Storage Manager is being run in *ignore* mode. Each line of a directive file, and each line of the directive attribute, contains one directive. Any text after a "#" character until the end of the line is treated as a comment and discarded. Directives appear in one of three distinct forms:

```
[+] ASM [args ...] : pattern ...
save environment
<< dir >>
```

The three forms are referred to as ASM specifications, *save environment* directives, and << *dir* >> directives, respectively.

Use ASM specifications (name and any arguments), to specify how files or directories with a matching *pattern* are backed-up. When a *pattern* matches a directory, the specified ASM is responsible for handling the directory and its contents. Any *pattern* or ASM arguments requiring special control or white space characters should be quoted using double quotation marks (").

A colon (:) is used as the separator between the ASM specification (and any arguments) and the *pattern* specification list. The *pattern* list for each ASM specification consists of simple filenames or patterns. The *pattern* cannot be "." and must not contain any "/" characters (all names must be within the current directory). The string "." can be used to match the current directory. Standard **sh(1)** file pattern

matching (\*, [...], [...], [x-y], ?) can be used to match filenames. If a "+" precedes the ASM name, then the directive is propagated to subdirectories. When a directory is first visited, it is searched for a **file**. If one is found, it is then read. Each file is only read once. When starting a save at a directory below /, any files on the normalized path of the current working directory are read before any files are saved to catalog any propagated directives.

The following algorithm is used to match files to the appropriate ASM specification. First the file in the current directory (if any) is scanned from top to bottom for an ASM specification without a leading "+" whose *pattern* matches the filename. If no match is found then the file in the current directory is re-scanned for an ASM specification with a leading "+" whose *pattern* matches the filename (for clarity, we recommend placing all propagating "+" directives after all the non-propagating directives in a file). If no match is found, then the file found in the "." directory (if any) is scanned from top to bottom looking for a match with an ASM specification that has a leading +. This process continues until the file in the "/" directory (if any) is scanned. If no match is found (or a match is found with an ASM specification whose name is the same as the currently running ASM), then the currently running ASM will handle the save of the file.

Use *save environment* directives to change how ASM specifications and future files are used. The *save environment* directives do not take any file patterns. They affect the currently running ASM and subsequent ASMs invoked below this directory. There are three different possible *save environment* directives that can be used:

**forget**

Forget all inherited directives (those starting with a "+" in parent directories).

**ignore**

Ignore subsequent files found in descendent directories.

**allow**

Allow file interpretation in descendent directories.

The << *dir* >> directive can be used to specify a directory where subsequent ASM specifications from the current file should be applied. This directive is intended to be used to consolidate the contents of several files to a single location or directory. The *dir* portion of this directive must resolve to a valid directory at or below the directory containing this directive or subsequent ASM specifications will be ignored. Relative path names should be used for filenames to ensure the interpretation of subsequent ASM directives is consistent, even if a directory is mounted in a different absolute part of the file system.

There **must** be a << *dir* >> as the first directive in a directive file used in conjunction with the **-f** option to **save(8)**, **savefs(8)** or with an ASM program. Also, when << *dir* >> directives are used in this manner, whether first or later in the file, absolute path names should be used to ensure appropriate interpretation. Absolute path names should also be used for each directory specified within the **directive** attribute of the **NSR directive** resource (see **nsr\_directive(5)**).

When a << *dir* >> directive is used, subsequent directives are parsed and logged for later use. When a directory specified by *dir* is opened, any *save environment* directives specified for that directory (for example, **allow**, **ignore**, and **forget**) are processed first. If the ASM is not currently ignoring files and a local file exists, the file is read and processed. Finally, any of the non *save environment* directives specified for that directory are handled as if they were appended to the end of a file in that directory. If multiple << *dir* >> specifications resolve to the same directory, then the corresponding save directives are handled logically in "last seen first" order.

## Examples

Having a */usr/src/nsr* file containing:

```
+skip: errs *.o
+compressasm: .
```

will cause all files (or directories) located in the */usr/src* directory named **errs** or **\*.o** (and anything contained within them) to be skipped. In addition, all other files contained in the */usr/src* directory will be compressed during save and will be set up for automatic decompression on recover.

Having a */var/nsr* file containing:

```
compressasm: adm .nsr
null: * .*
```

causes all files (or directories) and their contents located within the */var* directory and anything contained within them, (except for those files located in the */var/adm* directory and the file itself) to be skipped, although all the names in the directory would be backed-up. In addition, since **compressasm** is a *searching* directive (see **uasm(8)**), the files contained within the */var/adm* directory will be compressed during back up and will be set up for automatic decompression on recover.

The following is an example of using the `./nsr` file as a master save directive file for the entire file system by using `<< dir >>` directives to consolidate the various ASM save directives to a single location:

```
# Master Legato Storage Manager directive file for this machine
<< ./ >>
# /mnt and /a are used for temporary fs mounting
# and need not be saved
  skip: mnt a
    +skip: core errs dead.letter *% *~
# Don't bother saving anything within /tmp
<< ./tmp >>
  skip: .* *
<< ./export/swap >>
  swapasm: *
# Translate all mailboxes. Also, use mailasm to save each mail file to maintain
# mail file locking conventions and to preserve the last file access time.
<< ./usr/spool/mail >>
  xlateasm: .
  mailasm: *
# Allow .nsr files to be interpreted in /nsr, even if we are currently ignoring
# .nsr files. Legato Storage Manager applications (such as nsrindexd) set up
# their own private .nsr files which save index files more intelligently.
<< ./nsr >>
  allow
# We can rebuild any .o files in /usr/src from sources except those in
# /usr/src/sys.
<< ./usr/src >>
  +skip: *.o
<< ./usr/src/sys >>
  forget
```

## Files

Save directive file in each directory.

**See Also:** [sh\(1\)](#), [nsr\\_directive\(5\)](#), [nsrindexasm\(8\)](#), [nsrmmdbasm\(8\)](#), [recover\(8\)](#), [save\(8\)](#), [savefs\(8\)](#), [uasm\(8\)](#)

## nsr(8)

### Name

NSR – Introduction and overview of Legato Storage Manager

### Description

Legato Storage Manager facilitates the backup and recovery of files on a network of computer systems. Files and file systems may be backed up on a scheduled basis. Recovery of entire file systems and single files is simplified by use of an on-line index of saved files.

Legato Storage Manager uses a client/server model to provide the file backup and recover service. At least one computer on the network is designated as the Legato Storage Manager *Server*, and the computers with disks to be backed up are Legato Storage Manager *clients*. Five daemons provide the Legato Storage Manager service, control access to the system, and provide index and media support. On the clients, there are special programs to access the file systems and communicate with the Legato Storage Manager server.

The Legato Storage Manager system has several parts. Commands and files are only briefly mentioned here; see the appropriate reference manual page for more detailed information. Each command has a manual page entry in section 8. The files and their formats are explained in section 5 manual pages.

The *Legato Storage Manager Administrator's Guide* provides information on configuring and administering a Legato Storage Manager system. It includes many examples and rationales for setting up and running a successful backup operation.

### Installation

How Legato Storage Manager is installed varies depending on the architecture of the computer upon which you're installing. See the Oracle Server Installation Guide for detailed installation instructions.

#### **lsminst**

The Legato Storage Manager installation script. The script will install both clients and servers. The **lsminst** script can also be used to remove Legato Storage Manager. Note that some systems use other methods for installing and removing Legato Storage Manager, in which case the **lsminst** script will not exist.

**nsr\_layout(5)**

Describes where Legato Storage Manager programs, files, and manual pages are installed.

## Server Daemons

Legato Storage Manager uses a client/server model to provide a backup and recover service. The following daemons encompass the server side of Legato Storage Manager.

**nsrd(8)**

The main Legato Storage Manager daemon. **nsrd** handles initial communication with clients, and starts and stops the other Legato Storage Manager server daemons.

**nsrindexd(8)**

This server daemon provides access to the Legato Storage Manager on-line index. The index holds records of saved files. The index allows clients to selectively browse and choose files to recover without having to access the backup media.

**nsrmmdbd(8)**

The media management database daemon provides an index of save sets and media. The **nsrmmdbd** daemon provides a much coarser view of the saved files than does **nsrindexd**, and therefore the resultant index is usually much smaller.

**nsrmmmd(8)**

The media multiplexor daemon provides device support for Legato Storage Manager. When more than one client is saving files, the data from each client is multiplexed. During recovery operations, the data is demultiplexed and sent back to the requesting clients. When the multiple devices are enabled, several of these daemons may be active simultaneously.

## Administration

Legato Storage Manager is administered through *resources* and *attributes*. Every resource has one or more attributes associated with it. For example, a device is a Legato Storage Manager resource type; an attribute of devices is the device *type*, for example, 4mm or 8mm. The Legato Storage Manager resource format is documented in **nsr\_resource(5)**. There is also a manual page for each Legato Storage Manager resource in section 5 of the manual.

Resource files are not normally edited by hand. Rather, a Legato Storage Manager tool (usually **nwadmin(8)** or **nsradmin(8)**) is used to modify resource files dynamically so that values can be checked and changes can be propagated automatically to the interested programs.

**nwadmin(8)**

Monitors the activity of and administers Legato Storage Manager servers. The **nwadmin** command is an X Window System application, using a Motif look and feel. The **nwadmin** command is most users' primary interface to Legato Storage Manager.

**nsradmin(8)**

A **curses(3)** based tool for the administration of Legato Storage Manager servers.

**nsrwatch(8)**

A **curses(3)** based tool to monitor the activity of Legato Storage Manager servers.

**nsrmm(8)**

Media manager command. The **nsrmm** command is used to label, mount, unmount, delete and purge volumes. Mount requests are generated by **nsrmmmd**, and displayed by **nwadmin** or **nsrwatch**. The size of the on-line user file indexes may be controlled by deleting and purging volumes.

**nsrim(8)**

Automatically manages the on-line index. Usually run periodically by **savegrp**.

**mminfo(8)**

Provides information about volumes and save sets.

**nsrck(8)**

Checks and repairs the Legato Storage Manager on-line index. It is run automatically when **nsrd** starts up if the databases were not closed cleanly due to a system failure.

**nsr\_shutdown(8)**

A shell script used to safely shut down the local Legato Storage Manager server. The **nsr\_shutdown** script can only be run by the super user.

## Saving Files

Legato Storage Manager supports both scheduled and manual saving of files and file systems. Each client may be scheduled to save all or part of its file systems. Different clients may be scheduled to begin saving at different times.

**save(8)**

A command-line-based tool used to back up a specified file or group of files. The **save** command may be run manually by users and administrators, or automatically by **savegrp**.

**nwbackup(8)**

A Motif-based tool for backing up files. The **nwbackup** command is the graphical equivalent of **save**.

**Note: LSM does not support nwbackup(8).**

**savegrp(8)**

Used to start the backup of a group of client computers. Usually started automatically by the Legato Storage Manager server. The **savegrp** command also backs up the clients' on-line file indexes, which are stored on the server. When backing up the server itself, a *bootstrap* save set is also created.

**nsrexec(8)**

The *agent* **savegrp** process, spawned by **savegrp**. The **nsrexec** command monitors the progress of Legato Storage Manager commands.

**nsrclone(8)**

The Legato Storage Manager save set/volume cloning command. Using **nsrclone**, *clones*, or exact replicas, of save sets or entire volumes can be made. Clone data is indistinguishable from the original data, except for the Legato Storage Manager media volumes upon which the data reside.

**Note: LSM does not support nsrclone(8).**

**nsrexecd(8)**

Legato Storage Manager-specific remote execution service which runs on Legato Storage Manager clients. Used by **savegrp** to start **save** and **savefs** on client computers.

**savefs(8)**

Used by **savegrp** to determine characteristics of a client, and to map the save set *All* to a the current list of all save sets on a client.

## Recovering Files

Legato Storage Manager maintains an on-line index of user files that have been saved. Users may browse the index and select files for recovery. This information is used to build an representation of the file hierarchy as of any time in the past. Legato Storage Manager then locates the correct volume and recovers the requested files.

**recover(8)**

Browses the on-line user file index and selects files and file systems to recover.

**Note: LSM does not support recover(8).**

**nwrecover(8)**

A Motif-based tool for recovering files. The **nwrecover** command is the graphical equivalent of **recover**.

**Note: LSM does not support nwrecover(8).**

**mmrecov(8)**

Used only for disaster recovery. Recovers the special *bootstrap* index and the server's on-line file index. The **recover** or **nwrecover** commands are used to recover other on-line file indexes.

**scanner(8)**

Verifies correctness and integrity of Legato Storage Manager volumes. Can also recover complete save sets and rebuild the on-line file and media indexes.

**nsr\_crash(8)**

A man page describing disk recovery techniques.

**nsrinfo(8)**

Used to generate reports about the contents of a client's file index.

## Application Specific Modules

In order to process user files in an optimal manner, Legato Storage Manager provides the ASM mechanism. Pattern matching is used to select files for processing by the different ASMs. The patterns and associated ASMs are described in **nsr(5)**. The **save** command keeps track of which ASMs were used to process a file so that **recover** may use the same ASMs to recover the file.

**uasm(8)**

UNIX file system specific save/recover module. The **uasm** man page documents the general rules for all ASMs. The **uasm** command and its man page actually comprise several additional ASM including **compressasm**, **mailasm**, and **xlteasm**, to name a few.

**nsrindexasm(8)**

Processes the on-line user file indexes.

**nsrmmdbasm(8)**

Processes the on-line media database.

## Server Location

On large networks there may be several Legato Storage Manager servers installed. Each Legato Storage Manager client command must select a server to use.

For server selection, the client commands are classified into two groups: *administration* and *operation*. The administration commands include **nwadmin**, **nsrwatch**, and **mminfo**. The operation commands include **save**, **savefs**, and **recover**. Both groups of commands accept a **-s server** option to explicitly specify a Legato Storage Manager server.

When a server is not explicitly specified, the operation commands use the following steps to locate one. The first server found is used.

1. The local computer is examined to see if it is a Legato Storage Manager server. If it is, then it is used.
2. The computer where the current directory is actually located is examined to see if it is a Legato Storage Manager server. If it is, then it is used.
3. The computer specified with the **-c** option is examined to see if it is a Legato Storage Manager server. If it is, then it is used.
4. The list of trusted Legato Storage Manager servers is obtained from the local computer's **nsrexecd(8)**. Each computer on the list is examined to see if it is a Legato Storage Manager server. The first computer determined to be a Legato Storage Manager server is used.
5. A broadcast request is issued. The first Legato Storage Manager server to respond to the request is used.
6. If a Legato Storage Manager server still has not been found, then the local computer is used.

The administrative commands only use step 1.

## Security

Before a save is allowed, there must be a **NSR client** resource created for the given client. Before a recovery is allowed, the server validates client access by checking the **remote access** attribute in the **NSR client** resource (see **nsr\_client(5)**). Earlier versions Legato Storage Manager required that connections come to the server using ports in the reserved port range. This required UNIX clients to be run with an effective uid of root at connection set up time. Newer versions of Legato Storage Manager use a stronger RPC authentication flavor and do not depend on using reserved ports and so **setuid root** operation is not required.

The **savegrp(8)** command initiates the **save(8)** command on each client computer in an NSR group by using the **nsrexecd(8)** remote save execution service. See the **nsrexecd(8)** man page for details. For backward compatibility with older versions of Legato Storage Manager, **savegrp(8)** will fall back on using the **rsh(1)** protocol for remote execution if **nsrexecd** is not running on a particular client.

Access to the NSR resources through the **nsradmin(8)** or **nwadmin(8)** commands is controlled by the **administrator** attribute on each resource. This attribute has a list of names of the users who have permission to administer that resource. Names that begin with an ampersand (&) denote netgroups (see **netgroup(5)**). Also names can be of the form *user@host* to authorize a specific user on a specific host.

## Naming and Authentication

As described previously, the NSR server only accepts connections initiated from the computers listed as clients or listed in the **remote access** list (for recovering). Since computers may be connected to more than one physical network and since each physical network connection may have numerous aliases, the following policies are used as a compromise between security and ease of use. For further information about naming in the UNIX environment, refer to **gethostent(3)**, or other documentation on name services.

A client determines its own name as follows. First the client's UNIX system name is acquired using the **gethostname(2)** system call. The UNIX system name is used as a parameter to the **gethostbyname(3)** library routine. The client declares its name to be the official (or "primary") name returned by **gethostbyname**. This name is passed to the Legato Storage Manager server during connection establishment.

A server authenticates a client connection by reconciling the connection's remote address with client's stated name. The address is mapped to a list of host names using the **gethostbyaddr(3)** library function. Next, the client's stated name is used as a parameter to **gethostbyname** to acquire another list of host names. The client is successfully authenticated if and only if there exists a common name between the two lists.

The Legato Storage Manager server maps a client's name to an on-line index database name by resolving the client's name to the official name returned by **gethostbyname**. This mapping takes place both at client creation time and at connection establishment time.

To ensure safe and effective naming, the following rules should be employed:

1. The Legato Storage Manager clients and servers should access consistent host name databases. NIS (YP) and the Domain Name System (DNS) are naming subsystems that aid in host name consistency.
2. All hosts entries for a single computer should have at least one common alias among them.
3. When creating a new client, use a name or alias that will map back to the same official name that the client computer produces by backward mapping its UNIX system name.

**See Also:** [rsh\(1\)](#), [gethostname\(2\)](#), [gethostent\(3\)](#), [netgroup\(5\)](#), [nsr\(5\)](#), [nsr\\_layout\(5\)](#), [nsr\\_resource\(5\)](#), [ypfiles\(5\)](#), [ypmake\(5\)](#), [mminfo\(8\)](#), [nsr\\_crash\(8\)](#), [nsr\\_shutdown\(8\)](#), [nsradmin\(8\)](#), [nsrck\(8\)](#), [nsrclone\(8\)](#), [nsrd\(8\)](#), [nsrexecd\(8\)](#), [nsrim\(8\)](#), [nsrindexasm\(8\)](#), [nsrindexd\(8\)](#), [nsrinfo\(8\)](#), [nsrls\(8\)](#), [nsrmm\(8\)](#), [nsrmmmd\(8\)](#), [nsrmmdbasm\(8\)](#), [nsrmmdbd\(8\)](#), [nsrwatch\(8\)](#), [nwadmin\(8\)](#), [nwbackup\(8\)](#), [nwrecover\(8\)](#), [recover\(8\)](#), [mmrecov\(8\)](#), [save\(8\)](#), [savefs\(8\)](#), [savegrp\(8\)](#), [scanner\(8\)](#), [uasm\(8\)](#).

*Legato Storage Manager Administrator's Guide*

## nsr\_archive\_client(5)

### Name

**NSR archive client** – Legato Storage Manager resource type “NSR archive client”

### Synopsis

**type: NSR archive client**

### Description

Each NSR archive client is described by a single resource of type **NSR archive client** (see **nsr\_resource(5)**). To edit the NSR archive client resources for a Legato Storage Manager server type:

```
nsradmin -c "type:NSR archive client"
```

See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program. The archive client resource may also be edited using the command **nwadmin(8)**.

This resource describes systems that are allowed to use archive services on a Legato Storage Manager server. There is only one resource for each archive client.

### Attributes

The following attributes are defined for resource type **NSR archive client**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or by selecting the details Menu Item in the View Menu for a particular window in **nwadmin(8)**. **Dynamic** attributes have values which change rapidly. **Encrypted** attributes contain data that is not displayed in its original form. The assumption is that the data is sensitive in nature and must be protected from accidental disclosure. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read-only, single string)

This attribute specifies the hostname of the Legato Storage Manager archive client.

*Example:* name: venus;

**archive users list** (read/write, string list)

This attribute specifies a list of users that are allowed to use the archive services on the client. If no users are listed, only administrators and the local root user are allowed to use the archive services on the client. A value of '\*' implies any user is allowed to archive or retrieve data. The '/' and '@' characters are not allowed as part of the username.

*Example:* archive users list: paul;

**remote access** (read/write, string list)

This attribute controls who may back up, browse, and recover a client's files. By default this attribute is an empty list, signifying that only users on the client are allowed to back up, browse, and recover its files. Additional users, hosts, and netgroups may be granted permission to access this client's files by adding their names to this attribute. Netgroup names must be preceded by an ampersand ('&'). Input of the form <user>@<host> or <host>/<user>, grants access to the client's files to the specified users. The <user> and/or <host> may be a wild card, "\*". If a username is a wild card, it means all users at the host are granted access the the client's data. When a host name is a wild card, that user on all hosts is granted access to the client's data. All users on a host may also be granted access to the client's data by just listing the host's name, <host> is equivalent to \*@<host> or <host>/\*. A plus sign, "+", grants access to the client's files to all users on any host whose root user is trusted by the server's remote command system. Note that this attribute does not override file system permissions, the user still needs the necessary file system permissions to back up, browse, or recover a file. The following example grants access to the client's data for all users that satisfy at least one of the following criteria, <username, user's hostname, server's domain> is a member of the netgroup "netadmins", the user is from the host mars, the user is from the host jupiter, the user's name is sam from host pluto, or the user's id is root from any host.

*Example:* remote access: &netadmins, mars, \*@jupiter, sam@pluto, \*/root;

**remote user** (read/write, string)

This attribute specifies the user login name the Legato Storage Manager server will use to run commands on the client. The default value is NULL, implying that 'root' should be used. When nsralist (see **nsralist(8)**) is run on the Legato Storage Manager server, the server runs commands on the client to perform an archive list.

*Example:* remote user: operator;

**password** (read/write, encrypted)

The nsralist command uses this attribute when initiating the command nsrchive on the client's computer. The command nsrchive uses the password to gain access to the files being archived. If a password is given, then the "remote user" attribute

for this resource must also be defined. This attribute does not need to be set for UNIX clients.

**executable path** (read/write, string, hidden)

This attribute specifies the path to use when the Legato Storage Manager server is executing commands on the client. When no path is specified, the "remote user's" \$PATH is used.

*Example:* executable path: /etc/nsr;

**aliases** (read/write, string list, hidden)

This is a list of aliases (nicknames) for the client computer that queries can match. If this list is empty, match on client name alone.

*Example:* aliases: mars;

**server network interface** (read/write, string, hidden)

The name of the network interface on the server to be used for archives and retrieves.

*Example:* server network interface: mars-2 ;

## Example

**Note:** The hidden options are not shown in this example.

A resource to define an archive client, called pluto:

```
type:          NSR archive client;
name:          pluto;
archive users: ;
remote access: ;
```

**See Also:** [nsr\(5\)](#), [nsr\\_directive\(5\)](#), [nsradmin\(8\)](#), [nsralist\(8\)](#), [nsrarchive\(8\)](#), [nsrretrieve\(8\)](#), [nwadmin\(8\)](#)

## nsr\_archive\_request(5)

### Name

**nsr\_archive\_request** – Legato Storage Manager resource type “NSR archive request”

### Synopsis

**type: NSR archive request**

### Description

Each NSR archive request is described by a single resource of type **NSR archive request** (see **nsr\_resource(5)**). To edit the NSR archive request resources for a Legato Storage Manager server type:

**nsradmin -c "type:NSR archive request"**

See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program. The archive request resource may also be edited using the command **nwadmin(8)**.

This resource allows administrators to set up an archive to occur later or to set up frequent archives of a set of data. The administrator can run an archive on a specified client within the next 24 hours. The archive is performed using the **nsralist(8)** command.

### Attributes

The following attributes are defined for resource type **NSR archive request**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or by selecting the details Menu Item in the View Menu for a particular window in **nwadmin(8)**. **Dynamic** attributes have values which change rapidly. **Encrypted** attributes contain data that is not displayed in its original form. The assumption is that the data is sensitive in nature and must be protected from accidental disclosure. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read/write)

This attribute specifies the name of this Legato Storage Manager archive request.

*Example:* name: Product Source Tree;

**annotation** (read/write)

This attribute contains the annotation text associated with the archive save set generated from this archive request.

*Example:* annotation: Product Release 4.1;

**status** (read/write, choice)

This attribute determines if an archive request should be run. No value implies the archive request is not scheduled. Selecting *start now* causes the archive request to be run immediately. Selecting *start later* causes the archive request to be run at the time specified by the **start time** attribute (see the following description).

*Example:* status::;

**start time** (read/write)

This attribute determines when the archive request will be run. The **status** attribute (see the previous description) must be set to *start later* for the archive request to be scheduled. The 24 hour clock format is "hours:minutes".

*Example:* start time: 3:33;

**completion time** (read/write, hidden)

This attribute indicates when the archive request completed. The format is "day-of-week month day hours:minutes:seconds year".

*Example:* "Thu Oct 22 17:00:37 1999";;

**client** (read/write)

This attribute indicates what Legato Storage Manager archive client the archive request is to be executed on.

*Example:* client: neptune;

**save set** (read/write)

The *save set* attribute lists the path names to be archived on the archive client. The names should be separated by a comma and a space (" , ").

*Example:* save set: /product/src, /accounting/db;

**directive** (read/write)

This attribute specifies the directive to use when running the archive. The default value is nothing selected. The valid choices for the directive resource are names of the currently defined 'NSR directive' resources, see **nsr\_directive(5)**.

*Example:* directive: Default with compression;

**archive pool** (read/write)

This attribute can be used to override the normal media pool selection applied to

the archive save set generated from the archive request. Selecting a pool will direct the archive to that media pool.

*Example:* archive pool: Archive;

**verify** (read/write, choice)

This attribute indicates the archive request should verify the archive. See **nsr\_archive(5)** for more information on archiving. Selecting the *Yes* choice causes the verification to occur. Selecting the *No* choice will not cause any verification. If the user also requests that the archive save set be cloned, the verification is performed on the clone since the cloning operation will have verified the original archive save set.

*Example:* verify: Yes;

**verified** (read/write, hidden)

This attribute is unused.

*Example:* verified: No;

**clone** (read/write)

This attribute controls whether the archive save set generated by the archive request is to be cloned. A value of *Yes* implies the archive save set should be cloned. A value of *No* does not imply cloning.

*Example:* clone: No;

**cloned** (read/write, hidden)

This attribute is unused.

*Example:* cloned: No;

**archive clone pool** (read/write)

This attribute indicates the archive clone media pool the archive request should use when cloning the archive save set generated by this archive request.

*Example:* archive clone pool: Archive clone;

**grooming** (read/write)

This attribute indicates any grooming actions to be taken once the archive save set generated by the archive request has been created, verified, and cloned. A value of *none* implies no action. A value of *remove* implies the files and directories specified in the **save set** attribute will be removed using the **rmdir(2)** and **unlink(2)** system calls.

*Example:* grooming: none;

**archive completion** (read/write)

A notification action to be executed to send status of the archive request to.

*Example:* archive completion: /usr/ucb/mail -s "Product Archive" systemadmin;

**log** (read/write, hidden)

This attribute contains any information pertaining to the execution of the **nsralist** command.

*Example:* log;;

## Example

**Note:** The hidden options are not shown in this example.

A resource to define an archive request, called Product:

```

type:                NSR archive request;
name:                Product Source;
annotation:         Product Release 3.0;
status:             Start later;
start time:         "2:00";
client:             space;
save set:           /product/source;
directive:         Default with compression;
archive pool:      Archive;
verify:            Yes;
clone:             Yes;
archive clone pool: Archive Clone;
grooming:          none;
archive completion: mail -s Product Source Archive productteam;

```

**See Also:** [nsr\(5\)](#), [nsr\\_directive\(5\)](#), [nsr\\_resource\(5\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#), [rmdir\(2\)](#), [unlink\(2\)](#)

## nsr\_client(5)

### Name

**nsr\_client** – Legato Storage Manager resource type “NSR client”

### Synopsis

**type: NSR client**

### Description

Each NSR client is described by a single resource of type **NSR client** (see **nsr\_resource(5)**). To edit the NSR client resources for a Legato Storage Manager server type:

```
nsradmin -c "type:NSR client"
```

See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program. The client resource may also be edited using the command **nwadmin(8)**.

For each Legato Storage Manager client, this resource describes which files should be saved, the schedule used to save these files, which directive should be used to omit files from the save, how long the files’ index entries should be kept in the on-line file index and the media index, and who is allowed to back up, browse, and recover this client’s files. A client may have more than one resource describing it.

### Attributes

The following attributes are defined for resource type **NSR client**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or by selecting the details Menu Item in the View Menu for a particular window in **nwadmin(8)**. **Dynamic** attributes have values which change rapidly. **Encrypted** attributes contain data that is not displayed in its original form. The assumption is that the data is sensitive in nature and must be protected from accidental disclosure. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read-only, single string)

This attribute specifies the hostname of this Legato Storage Manager client.

*Example:* name: venus;

**server** (constant, single string)

This attribute specifies the hostname of this client's Legato Storage Manager server. The server's hostname will be used as the default value.

*Example:* server: jupiter;

**archive services** (read/write, choice)

This attribute determines if this system can use archive services. This attribute can only be set if archive support has been enabled on the server. The choices are enabled or disabled.

*Example:* archive services: enabled;

**schedule** (read/write, choice)

This attribute specifies the name of the schedule controlling the backup levels for the save sets listed in the 'save set' attribute. The default value is 'Default'. Any currently defined schedule names may be used, see **nsr\_schedule(5)**.

*Example:* schedule: Default;

**browse policy** (read/write, choice)

This attribute specifies the name of the policy controlling how long entries will remain in this client's on-line file index. The default value is 'Month'. Any currently defined policy name may be used as long as the period defined by the policy is not longer than the retention policy's period, see **nsr\_policy(5)**.

*Example:* browse policy: Month;

**retention policy** (read/write, choice)

This attribute specifies the name of the policy controlling how long entries will remain in the media index before they are marked as recyclable. The default value is 'Year'. Any currently defined policy name may be used as long as the period defined by the policy is not shorter than the browse policy's period, see **nsr\_policy(5)**.

*Example:* retention policy: Year;

**directive** (read/write, choice)

This attribute specifies the directive to use when backing up the client. The default value is NULL. The valid choices for the directive resource are names of the currently defined 'NSR directive' resources, see **nsr\_directive(5)**.

*Example:* directive: Unix with compression directives;

**group** (read/write, choice list)

This attribute specifies the group this client is a member of. The group controls the start time for automatic backups. The value may be one of currently defined 'NSR

group' resources, see **nsr\_group**(5). The default value is 'Default'.

*Example:* group: Default;

**save set** (read/write, list)

The *save set* attribute lists the path names to be saved for this client. The names should be separated by comma space (, ). The default value is 'All'. On UNIX clients, 'All' refers to the mounted file systems. On DOS clients, 'All' refers to file systems that have been specified on the client through the 'Change Automatic Backup' selection of the *Legato Storage Manager for DOS*. By default, all of a DOS client's hard disks are backed up.

When a client must have different file systems saved on different schedules, a client resource is needed for each set of file systems on a particular schedule. For all the client resources with the same name in a group, a given path name may only appear once. When a client resource lists the save set 'All', it must be the only client resource with its name belonging to its group.

*Example:* save set: /, /usr, /usr/src;

**priority** (hidden, read/write, choice)

This attribute controls the backup priority of this client. Priority 1 is the highest, 1000 is the lowest. Automated savegrp's will attempt to back up clients with higher priorities before clients with lower priorities. Note that this is a hint only. The **savegrp** command has many parameters to consider, and may choose a lower priority client while trying to balance the load.

*Example:* priority: 500;

**remote access** (read/write, string list)

This attribute controls who may back up, browse, and recover a client's files. By default this attribute is an empty list, signifying that only users on the client are allowed to back up, browse, and recover its files. Additional users, hosts, and netgroups may be granted permission to access this client's files by adding their names to this attribute. Netgroup names must be preceded by an ampersand ('&'). Input of the form <user>@<host> or <host>/<user>, grants access to the client's files to the specified users. The <user> and/or <host> may be a wild card, "\*". If a username is a wild card, it means all users at the host are granted access to the client's data. When a host name is a wild card, that user on all hosts is granted access to the client's data. All users on a host may also be granted access to the client's data by just listing the host's name, <host> is equivalent to \*@<host> or <host>/\*. Note that this attribute does not override file system permissions, the user still needs the necessary file system permissions to back up, browse, or recover a file. The following example grants access to the client's data for all users that satisfy at least one of the following criteria, <username, user's hostname, server's domain> is a member of the netgroup "netadmins", the user is from the host mars, the user is from the host jupiter, the user's name is sam from host pluto, or the

user's id is root from any host.

*Example:* remote access: &netadmins, mars, \*@jupiter, sam@pluto, \*/root;

**remote user** (read/write, string)

This attribute has several uses. For those clients that are accessed through the **rsh**(1) protocol (new clients use **nsrexecd**(8) instead), this attribute specifies the user login name the Legato Storage Manager server will use to authenticate itself with the client. The default value is NULL, implying that 'root' should be used. When **savegrp -p** (see **savegrp**(8)) is run on the Legato Storage Manager server, the server runs commands on the client to determine which files to save. Note that when the **nsrexecd**(8) protocol is used to access the client, the remote user attribute is not used for authentication.

Certain clients, such as NetWare file servers, use this attribute along with the following **password** attribute, to gain access to the files being backed up. Other clients that back up application data, such as Sybase databases, use this attribute along with the password to gain access to the application data. There may be a different value of this attribute for each resource that describes the same client.

*Example:* remote user: operator;

**password** (read/write, encrypted)

The **savegrp** command uses this attribute when initiating the commands **savefs** and **save** on the client's computer. The commands **savefs** and **save** use the password to gain access to the files being backed up. If a password is given, then the "remote user" attribute for the client resource must also be defined. There may be a different value of this attribute for each resource that describes the same client. This attribute does not need to be set for existing UNIX clients that are not backing up any application specific data.

**backup command** (read/write, string)

The remote command to run to backup data for this client and save sets. This command can be used to perform pre and post backup processing and defaults to the **save** command. The value must not include a path and must start with the prefix "save" or "nsr".

*Example:* backup command: savemsg;

**executable path** (read/write, string, hidden)

This attribute specifies the path to use when the Legato Storage Manager server is executing commands on the client. When no path is specified, the "remote user's" \$PATH is used.

*Example:* executable path: /etc/nsr;

**server network interface** (read/write, string, hidden)

The name of the network interface on the server to be used for saves and recovers.

*Example:* server network interface: mars-2;

**aliases** (read/write, string list, hidden)

This is a list of aliases (nicknames) for the client computer that queries can match. If this list is empty, match on client name alone.

*Example:* aliases: mars;

**owner notification** (read/write, hidden)

A notification action to be executed to send the contents of status messages to the owner/primary user of a computer (for example, savegroup completion messages).

*Example:* owner notification: /usr/ucb/mail -s "mars' owner notification" carl@mars;

**statistics** (constant, hidden, dynamic)

This attribute contains three values: the size of the client's on-line file index in kilobytes, the number of kilobytes actually used, and the number of entries in the index.

*Example:*

statistics: elapsed = 1761860, index size (KB) = 776, amount used (KB) = 680, entries = 2216;

**index save set** (update-only, hidden, dynamic)

This attribute specifies the client file index save set to purge when the index operation is set to purging oldest cycle.

*Example:* index save set: /;

**index path** (read/write, hidden)

This attribute is used to allow the Legato Storage Manager administrator to balance Legato Storage Manager online file index disk utilization across multiple disk partitions. If set, this attribute contains the full path to the directory containing the client's online file index. Note that the last component of the path must match the *name* attribute of the client resource (see the previous description). If left blank, the index path defaults to the path /nsr/index/*name*, where *name* is the name attribute from the client resource.

*Example:* index path: /disk2/index/venus;

**index message** (update-only, hidden, dynamic)

This attribute contains the ending status message for the previous index operation. This attribute is typically blank, indicating that the previous operation completed successfully.

*Example:* index message::

**index operation start** (update-only, hidden, dynamic)

This attribute contains the starting time of the current index operation. This attribute is a null string ("") when the operation is 'Idle'. The format is weekday

followed by hour and minutes.

*Example:* index operation start: Wednesday 02:45;

**index progress** (update-only, hidden, dynamic)

This attribute contains the progress the index has made toward finishing the current task. This attribute is blank when the operation is 'Idle'. The progress is expressed as a percentage.

*Example:* index progress: 45;

**index operation** (update-only, hidden, dynamic)

This attribute contains the current index operation. It is normally 'Idle'.

*Example:* index operation: Reclaiming space;

**parallelism** (read/write, hidden)

This attribute specifies the maximum number of saves that should be run at the same time for the client.

*Example:* parallelism: 2;

**archive users** (read/write, string list)

This attribute specifies a list of users that are allowed to use the archive services on the client. This attribute can only be set if archive support has been enabled on the server. If no users are listed, only administrators and the local root user are allowed to use the archive services on the client. A value of '\*' implies any user is allowed to archive or retrieve data. The '/' and '@' characters are not allowed as part of the username.

*Example:* archive users: paul;

**application information** (read/write, hidden, string list)

This attribute contains client application information.

*Example:* application information: ;

**storage nodes** (read/write, string list)

This is an ordered list of storage nodes for the client to use when saving its data. Its saves are directed to the first storage node that has an enabled device and a functional media daemon, **nsrmmmd**(8). The default value of 'nsrserverhost' represents the server. See **nsr\_storage\_node**(5) for additional detail on storage nodes.

**clone storage nodes** (read/write, string list)

This is an ordered list of storage nodes for the storage node to use when its data is being cloned. Cloned data originating from the storage node will be directed to the first storage node that has an enabled device and a functional media daemon, **nsrmmmd**(8). There is no default value. If this attribute has no value, the server's 'clone "storage" "nodes"' will be consulted. If this attribute also has no value, then

the server's 'storage "nodes"' attribute will be used to select a target node for the clone. See **nsr\_storage\_node(5)** for additional detail on storage nodes.

## Examples

**Note:** The hidden attributes are not shown in these examples.

A resource to define a client, called venus, backing up all of its files to the Legato Storage Manager server mars:

```
type:                NSR client;
name:                venus;
server:              mars;
archive services:    Disabled;
schedule:            Full Every Friday;
browse policy:       Month;
retention policy:    Quarter;
directive:           Unix with compression directives;
group:               Default;
save set:            All;
remote access:       ;
remote user:         ;
password:            ;
backup command:      ;
aliases:             venus, venus.legato.com;
archive users:       ;
storage nodes:       nsrserverhost;
clone storage nodes: ;
```

The resources for a client backing up different file systems on different schedules:

```
type:                NSR client;
name:                saturn;
server:              mars;
archive services:    Disabled;
schedule:            Default;
browse policy:       Month;
retention policy:    Quarter;
directive:           ;
group:               engineering;
save set:            /, /usr, /usr/src;
remote access:       venus, sam@*, jupiter/john;
remote user:         operator;
password:            ;
backup command:      ;
aliases:             saturn.legato.com;
```

```
archive users:      ;
storage nodes:     nsrserverhost;
clone storage nodes: ;
type:              NSR client;
name:              saturn;
server:            mars;
archive services:  Disabled;
schedule:          Full on 1st Friday of Month;
browse policy:     Month;
retention policy:  Quarter;
directive:         Unix standard directives;
group:             Default;
save set:          /usr/src/archive;
remote access:     sam@venus, &netadmins, root@*;
remote user:       operator;
password:          ;
backup command:    ;
aliases:           saturn.legato.com;
archive users:     ;
storage nodes:     nsrserverhost;
clone storage nodes: ;
```

**See Also:** [rsh\(1\)](#), [ruserok\(3\)](#), [nsr\(5\)](#), [nsr\\_schedule\(5\)](#), [nsr\\_directive\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_policy\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [save\(8\)](#), [savegrp\(8\)](#), [savefs\(8\)](#), [nsradmin\(8\)](#), [nsrexecd\(8\)](#), [nwadmin\(8\)](#)

## nsr\_crash(8)

### Name

**nsr\_crash** – How to recover from a disaster with Legato Storage Manager

### Description

Legato Storage Manager can be used to recover from all types of system and hardware failure that result in loss of files.

When a Legato Storage Manager client has lost files, the **recover** command can be used to browse, select, and recover individual files, selected directories, or whole file systems. If the Legato Storage Manager **recover** command is lost or damaged it will have to be copied either from a Legato Storage Manager client or from the Legato Storage Manager distribution media.

When recovering a large number of files onto a file system that was only partially damaged, you may not want to overwrite existing versions of files. To do this, wait until **recover** asks for user input to decide how to handle recovering an existing file. You can then answer **N** meaning “always no” to cause recover to avoid overwriting any existing files, or **n** if you want to protect this file but you want **recover** to ask again on other files.

If you do want to replace the existing version of a file or set of files with the saved versions use the **add** command in **recover** to select which files should be retrieved, and answer **Y** or **y** when it asks if it should overwrite existing files (**Y** means “always yes” for future overwrite cases; **y** means just overwrite this one file).

For more information on using the **recover** command see the **recover(8)** manual page.

If the Legato Storage Manager server daemons or commands are lost, it may be necessary to re-install the server from the Legato Storage Manager distribution media. Once the Legato Storage Manager server is installed and the daemons are running, other Legato Storage Manager server files can be recovered using the **recover** command. When re-installing Legato Storage Manager you must be sure to install the `/nsr` directory in exactly the same place as it was originally installed. The computer used to recover files may be different that the one used to save the files, but it must have the same hostname as the original computer. This is important because recovery of the Legato Storage Manager on-line index requires that the index files have the same path name, which includes the server’s hostname, as they did at the time of the latest save.

In the event that the Legato Storage Manager server's index is lost, it will be necessary to first recover the index from media before the **recover** command can be used to browse and recover other files. To recover the Legato Storage Manager server's index use the **mmrecov** command. The **mmrecov** command quickly recovers the lost on-line index for a Legato Storage Manager server by locating the **bootstrap** save set produced by the **savegrp(8)** command at the end of an automatic save. The bootstrap save set tells **mmrecov** which save sets to extract from which volumes to recover the whole index. The save set identifier and other information about the bootstrap save set is printed by **savegrp** at the end of each automatic save. See the **savegrp(8)** manual page for more details.

After **mmrecov** completes, the server's file index and media database will be fully recovered (unless **mmrecov** prints an error message to the contrary). However, the Legato Storage Manager server's `/nsr/res` directory, which contains the resource files describing your Legato Storage Manager installation, require additional work on the part of the administrator before they will be used. Perform the following steps to complete the **mmrecov** process (these steps will be incorporated into **mmrecov** in a future release).

1. Shut down your Legato Storage Manager server (**nsr\_shutdown -a**).
2. Change to the `/nsr` directory (**cd /nsr**).
3. Save the temporary resource directory created by **nsr\_ize** (**mv res res.save**).
4. Move the recovered resource directory into place (**mv res.R res**).
5. Restart the Legato Storage Manager server (**cd / ; nsrd**).
6. After verifying that the recovered resources are valid, remove the temporary resource directory (**rm -r /nsr/res.save**).

---

---

**Note:** The **mmrecov** command is only used to recover the Legato Storage Manager server's index. Use **recover** to recover a client's index.

---

---

Once **mmrecov** is run and the media database has been recovered, one can use the **recover** by save set feature to restore entire file systems. This method can be faster as one does not have to use a browser to mark all the files and directories in a file system, an operation which can take a long time. You should only use this method for recovery of a file system when you can locate a recent save set for that file system which was saved with **level=full**. When recovering multiple save sets which are interleaved on media, **recover** will recover all save sets concurrently instead of

making a separate pass over the media for each save set. See the **recover(8)** man page for details on running recover by save set.

If you want to recover whole file systems, an alternative to using **mmrecov** and **recover** is to use the **scanner** command to recover all of the files in a particular save set. The **scanner** command can also be used to print a table of contents for a volume to help you locate the correct save set for a file system. Similar to recover by save set, you should only use **scanner** for recovery of a file system when you can locate a recent save set for that file system which was saved with **level=full**. If using this method of recovery, use of the **-x** option is preferred to simply piping the output through **uasms** as multiple save sets interleaved on the media can be read concurrently instead of making a separate pass over the media for each save set. For example, the first command is preferred to the second:

```
scanner -s 16234 -s 16257 /dev/nrst8 -x uasms -rv  
scanner -s 16234 /dev/nrst8 | uasms -rv  
scanner -s 16257 /dev/nrst8 | uasms -rv
```

See the **scanner(8)** manual page for more details.

If the server is damaged so badly that it will not run at all, you will need to follow the manufacturer's instructions for re-installing and rebooting a multiuser system. Once you have the system up and running in multiuser mode, you can re-install Legato Storage Manager (that is, extract Legato Storage Manager from the distribution media and install it, using **nsr\_ize(8)** or **pkgadd(1M)**, depending on your system) and use **mmrecov** to rebuild the on-line index. Finally, you will want to recover files which previously existed on the computer, but which do not exist on the manufacturer's distribution media. This may include: system files which had been customized, a specially tailored kernel, new special device entries, locally developed software, and user's personal files.

**See Also:** [nsr\\_layout\(5\)](#), [nsr\(8\)](#), [recover\(8\)](#), [savegrp\(8\)](#), [mmrecov\(8\)](#), [scanner\(8\)](#)

## nsr\_data(5)

### Name

**nsr\_data** – Data formats for Legato Storage Manager Save and Recover

### Description

All data in the Legato Storage Manager system is encoded using the *eXternal Data Representation* (XDR) standard. When files are passed between client (see **save(8)** and **recover(8)**) and server (see **nsrd(8)**) and media (see **nsrmmmd(8)**), they are represented as a *savestream*, which is encoded as a linked list of *savefiles*. There are currently 2 different *savefile* formats. A magic number at the start of each file indicates the particular type of the following *savefile* thus allowing for self identifying *savestreams* containing more than one *savefile* type. Logically each *savefile* consists of some header information followed by file data. The original *savefile1* format uses a doubly wrapped set of client attributes describing the file attributes and the file data is encoded as a *bucketlist*. The newer *savefile2* format uses an alternate singularly wrapped client attributes with the file data encoded as a bucket-less succession of self describing sections each containing a type, a length, and bytes of data. The file data section of a file is terminated by an ending section with a type of 0 (NSR\_ASDF\_END).

The XDR language description of the OPERATING SYSTEM independent portion of the *savestream* data structures is shown as follows.

```
const NSR_IDLEEN = 1024;           /* length of file id */
const NSR_MAXNAMELEN = 1024;      /* max length of file system name
                                   */
const NSR_MAXCATTRSIZE = 8192;    /* max size of client specific
                                   attributes */
const NSR_MAXBUCKETDATA = 8192;   /* max size of file bucket's data
                                   (w/o slop) */
const NSR_MAXBUCKETSIZE = 9000;   /* max total size of file bucket
                                   (w/ slop) */
const NSR_MAXCLNTSIZE = 16384;    /* max size of a clntrec */
typedef opaque fileid<NSR_IDLEEN>; /* file identifier */
typedef string nsrname<NSR_MAXNAMELEN>; /* file name type */
typedef opaque clientattr<NSR_MAXCATTRSIZE>; /* client attributes */
sbtypedef opaque wraposaverec<NSR_MAXCLNTSIZE>; /* wrapped osaverec */
typedef nulong_t checksum;         /* 4 bytes for checksum */
typedef u_long sfid_t;             /* savefile id (offset) */
struct id {
```

```

        string id_str<>;                /* id string */
        id *id_next;                    /* next such structure */
};
struct asmrec {
    id *ar_info;                        /* name and args to ASM */
    nsrname *ar_path;                  /* not currently used */
    asmrec *ar_next;                   /* next such structure */
};
const NSR_MAGIC1 = 0x09265900;        /* older format using buckets &
                                       ssaverec's */

struct osaverec {
    nsrname sr_filename;               /* name of this file */
    fileid sr_fid;                     /* client specific file id */
    asmrec *sr_ar;                     /* ASM list for this file */
    u_long sr_catype;                  /* client specific attribute type */
    clientattr sr_cattrib;             /* client specific file attributes
                                       */
};

struct ssaverec {
    sfid_t sr_id;                       /* savefile id in the savestream */
    u_long sr_size;                     /* size of encoded savefile*/
    nulong_t sr_savetime;               /* savetime of this saveset*/
    wraposaverec sr_wcr;                /* a wrapped osaverec */
};
/*
 * File data for older style savestream is logically
 * expressed as a linked list of file buckets.
 */
struct bucketlist {
    bucket bl_bucket;
    bucketlist *bl_next;
};
/*
 * XDR description of the original savefile1 format.
 */
struct savefile1 {
    u_long sf_magic;                    /* magic number (must be NSR_MAGIC1) */
    u_long sf_chksumtype;                /* file checksum type */
    ssaverec sf_saverec;                 /* wrapped file attributes */
    bucketlist *sf_data;                 /* file data in buckets */
    checksum sf_checksum;                /* checksum value */
};
/*
 * New savestream defines and structures.
 */

```

```

const NSR_MAGIC2 = 0x03175800;          /* newer bucketless format */
const NSRAPP_BACKUP = 1;                /* backup application name space */
const NSRAPP_HSM = 2;                   /* HSM application name space */
const NSRAPP_ARCHIVE = 3;              /* Archive application name space */
struct saverec {
    sfid_t sr_id;                        /* savefile id in the savestream */
    u_long sr_size;                      /* size of encoded savefile*/
    nulong_t sr_savetime;                /* savetime of this saveset*/
    nulong_t sr_appid;                   /* application id */
    nsrname sr_filename;                 /* name of encoded file */
    fileid sr_fid;                       /* client specific file id */
    asmrec *sr_ar;                       /* ASM list for this file */
    u_long sr_catype;                    /* client specific attribute type */
    clientattr sr_cattr;                 /* client specific file attributes */
};
/*
 * Defines for self describing data sections.
 * The NSR_ASDF_END type defines the end of the file data.
 * The NSR_ASDF_FILE_DATA_TYPE type has the file data preceded
 * by a nulong_t that is the relative offset from the last
 * block into the file.
 */
const NSR_ASDF_END = 0x0;                /* end of ASDF data */
const NSR_ASDF_FILE_DATA_TYPE = 0x100; /* normal file data */
/*
 * Describes a section of Legato Storage Manager "file data"
 * when using ASM Structured Data Format (ASDF) sections.
 */
struct asdf_hdr {
    nulong_t typevers;                   /* type of file data */
    nulong_t length;                     /* section length */
};
/*
 * Pseudo XDR description of the newer savefile2 format.
 * The new savefile2 format uses the unwrapped saverec
 * structure and a "bucketless" file data format that is based
 * on ASDF. The data portion ends with a 0 sized section of
 * type NSR_ASDF_END.
 */
struct savefile2 {
    u_long sf_magic;                      /* magic number (must be SF_MAGIC2) */
    u_long sf_chksumtype;                 /* file checksum type */
    saverec sf_saverec;                   /* new saverec structure */
    <asdf_hdr & data>                    /* ASDF section sans buckets */
    ...

```

```
<asdf_hdr & data> /* ASDF section sans buckets */
<asdf_hdr.typevers = 0> /* final ASDF section type =
                          NSR_ASDF_END */
<asdf_hdr.length = 0> /* final ASDF section len = 0 */
checksum sf_checksum; /* checksum value */
};
```

**See Also:** [mm\\_data\(5\)](#), [nsr\(8\)](#), [nsrmmmd\(8\)](#), [nsrd\(8\)](#), [recover\(8\)](#),  
[save\(8\)](#), [xdr\(3n\)](#).  
**RFC 1014 XDR Protocol Spec.**

## nsr\_device(5)

### Name

**nsr\_device** – Legato Storage Manager resource type "NSR device"

### Synopsis

**type: NSR device**

### Description

Each storage device used by a Legato Storage Manager server is described by a single resource of type **NSR device**. See **nsr\_resource(5)** for information on Legato Storage Manager resources. To edit the **NSR device** resources run:

```
nsradmin -c "type:NSR device"
```

Be sure to include quotation marks and to insert a space between **NSR** and **device**. See **nsradmin(8)** for information on using the Legato Storage Manager administration program. The mounting and unmounting of individual volumes (tapes or disks) is performed using the **nsrmm(8)** and **nwadmin(8)** commands.

### Attributes

The following attributes are defined for resource type **NSR device**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** indicates a value that can be set as well as read. **Hidden** indicates a hidden attribute of interest only to programs or experts. These attributes can only be seen when the hidden option is turned on in **nsradmin(8)**, or if the **Details View** option is selected in the **Media Devices** window in **nwadmin(8)**. **Static** attributes change values rarely, if ever. **Dynamic** attributes have values that change rapidly. For example, an attribute marked (**read-only, static**) has a value that is set when the attribute is created and never changes.

**name** (read-only, static)

The name attribute specifies the path name of the device. Only non-rewinding tape devices are supported. For systems that support "Berkeley style" tape positioning, use the BSD tape device name. The name given to Optical disks is typically the name given to the "c" partition of the raw device.

A logical device type has been defined to facilitate interaction with external media management services. When interacting with external media management services, the device name may be determined by the media management service associated with the device where a volume is loaded. The logical device is used to define a Legato Storage Manager device resource. The number of device resources that can exist is limited by the number of volumes managed by the service that Legato Storage Manager may access simultaneously. The name given to a logical device is not related to any specific device, but is required to be a unique name for the device. For logical devices both the media type and the family are set to **logical**. The name, type, and family are determined after the media management service has loaded a volume into a device in response to a request made by Legato Storage Manager. The name, type, and family of the actual device are then stored in the attributes **logical name**, **logical type**, and **logical family**, respectively. The association between the logical device and the actual device only exists when the volume is loaded into the device and allocated for use by Legato Storage Manager.

When defining a remote device on a storage node, include the prefix "rd=hostname:", in the path name; where *hostname* is the system to which the device is directly attached (the storage node). For more information, see **nsr\_storage\_node(5)**.

*Example:* name: /dev/rmt/0hbn;

**media type** (read-only, static)

This attribute indicates the type of media a device uses. The *media type* varies depending on the Operating System/platform. Potential values, their meaning, and default capacities are:

**4mm** – 4mm digital audio tape (1 GB);

**8mm** – 8mm video tape (2 GB);

**8mm 5GB** – 8mm video tape (5 GB);

**dlt** – digital linear tape cartridge (10 GB);

**vhs** – VHS data grade video tape (14 GB);

**3480** – high-speed cartridge tape (200 MB);

**qic** – quarter inch data cartridge (150 MB);

**himt** – half inch magnetic tape (100 MB);

**tk50** – DEC TK50 cartridge tape (94 MB);

**tk70** – DEC TK70 cartridge tape (296 MB);

**optical** – optical disks, Write Once Read Many (WORM), Erasable Optical Disks (EOD), or standard UNIX files are supported;

**file** – file device type, standard UNIX file system is supported;

**logical** – used when interacting with an external media management service.

*Example:* media type: 8mm 5GB;

**enabled** (read/write)

This attribute indicates whether a device is available for use. The value for this attribute is either **yes** or **no**. If the value is set to **no**, no volumes may be mounted into the device. This value cannot be changed if a volume is mounted.

*Example:* enabled: yes;

**read only** (read/write)

This attribute indicates whether a device is reserved for read only operations, such as recover or retrieve. The value for this attribute can be either **yes** or **no**. If the value is set to **yes**, only read operations are permitted on the device. This value cannot be changed if when the volume is mounted.

*Example:* read only: yes;

**target sessions** (read/write)

This attribute indicates the target number of saves for a device. Saves are allocated to a device in sequential order, until the target number assigned to a device is reached and the next device is used. When all devices have reached their corresponding target number, additional sessions are allocated equally across all devices. This value can only be set by root or administrator.

Use higher values to multiplex many clients onto each tape. There might be value in spreading the clients over as many devices as possible. This attribute is not a maximum number for a device, but is used for load-balancing.

*Example:* target sessions: 3;

**media family** (read-only, static, hidden)

The media family attribute describes the class of storage media, as determined from the media type. The only legal values are: **tape** – tape storage device; **disk** – disk storage device; **logical** – used when interacting with an external media management service.

*Example:* media family: tape;

**message** (read-only, dynamic, hidden)

This attribute specifies the last message received from the Legato Storage Manager server regarding this device. The values for this attribute may include information on the progress or rate of the operation.

*Example:* message: "Tape full, mount volume mars.017 on /dev/nrst8";

**volume name** (read-only, dynamic, hidden)

This attribute monitors the mounting and unmounting of volumes for a device. When a volume is mounted, the value is the volume name, otherwise there is no value.

*Example:* volume name: mars.017;

**write enabled** (read/write, dynamic, hidden)

This attribute indicates whether writing to the current volume is allowed. The value for this attribute may be set to **yes** or **no**. This value can only be set when a volume is not mounted.

*Example:* write enabled: no;

**volume operation** (read/write, dynamic, hidden)

The *volume operation* attribute manipulates the media (volume) currently located inside the device. This attribute can be set to one of the following values: **Unmount**, **Mount**, **Verify label**, **Verify write time**, **Label**, **Label without mount**, **Eject**, or **Monitor device**. Each of these operations may require parameters to be set.

When the value is **Unmount**, Legato Storage Manager releases the device. The **Unmount** operation is asynchronous.

When the value is **Mount**, Legato Storage Manager mounts the loaded volume into the device. The **Mount** operation is asynchronous.

When the value is **Verify label**, the volume's label is read by Legato Storage Manager, and the attributes **volume label** and **volume expiration** are set. The **Verify label** operation is synchronous, and therefore the operation may take a long time to complete.

When the value is **Verify write time**, the volume's label is read by Legato Storage Manager, and the attributes **volume label**, **volume expiration**, and **volume write time** are set. The **Verify write time** operation is synchronous, and therefore the operation may take a long time to complete.

When the value is **Label** or **Label without mount**, the volume receives a new label as determined by the attributes as follows. When the value is **Label**, the volume is then mounted. These operations are asynchronous.

When the value is **Eject**, Legato Storage Manager ejects the volume from the device. The **Eject** operation is asynchronous.

When the value is **Monitor device** and the device is idle (no volume loaded into the device), Legato Storage Manager will periodically check the device to determine whether a volume has been loaded into the device. When a volume containing a readable Legato Storage Manager label is loaded, the volume is placed into the Legato Storage Manager media database. The volume can then be written to by Legato Storage Manager if the volume is mounted with write permissions turned on; otherwise, the volume is mounted as read only, and cannot be written to by Legato Storage Manager. When a volume without a readable Legato Storage Manager label is loaded into the device, the device's **unlabeled volume loaded** attribute is set to **yes**, and the volume may be labeled at a later date. The **Monitor**

**device** operation is never performed on jukebox devices, because Legato Storage Manager only monitors non-jukebox devices.

**volume label** (read/write, dynamic, hidden)

This attribute is set by the **Verify label** operation and can be performed before the **Label** operation. If this attribute is blank during the labeling process, then the volume's current label is reused.

**volume default capacity** (read/write, static, hidden)

This attribute is used by the **Label** operation when the **volume current capacity** attribute is blank. A non-blank value is used to override the default capacity associated with the media type. The value of this attribute must end with **K**, **M**, or **G**, where **K** represents Kilobytes, **M** represents Megabytes, and **G** represents Gigabytes.

This hidden attribute can be modified by a user, and can be used to override default sizes when using devices (and/or tapes) with different capacities than the defaults.

*Example:* To override the default capacity of a tape drive to 10 Gb for all future volume label operations, set the value as follows:

volume default capacity: 10G;

**volume current capacity** (read/write, dynamic, hidden)

If the attribute's value is non-blank, it determines the capacity of a volume during the **Label** operation. Its format is the same as **volume default capacity**.

*Example:* volume current capacity: 5G;

**volume expiration** (read/write, dynamic, hidden)

This attribute is set by the **Verify label** operation and can also be used by the **Label** operation. The value for this attribute is specified in **nsr\_getdate(3)** format. A blank value causes the default expiration to be used during labeling.

*Example:* volume expiration: next year;

**volume pool** (read/write, hidden)

This attribute indicates the pool that a mounted volume belongs to. If this attribute is set during a **Label** or **Label without mount** operation, this value will indicate the pool a volume is being assigned to. See **nsr\_pool(5)** for more information on volume pools.

*Example:* volume pool: Default;

**NSR operation** (read-only, dynamic, hidden)

This attribute indicates the current operation being performed by a device. The valid values for this attribute are: **Idle**, **Write**, **Read**, **Eject**, **Verify label**, or **Label**.

*Example:* NSR operation: Write;

**minor mode** (read-only, dynamic, hidden)

This attribute indicates the current state of a device. The **NSR operation** attribute is the major mode. The valid values for this attribute are: **idle, reading, writing, rewinding, moving forward, moving backward, error, done, writing eof, or finding eom.**

*Example:* minor mode: moving forward;

**statistics** (read-only, dynamic, hidden)

This attribute reports the statistics for the operation of this device. The statistics include:

the time of operation ("**elapsed**"), the number of errors ("**errors**"), the last writing rate ("**last rate**"), the max number of concurrent clients ("**max clients**"), the number of file marks written ("**file marks**"), the number of rewinds ("**rewinds**"), the number of files skipped ("**files skipped**"), the number of records skipped ("**records skipped**"), the current file number ("**current file**"), the current record number ("**current record**"), the relative number of files being spaced over ("**seek files**"), the relative number of records being spaced over ("**seek records**"), the total estimated amount read/written on the volume, in KB ("**estimated KB**", to be implemented in a future release), the total amount read/written on the volume, in KB ("**amount KB**"), the current amount read/written on this file, in KB ("**file amount KB**"), and the current number of sessions assigned to this device ("**sessions**").

**cleaning required** (read/write)

This attribute indicates whether a device must be cleaned. The value for this attribute may be either **yes** or **no**. If the value of this attribute changes from yes to no and the value of **date last cleaned** attribute is not updated, then the **date last cleaned** attribute is set to the current time. Legato Storage Manager might set this attribute to yes if at the time the device is next scheduled to be cleaned it is not available to be cleaned. In this case, the following message is displayed: **device cleaning required**. This message indicates that the device must be cleaned. This attribute can only be used for a device whose **media family** is **tape** and **jukebox device** is **yes**. For all other devices the value of this attribute is always **no**.

**cleaning interval** (read/write)

This attribute indicates the amount of time from the **date last cleaned** until the next scheduled cleaning for the device. This value can be specified in **days, weeks, or months**. One day, week, or month is implied if a number is not specified. If this attribute is set and **date last cleaned** is blank, **date last cleaned** is set to the current time. This attribute may only be used for a device whose **media family** is **tape** and **jukebox device** is **yes**.

*Example:* cleaning interval: 2 weeks;

**date last cleaned** (read/write)

This attribute indicates the time and day a device was last cleaned. Input may be in any format acceptable to **nsr\_getdate(3)**. Some values acceptable to **nsr\_getdate(3)** are relative, for example, **now**. For that reason all input is converted into **ctime(3)** format, weekday, month, day, time, year. As noted in the description of **cleaning required** and **cleaning interval**, the value of this attribute might be set automatically by Legato Storage Manager. This attribute can only be used for a device whose **media family** is **tape** and **jukebox device** is **yes**.

**volume block size** (read-only, dynamic, hidden)

This attribute indicates the block size of the currently mounted volume.

**volume id** (read-only, dynamic, hidden)

This attribute indicates the volume id for the currently mounted volume.

**access count** (read-only, dynamic, hidden)

This attribute indicates the total number of operations performed on the device since it was configured as a Legato Storage Manager device.

**access weight** (read-only, dynamic, hidden)

This attribute indicates the weight of a single operation performed on the device. The "access count" attribute will be incremented by "access weight" each time an operation performed on the device. The higher the weight, the less often the device will be selected for new operations.

**consecutive errors** (read-only, dynamic, hidden)

This attribute indicates the current number of consecutive errors on a device.

**max consecutive errors** (read-only, hidden)

This attribute indicates the maximum number of consecutive errors allowed before disabling the device.

**operation arg** (read-only, dynamic, hidden)

This attribute indicates extra parameters to be used during device operations. Parameters are packed into a string and parsed by the associated operation's function.

**volume message** (read-only, dynamic, hidden)

This attribute indicates the result of the last volume operation.

**volume write time** (read-only, dynamic, hidden)

This attribute indicates the time that a save set was first written to the volume.

**volume flags** (read/write, hidden)

This attribute displays the new flags for the volume being operated on. This attribute is used during "Label" or "Label without mount" operations.

**jukebox device** (read/write, dynamic, hidden)

This attribute indicates the media device that is part of a jukebox device. This value can be either **yes** or **no**.

**unlabeled volume loaded** (read only, dynamic, hidden)

This attribute indicates whether a volume loaded into the device has a readable Legato Storage Manager volume label. This value can be either **yes** or **no**. This attribute is set to **yes** when Legato Storage Manager is monitoring the device, a volume is loaded into the device, and the volume does not have a valid Legato Storage Manager label that can be read by this device. This attribute is set to **no** when the volume in the device is labeled or ejected from the device.

**auto media management** (read/write)

This attribute indicates whether "automated media management" is enabled for a device. For jukebox devices this value is always **no**. For non-jukebox devices, this value can be either **yes** or **no**. If this value is set to **yes**, then any recyclable volumes loaded into the device might be automatically re-labeled by Legato Storage Manager for reuse, and unlabeled volumes loaded into the device can be automatically labeled. When Legato Storage Manager is labeling a volume that is not expected to have a valid Legato Storage Manager label, it verifies that the volume is unlabeled before labeling the volume. A volume is considered to be unlabeled if the volume does not contain a label that may be read by this device.

**Note:** If a volume contains a label, but the label is written at a density that cannot be read by the associated device, the volume is considered to be unlabeled. If the volume contains data written by an application other than Legato Storage Manager, it most likely does not have a label recognizable by Legato Storage Manager, and the volume is considered to be unlabeled. With this attribute enabled, care should be taken when loading any volume considered to be unlabeled or recyclable into the device. The volume might be re-labeled and the data previously on the volume over-written by Legato Storage Manager.

When this attribute is set to **yes** for a device, and the device is idle (no tape loaded into the device), Legato Storage Manager will monitor the device and wait for a volume to be loaded. See the description of **Monitor device** in the discussion of the **volume operation** attribute.

*Example:* auto media management: yes;

**logical name** (read-only, hidden, no create)

This attribute indicates the name of the actual device associated with the logical device. This attribute is only used for logical devices.

*Example:* logical name: /dev/rmt/0hbn;

**logical type** (read-only, hidden, no create)

This attribute indicates the actual device type associated with the logical device. The values that can be associated with this attribute are the values that are valid for the attribute **media type**. The only exception is that the value of this attribute cannot be set to **logical**. This attribute is only used for logical devices.

*Example:* logical type: 8mm 5GB;

**logical family** (read-only, hidden, no create)

This attribute indicates the family of the actual device currently associated with the logical device. The values that can be associated with this attribute are the values that are valid for the attribute **media family**. The only exception is that the value of this attribute cannot be set to **logical**. This attribute is only used for logical devices.

*Example:* logical family: tape;

**connection process id** (read only, hidden, no create)

This attribute indicates the process identifier maintaining the connection with an external media management service.

External media management services often require a connection to be maintained while an application is using allocated resources. If the connection is not maintained the service may attempt to reclaim any resources allocated to an application. This may include unloading a volume currently mounted into a device. Therefore, while Legato Storage Manager has a volume mounted into a device being managed by such a service, a process must maintain an open connection with the media management service.

**connection message** (read only, hidden, no create)

This attribute records any error message(s) reported upon exit by a process maintaining a connection with an external media management service.

**connection status** (read only, hidden, no create)

This attribute records the exit status reported by a process maintaining a connection with an external media management service. A status of zero indicates that the process exited successfully. A nonzero status indicates an error occurred while the process was exiting.

**save mount timeout** (read/write, hidden, no create)

This attribute indicates the timeout value for an initial save mount request for the storage node on which a device is located. If the request is not satisfied within the indicated time, the storage node will be locked from receiving save processes for the "save lockout" time. See **nsr\_storage\_node(5)** for a description of storage nodes. This attribute can be used for local devices as well, but "save lockout" cannot be changed from its default value of zero. Hence, local devices cannot be locked out from save requests.

**save lockout** (read/write, hidden, no create)

This attribute indicates the number of minutes a storage node will be locked from receiving save assignments after it reaches the **save mount timeout** time during a save mount request. A value of zero indicates that the node will not be locked. This attribute cannot be changed for local devices.

## Example

A complete example follows:

```
type:                NSR device;
name:                /dev/nrst8;
message:             writing, done
volume name:         mars.017;
media family:        tape;
media type:          8mm 5GB;
enabled:             Yes;
write enabled:       Yes;
read only:           No;
target sessions:    4;
volume label:        mars.017;
volume default capacity: ;
volume current capacity: 5000 MB;
volume expiration:   "Thu Sep 21 17:23:37 1999";
volume pool:         Default;
volume flags:        ;
volume operation:    ;
volume write time:   ;
volume block size:   32 KB;
volume id:           32449;
accesses:            199;
access weight:       1;
consecutive errors: 0;
max consecutive errors: 20;
operation arg:       ;
volume message:      ;
NSR operation:       ;
minor mode:          idle;
jukebox device:      Yes;
statistics:          elapsed = 257572, errors = 0,
                    last rate = 397, max clients = 3,
                    file marks = 22, rewinds = 4,
                    files skipped = 1976, records
                    skipped = 0, current file = 2389,
                    current record = 162, seek
```

```
files = 0, seek records = 0,  
estimated kb = 0, amount kb = 6273,  
file amount kb = 6273, sessions= 1;  
cleaning required:      No;  
cleaning interval:     2 weeks;  
date last cleaned:     "Tue Apr 11 15:10:32 1999";  
auto media management: No;  
unlabeled volume loaded: No;  
logical name:          ;  
logical type:          ;  
logical family:       ;  
connection process id: ;  
connection message:   ;  
connection status:    ;  
save mount timeout:   30;  
save lockout:         0;
```

## Files

*/nsr/res/nsr.res* – this file should never be edited directly. Use **nsrmm(8)**, **nsradmin(8)**, or **nwadmin(8)** instead.

**See Also:** [nsr\\_getdate\(3\)](#), [ctime\(3\)](#), [nsr\\_resource\(5\)](#), [nsr\\_pool\(5\)](#), [nsr\\_schedule\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [nsr\(8\)](#), [nsrmm\(8\)](#), [nsrmm\(8\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#)

## nsr\_directive(5)

### Name

**nsr\_directive** – Legato Storage Manager resource type “NSR directive”

### Synopsis

**type: NSR directive**

### Description

Each NSR directive is described by a single resource of type **NSR directive** (see **nsr\_resource(5)**). To edit the NSR directive resources for a Legato Storage Manager server use **nsradmin(8)** or **nwadmin(8)**. See the corresponding manual page for more information on the use of these Legato Storage Manager administration programs.

These resources are used by the Legato Storage Manager *asm* family of commands when processing files, see **uasmm(8)** and **nsr(5)**. Directives can be used to improve the efficiency of backups by controlling which files get saved and specifying special handling on certain types of files.

### Attributes

The following attributes are defined for resource type **NSR directive**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed after the resource has been created. **Read/write** means the value can be updated by authorized administrators. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or the details view is enabled in **nwadmin(8)**. **Dynamic** attributes have values which change rapidly. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only)

The names of directive resources are displayed as choices when creating or updating Legato Storage Manager client resources, see **nsr\_client(5)**. The name can generally be chosen at the administrator’s convenience, but it must be unique for this Legato Storage Manager server. The directive resource named ‘Unix standard

directives' may be modified, but it may not be deleted. Other directives can only be deleted if no clients or archive lists are using them.

*Example:* name: Unix standard directives;

**directive** (read/write)

This attribute contains the rules defining the directive. The value of this attribute is similar to the contents of a file except that absolute path names must be specified for each << path >> directive. See **nsr(5)** for more information on the format of Legato Storage Manager directives.

*Example:* directive: "<< / >> skip : core";

**NOTE:**

Legato Storage Manager comes with four directive resources already defined, "Unix standard directives", "Unix with compression directives", "DOS standard directives", and "NetWare standard directives". The first two are meant for use with clients running on UNIX platforms. "DOS standard directives" is intended for use with clients on computers running DOS. The last directive, "NetWare standard directives", is meant for use with clients running on NetWare platforms. There may also be two other directives "Default" and "Default with compression". These are old names for "Unix standard directives" and "Unix with compression directives", respectively. Legato Storage Manager will remove the directive resources using the old names when they are no longer of use.

## Example

An example NSR directive resource, named 'Unix directive', follows:

```
type:      NSR directive;
name:      Unix directive;
directive: "
<< / >>
    +skip : core
    skip : tmp
<< /usr/spool/mail >>
    mailasm : *
<< /nsr >>
    allow
";
```

**See Also:** [nsr\(5\)](#), [nsr\\_resource\(5\)](#), [savegrp\(8\)](#), [savefs\(8\)](#), [uasm\(8\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#)

## nsr\_getdate(3)

### Name

**nsr\_getdate** – Convert time and date from ASCII

### Synopsis

```
#include <sys/types.h>
```

```
time_t nsr_getdate(buf)  
char *buf;
```

### Description

The **nsr\_getdate(0)** routine converts most common time specifications to standard UNIX format. It takes a character string containing time and date as an argument and converts it to a time format.

The character string consists of zero or more specifications of the following form:

*tod*

A *tod* is a time of day, which is of the form *hh[:mm[:ss]]* (or *hhmm*) [*meridian*] [*zone*]. If no meridian – *am* or *pm* – is specified, a 24-hour clock is used. A *tod* may be specified as just *hh* followed by a *meridian*. If no zone (for example, GMT) is specified, the current time zone, as determined by the second parameter, *now*, is assumed.

*date*

A *date* is a specific month and day, and possibly a year. The acceptable formats are *mm/dd[/yy]* and *monthname dd[, yy]*. If omitted, the year defaults to the current year. If a year is specified as a number in the range 70 and 99, 1900 is added. If a year is in the range 00 and 30, 2000 is added. The treatment of other years less than 100 is undefined. If a number not followed by a day or relative time unit occurs, it will be interpreted as a year if a *tod*, *monthname*, and *dd* have already been specified; otherwise, it will be treated as a *tod*. This rule allows the information from **date(1)** or **ctime(3)** to be passed as input to *nsr\_getdate*.

*day*

A *day* of the week may be specified; the current day will be used if appropriate. A *day* may be preceded by a *number*, indicating which instance of that day is desired; the default is 1. Negative *numbers* indicate times past. Some symbolic *numbers* are accepted: **last**, **next**, and the ordinals **first** through **twelfth** (**second** is ambiguous,

and is not accepted as an ordinal number). The symbolic number **next** is equivalent to **2**; thus, **next monday** refers not to the immediately coming Monday, but to the one a week later.

relative time

Specifications relative to the current time are also accepted. The format is *[number]unit*; acceptable units are **year**, **month**, **fortnight**, **week**, **day**, **hour**, **minute**, and **second**.

The actual date is formed as follows: first, any absolute date and/or time is processed and converted. Using that time as the base, day-of-week specifications are added; last, relative specifications are used. If a date or day is specified, and no absolute or relative time is given, midnight is used. Finally, a correction is applied so that the correct hour of the day is produced after allowing for daylight savings time differences.

*Nsr\_getdate* accepts most common abbreviations for days, months, and so on.; in particular, it will recognize them with uppercase or lowercase first letter, and will recognize three-letter abbreviations for any of them, with or without a trailing period. Units, such as **weeks**, may be specified in the singular or plural. Time zone and meridian values may be in uppercase or lowercase, and with or without periods.

**See Also:** `ctime(3)`, `date(1)`, `ftime(3c)`, `localtime(2)`, `time(2)`

## Bugs

The grammar and scanner are rather primitive; certain desirable and unambiguous constructions are not accepted. Worse yet, the meaning of some legal phrases is not what is expected; **next week** is identical to **2 weeks**.

The daylight savings time correction is not perfect, and can get confused if handed times between midnight and 2:00 am on the days that the reckoning changes.

Because `localtime(2)` accepts an old-style time format without zone information, passing *nsr\_getdate* a current time containing a different zone will probably fail.

## nsr\_group(5)

### Name

**nsr\_group** – Legato Storage Manager resource type “NSR group”

### Synopsis

**type: NSR group**

### Description

Each Legato Storage Manager group is described by a single resource of type **NSR group** (see **nsr\_resource(5)**). To edit the **NSR group** resources for a Legato Storage Manager server type:

```
nsradmin -c "type:NSR group"
```

or use the **nwadmin(8)** GUI. See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program.

These resources control when a group of Legato Storage Manager clients begin saving data and whether backups are started automatically each day. Each **NSR client** resource (see **nsr\_client(5)**) lists the groups of which that client (or save sets for that client) is a member. Groups can only be deleted if no clients are members of them.

### Attributes

The following attributes are defined for resource type **NSR group**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed by an administrator once the resource is created. **Read/write** means the value can be set as well as read at any time. **Choice** indicates that the value can only be selected from a given list. **Yes/no** means only a yes or no choice is possible. **Static** attributes change values rarely, if ever. **Dynamic** attributes have values which change rapidly. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)**. For example, an attribute marked **(create-only, static)** has a value which is set when the attribute is created and never changes. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only)

This attribute contains the name of the group defined by this resource. The name must be unique for this Legato Storage Manager server, but otherwise can be anything that makes sense to the administrator. This name will appear as a choice attribute of each **NSR client** and **NSR pool(5)** resource. The **NSR group** resource named 'Default' may be modified, but it may not be removed. The name can only be specified when the group is created.

*Example:* name: marketing;

**autostart** (read/write, choice)

The autostart attribute determines if this group will be saved automatically every day. It may be one of three values: **Enabled**, **Disabled** or **Start now**. When the value is **Enabled**, the members of this group will start saving data at the time specified in the **start time** attribute. When the value is **Disabled**, the member of this group will not automatically start saving their data. When the **Start now** value is specified, the member clients will start saving their data immediately. The attribute will then return to its prior value.

*Example:* autostart: Enabled;

**autorestart** (read/write, choice, hidden)

This controls whether this group should be automatically restarted when an incomplete run (due to a power failure or administrator intervention) is noticed during Legato Storage Manager server startup. Like the **autostart** attribute, setting this attribute's value to **Restart now** causes Legato Storage Manager to restart the group immediately. Enabling **autorestart** only has an effect if **autostart** is also enabled.

**stop now** (read/write, choice, hidden)

Setting this value to 'True' when this group is running causes this group to stop all of its saves immediately. Once the group is stopped, the value is set back to 'False'. These are the only valid values.

**start time** (read/write)

The start time attribute specifies the time of day when this group will start saving. The Legato Storage Manager server's local time is used. The time is specified as "*hours:minutes*". Note that the quotation marks may be necessary when using character-based administration tools such as the **nsradmin** program because of the colon in the value. The hours may range from 0 to 23 (using a 24 hour clock) and the minutes range from 0 to 59.

*Example:* start time: "4:53";

**last start** (read/write, hidden)

The last time this group was started. This attribute is for informational purposes only, and changing it has no effect on the system.

**interval** (read/write, static, hidden)

The interval time specifies how often this group is to be run automatically by Legato Storage Manager. Manually starting a group overrides the interval. The default value is 24:00, which means run once a day.

**force incremental** (read/write, static, hidden, choice)

Setting this attribute to 'Yes' will force an incremental level of a **savegrp**, when the **interval** attribute is less than 24 hours. The default value is 'Yes,' which means force an incremental if the group is run more than once a day. A value of 'No' would allow more than one full for each day.

**client retries** (read/write)

The number of times failed clients should be retried before **savegrp** gives up and declare them failed. Zero means don't retry. Abandoned saves are not retried, because they may eventually complete. A client's save sets are retried by **savegrp** whenever **savegrp** would otherwise not be able to start a new save set. That is, **savegrp** prefers to start new save sets first, and only retries when there is nothing else to do.

*Example:* client retries: 1;

**clones** (read/write, static, yes/no, choice)

Setting this value to 'Yes' causes saves of this group to automatically make a clone of every save set backed up. The save set clones will be sent to the pool named in the **clone pool** attribute.

**clone pool** (read/write, static, choice)

The pool to which save set clones should be sent when 'clones' is 'Yes'. Only pools of type 'Backup Clone' are allowed (see **nsr\_pool(5)**).

**options** (read/write, static, hidden)

The values specify flags with which this group will be run. The values No Monitor, No index save, No save, Verbose, Estimate, and Preview map to the **savegrp** command line flags -m, -I, -n, -v, -E, and -p respectively. Some of these values (Preview and No save) are automatically reset when a run of **savegrp** completes normally.

*Example:* options: Verbose;

**level** (read/write, hidden, choice)

This is an explicit level the **savegrp** will use when started automatically by Legato Storage Manager. *This hidden attribute can be modified by a user.* This value is not cleared automatically, that is, if one sets this attribute to full this **savegrp** will run with a full level until this value is manually cleared. When not specified (the normal case), the NSR Schedule for each client file system will be used to determine the level. Manually running **savegrp** from the command line overrides this value. The

choices are the standard level identifiers 'full', 'consolidate', 'incr', 'skip', and the number levels '1' through '9'.

**printer** (read/write, static, hidden)

The printer to which the bootstrap save set information will be printed, if one is generated by the run of this group. *This hidden attribute can be modified by a user.* If an invalid printer name is specified, bootstrap information will be included in the **savegrp** completion information piped through the *savegroup completion* notification (see **nsr\_notification(5)**).

*Example:* printer: ps;

**schedule** (read/write, choice, hidden)

The schedule to use for determining what level of save to perform. *This hidden attribute can be modified by a user.* This value is not cleared automatically, that is, if one sets this attribute to a particular schedule, all clients which are part of this group will have their schedules overridden until this value is manually cleared. This overrides the schedule specified for individual clients. See **nsr\_schedule(5)**.

**schedule time** (read/write, hidden)

An explicit time can be specified when looking at a schedule to determine which level of save to perform. A null value (normal setting) means use the current date to determine the level.

*Example:* schedule time: "3:00 am 01/11/93";

**inactivity timeout** (read/write, static, hidden)

The number of minutes that the **savegrp** command waits for any kind of activity on the server before concluding that a **savegrp** descendant is hung. *This hidden attribute can be modified by a user.* Once a suspension is detected, **savegrp** prints a message indicating that a **save** is being aborted, kills or stops the backup, and moves on to its next task. Inactivity is defined as the last time a client has sent data to the server. If a client has a very large file system and an incremental is run, it is possible for **savegrp** to stop a save set that only appears to be suspended. In these cases, the inactivity timeout should be increased to accommodate the particular client.

*Example:* inactivity timeout: 30;

**work list** (read/write, dynamic, hidden)

The list of saves still not completed. These come in sets of 3 values: the client name, the level of save, and the path to save.

*Example:* work list: mars, incr, /usr, mars, incr, /g, mars, venus, /usr

**completion** (read/write, dynamic, hidden)

The status of each save set that has been completed. These come in sets of 4 values: the client name, the path saved, a status message (succeeded, failed, or unexpectedly exited), and the information from the save.

**Example:**

completion: "mars", "/usr", "succeeded", "mars: / level=full, 6577 KB 00:06:41  
625 files"

**status** (read-only, dynamic, hidden)

The current status of this NSR group. Currently, this can have the values 'idle', 'running' and 'cloning'. The value 'idle' is set when the group is not active, it is 'running' while backups are in progress, and it is 'cloning' when the backups are complete and clones are automatically being made.

## Example

The default NSR group resource automatically starts its members at 33 minutes past 3 o'clock in the morning:

```
type:          NSR_group;
name:          Default;
autostart:     Enabled;
start time:    "3:33";
administrator: root;
```

A complete example follows, with the hidden attributes shown with values reasonable for an active group:

```
type:          NSR_group;
name:          Default;
autostart:     Enabled;
start time:    "3:33";
options:       Restartable;
printer:       lp2;
inactivity timeout: 30;
work list:     mars, incr, /g, mars, incr, index,
               venus, incr, /usr, venus, incr, index,
               jupiter, full, /, jupiter, full, /usr,
               jupiter, full, index
completion:    mars, /, succeeded,
               "mars: / level=incr, 31 KB 00:01:01
               72 files",
               mars, /usr, succeeded,
               "mars: /usr level=incr, 2 KB 00:00:48
               5 files",
               venus, /, succeeded,
               "venus: / level=incr, 7711 KB 00:04:37
               29 files";
administrator: root, &operator;
```

**See Also:** [nsr\(8\)](#), [nsr\\_notification\(5\)](#), [nsr\\_pool\(5\)](#), [nsr\\_resource\(5\)](#),  
[nsr\\_schedule\(5\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#), [savegrp\(8\)](#)

## nsr\_label(5)

### Name

**nsr\_label** – Legato Storage Manager resource type “NSR label”

### Synopsis

**type: NSR label**

### Description

Each NSR label template is described by a single resource of type **NSR label** (see **nsr\_resource(5)**). To edit the NSR label resources for a Legato Storage Manager server, type:

```
nsradmin -c "type:NSR label"
```

or use the **nwadmin(8)** GUI. See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program.

This resource describes the templates used to generate volume labels.

### Attributes

The following attributes are defined for resource type **NSR label**. The information in parentheses describes how the attribute values are accessed. Several additional attributes (for example, **administrator**) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only, single string, static)

This attribute specifies the name of this label template.

*Example:* name: Default;

**fields** (read/write, list of strings)

This attribute specifies the constituent fields of a *label template*. When generating a volume name, the current value of each field is concatenated. The first field is considered the most significant, the last field the least. If there is a *separator* (see following description) defined, then it will be placed between fields as they are concatenated to form a volume name. The fields are separated by commas.

There are four different types of fields: ‘numeric range’, ‘lower-case range’, ‘upper-case range’, and a ‘list of strings’. A ‘list of strings’ consists of space (‘ ’) separated strings. The other types are specified as starting and ending values

separated by a dash ('-'). The starting and ending values of a range must have the same number of characters.

The *next* attribute (see the following description) contains the current position or value of each field. After a volume name has been assigned to a volume, the *next* attribute is incremented. When the ending value is reached, the current value will wrap around to the starting value. A 'list of strings' field is incremented by selecting the next string in the list. A numeric range field is incremented by adding 1 to its current value. Lower-case and upper-case ranges are incremented by moving on to the next letter in the least significant position. In the following example, after *aa.99*, the next label would be *ab.00*.

*Example:* fields: aa-zz, 00-99;

**separator** (read/write, single choice, null ok)

This attribute specifies the character to use to separate the label fields. It may be one of '.', '\_', ':', '-' or NULL.

*Example:* separator: .;

**next** (read/write, single string)

This attribute specifies the next volume name to use. After it is assigned to a volume, the next volume name will be generated and remembered here. The attribute consists of a component for each of the specified fields and the separator.

*Example:*

next: aa.00;

Using the separator and field attributes shown previously, the *next* attribute would show: next:aa.01;

This would be followed by: next:aa.02;

## Examples

A label resource named *engineering* is shown as follows. (Hidden options are not shown.) There are two range type fields defined, the first ranging from 'aa' to 'zz', the second from '00' to '99'. The *separator* attribute has the value '.' and it will be inserted in between the two fields. The *next* attribute holds the next name that will be generated by this template. After *aa.00* is used, the *00* will be incremented. The new name will be *aa.01*. After 98 more names have been generated, the *next* attribute will hold the name *aa.99*. When this name is incremented, the *next* attribute will hold *ab.00*. After generating 67,500 more names, the *next* attribute will hold *zz.99*. This will be followed by *aa.00*.

```
type:      NSR label;
name:      engineering;
fields:    aa-zz, 00-99;
```

```
separator:  . ;
next:      aa.00;
```

A label resource named *accounting* is shown as follows. The field attribute defines five component fields. The *separator* attribute has the value `'.'`. It will be inserted in between adjacent fields. The *next* attribute holds the next name that will be used with this template. After *0.23.aa.AA.first* is used, the fifth field will be incremented. The new name will be *0.23.aa.AA.second*. This will be followed by *0.23.aa.AB.first*. After 1349 more volume names, the name will be *0.23.aa.ZZ.second*. This will be followed by *0.23.ab.AA.first*. After using *9.45.zz.ZZ.second*, the name will wrap around to *0.23.aa.AA.first*.

```
type:      NSR label;
name:      accounting;
fields:    0-9, 23-45, aa-zz, AA-ZZ, first second;
separator: . ;
next:      0.23.aa.AA.first;
```

**See Also:** [nwadmin\(8\)](#), [nsradmin\(8\)](#), [nsrmmm\(8\)](#), [nsr\(8\)](#)

## nsr\_layout(5)

### Name

**nsr\_layout** - Legato Storage Manager file layout

### Description

The Legato Storage Manager server file system has a directory called **/nsr** that contains log files, on-line indexes, and configuration information. This directory can be created in any file system with **/nsr** set up as a symbolic link to the actual directory (this is determined at installation time). The format of this directory is as follows:

#### **/nsr/logs**

Contains server logging messages. The files in this directory are in ASCII format.

#### **/nsr/res**

Contains the configuration files for various components of the Legato Storage Manager server. For example, the server stores configuration files in **/nsr/res/nsr.res** and **/nsr/res/nsrjb.res**.

#### **/nsr/mm**

Contains the media index. Information about the contents of this index file can be printed with the **nsrls(8)** command. See the **nsrmm(8)** and **mminfo(8)** manual pages on how to view and manipulate the media index information.

#### **/nsr/index**

This directory contains subdirectories with names that correspond to the Legato Storage Manager clients that have saved files. Each index directory contains files that allow the Legato Storage Manager server to provide an on-line database of the client's saved files. The most important element is the **db** directory which contains the Legato Storage Manager save records and access indexes to those records. The disk space utilized by the index grows with the number of files saved by the Legato Storage Manager service. Administrators should plan to use about 200 bytes for each saved file instance placed in this index. There are no practical limits on the maximum size of an online index, except that it must reside entirely within a single file system.

The format of the **db** directory is subject to change, and is accessible only through an RPC interface to **nsrindexd(8)**. However, the command **nsrls(8)** can be used to obtain some useful statistics from this directory. The **nsrck(8)** command is used for checking and rebuilding index files.

**/nsr/cores**

Contains directories that correspond to the Legato Storage Manager server daemons and certain executables. Each directory may contain core files from Legato Storage Manager server daemons or executables that have abnormally terminated.

**/nsr/drivers**

This directory may contain any device drivers for use with Legato Storage Manager.

**/nsr/tmp**

This directory contains temporary files used by the Legato Storage Manager system.

The executables for the Legato Storage Manager system are usually installed in the directories **/usr/etc** or **/usr/bin**, though alternate locations may be chosen when the **nsr\_ize(8)** installation script is run. Alternate locations are not available on SCO systems that install with **custom(ADM)**. See **pkgadd(1M)** for details on alternate executable locations for Solaris 2.x.

When executables for more than one architecture are installed, the non-native architectures are by default put in the directory **/export/exec/arch/etc**, where *arch* refers to a given architecture name. A different location to install non-native executables can be chosen at installation time.

## Files

**/nsr**

Legato Storage Manager indexes, log files, and configuration information.

**/usr/etc, /usr/bin**

Where Legato Storage Manager executables for the native architectures are normally installed.

**/export/exec/arch/etc**

Where Legato Storage Manager executables for non-native architectures are normally installed.

**/usr/bin, /usr/sbin, /usr/lib/nsr**

Where Legato Storage Manager executables for Solaris 2.x are normally installed.

**See Also:** [nsrck\(8\)](#), [nsrindexd\(8\)](#), [nsrls\(8\)](#), [nsrmm\(8\)](#), [mminfo\(8\)](#)

## nsr\_license(5)

### Name

**nsr\_license** – Legato Storage Manager resource type “NSR license”

### Synopsis

**type: NSR license**

### Description

A resource of type **NSR license** is used to describe each feature enabled in your Legato Storage Manager installation. See **nsr\_resource(5)** for more information on Legato Storage Manager resources. To inspect the NSR license resources type:

```
nsradmin -c "type:NSR license"
```

or use the **nwadmin(8)** GUI. NSR license resources may be created, enabled and authorized from the GUI, but the **nsrcap(8)** command must be used to update an existing license resource. See **nsradmin(8)** for more information on using the Legato Storage Manager administration program.

### Attributes

The following attributes are defined for resource type **NSR license**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed by an administrator, except when the resource is created. **Read/write** means the value can be changed at any time by authorized administrators. **Static** attributes change values rarely, if ever. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or by selecting the details Menu Item in the View Menu for the Server Registration window in **nwadmin(8)**. For example, an attribute marked **(create-only, static)** has a value which is set when the attribute is created and never changes. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only, static)

This attribute holds the name of the license resource.

**enabler code** (create-only, static)

This code is identical to the code entered into the **nsrcap(8)** command to enable the

feature named in this resource. The enabler code consists of 18 hexadecimal digits, printed in groups of 6 digits.

*Example:* enabler code: 123456-123456-123456;

**host id** (read-only, dynamic)

The unique host id associated with the computer or licensed operating system on which the enabler has been loaded. This value will often be an 8 digit hexadecimal number, however other formats are possible, depending on specific platform requirements.

*Example:* host id: 7260d859;

**expiration date** (read-only)

The date on which this enabler will expire, if the enabler is an evaluation enabler or an otherwise un-registered license enabler. The enabler expires at 12:00:01 am on the date listed. The special prefix **G** means that a grace period has been allowed for this enabler. Enablers with the grace period allowed should be registered immediately. If the enabler has been registered, and the **auth code** is filled in with a valid value, the value will be as shown in the following example.

*Example:* expiration date: Authorized - no expiration date;

**auth code** (read/write)

An 8 digit hexadecimal value used to permanently authorize an enabler. The unique, valid auth code for an enabler is obtained from Legato by registering each purchased license enabler. Eval enablers cannot be permanently authorized. If the server's host id changes, all auth codes will immediately be invalidated, and the enablers must be re-registered with Legato to obtain new auth codes.

*Example:* auth code: abcdef00;

**license type** (create-only, hidden)

A special code, used internally to describe the specific feature or features enabled by this license enabler.

*Example:* license type: J16;

**checksum** (read/write, hidden)

A coded checksum used to maintain consistency of a NSR license resource, and between license resources.

## Example

The following is a complete NSR license resource for an authorized base enabler:

```
type:          NSR license;
name:          Legato Storage Manager Advanced/10;
enabler code:  123456-123456-123456;
host id:       7260d859;
```

```
expiration date:   Authorized - no expiration date;  
auth code:        abcdef00;  
license type:     B10;  
checksum:         xxxxxxxxxxxxxxxxxxxxxxxxxxxx;
```

## Files

**/nsr/res/nsr.res** – this file should never be edited directly. Use **nsradmin(8)** instead.

**See Also:** [nsr\\_resource\(5\)](#), [nsr\(8\)](#), [nwadmin\(8\)](#), [nsradmin\(8\)](#),  
[nsrcap\(8\)](#)

## nsr\_migration(5)

### Name

**nsr\_migration** – Legato Storage Manager resource type “NSR migration”

### Synopsis

**type: NSR migration**

### Description

Each NSR migration client is described by a single resource of type **NSR migration** (see **nsr\_resource(5)**). To edit the NSR migration resources for a Legato Storage Manager server type:

**nsradmin -c "type:NSR migration"**

See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program. The client resource may also be edited using the command **nwadmin(8)**.

For each Legato Storage Manager migration client, this resource describes which files should be saved, the schedule used to save these files, which directive should be used to omit files from the save, the group that files will be pre-migrated with, the high-water mark for migration, the low-water mark for migration, the minimum last access time for file migration, the minimum file size for migration, a list of file owners and groups to include or exclude for migration, and a list of filename patterns to skip migration of. A client may have more than one resource describing it.

### Attributes

The following attributes are defined for resource type **NSR migration**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or by selecting the details Menu Item in the View Menu for a particular window in **nwadmin(8)**. **Dynamic** attributes have values which change rapidly. **Encrypted** attributes contain data that is not displayed in its original form. The assumption is that the data is sensitive in

nature and must be protected from accidental disclosure. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read-only, single string)

This attribute identifies the Legato Storage Manager client and save set whose migration attributes are stored in this resource.

*Example:* name: venus All;

**client** (read-only, single string)

The *client name* identifies the HSM client whose save sets are to be placed under migration control. This name must be a valid name for an existing NSR client resource.

*Example:* client: elantra;

**save set** (read/write, list)

The *save set* attribute lists the path names of file systems or sub-trees within file systems to put under migration control for this client. The names should be separated by comma space (, ). The default value is 'All'. On UNIX clients, 'All' refers to all mounted file systems, except for the file systems /, /usr, /var (and /opt on Solaris), where migration of files is not allowed.

*Example:* save set: /usr/src, /spare;

**enabled** (read/write, choice)

The *enabled* attribute determines whether the save set named in this resource should be automatically migrated. A resource can be disabled to temporarily keep migration from occurring. On update, any ongoing migration operations will complete. This attribute has no affect on recall operations.

*Example:* enabled: No;

**directive** (read/write, choice)

Directives tell the client how to migrate certain files. The choices are defined by the set of existing directives. The default value is NULL. The valid choices for the directive resource are names of the currently defined 'NSR directive' resources, see **nsr\_directive(5)**.

*Example:* directive: Unix with compression directives;

**group** (read/write, list)

The groups this client/saveset is part of for pre-staging migrated files. The choices are defined by the set of existing groups.

*Example:* group: Default;

**high water mark (%)** (read/write, single string)

The point at which files should start being replaced by stubs, measured as the percentage of available space used on the file system. Migration (stub replacement) will continue until the lower water mark is reached.

*Example:* high water mark (%): 90;

**low water mark (%)** (read/write, single string)

The point at which files should be stop being replaced by stubs, measured as the percentage of available space used on the file system.

*Example:* low water mark (%): 80;

**last access time** (read/write, single string)

Migrate only those files that have not been accessed in the past specified relative time. If this value is empty, the last access time to not be considered.

Example values: 5 days ago, 1 month ago, 1 second ago.

*Example:* last access time: 7 days ago

**minimum file size (KB)** (read/write, single string)

Migrate only those files that are larger than then specified size (in Kilobytes). Setting this value to zero causes file size to not be considered.

*Example:* minimum file size (KB): 5;

**file owner** (read/write, list)

A list of the users whose files should be migrated. A leading dash ('-') in a username indicates negation, in which case all users except the named user's files will be migrated.

*Example:* file owner: karl, cohers;

**file group** (read/write, list)

A list of the groups whose files should be migrated. A leading dash ('-') in a group name indicates negation, in which case all groups except the named group's files will be migrated.

*Example:* file group: staff, developers;

**preserve** (read/write, list)

A list of regular expressions, in the client shell syntax (for example, /bin/sh syntax on UNIX). A filename which matches any pattern in the list will be preserved and will never be migrated.

*Example:* preserve: \*.exe \*.dll;

**statistics** (read/write, hidden, list)

A list of statistics about recent migration activity for the save set(s) managed using this resource. The first value is the last update time. Subsequent groups of values contain a save set name, and statistics. The statistics are currently a date, K-bytes

pre-migrated, files pre-migrated, K-bytes stubbed, files stubbed, K-bytes de-migrated and files de-migrated.

**update statistics** (read/write, hidden, choice)

This controls whether the statistics in this resource should be updated to match the current values on the client. Selecting "Yes" causes the statistics to be updated, but the attribute value will not actually change.

*Example:* update statistics: No;

## Examples

**Note:** The hidden options are not shown in these examples.

A resource to define an HSM client, called elantra, migrating and recalling files in the /test file system from a Legato Storage Manager server:

```

type:                NSR migration;
name:                "elantra:/test";
client:              elantra;
save set:            /test;
enabled:             Yes;
directive:           Unix with compression directives ;
group:               Default;
high water mark (%): 90;
low water mark (%): 80;
last access time:   ;
minimum file size (KB): 5;
file owner:          joe, dave;
file group:          staff, developers;
preserve:            *.exe *.dll;

```

A resource to define an HSM client, called elantra, migrating and recalling files owned by the user "karl" in all file systems except for /, /usr, and /var (and /opt on solaris), to the Legato Storage Manager server jupiter:

```

type:                NSR migration;
name:                "elantra:All";
client:              elantra;
save set:            All;
enabled:             Yes;
directive:           Unix with compression directives ;
group:               Default ;
high water mark (%): 90;
low water mark (%): 60;
last access time:   ;

```

```
minimum file size (KB): 10;  
file owner:             ;  
file group:            ;  
preserve:              ;
```

**See Also:** [nsr\(5\)](#), [nsr\\_directive\(5\)](#), [nsr\\_group\(5\)](#), [savegrp\(8\)](#), [savefs\(8\)](#), [nsrpmig\(8\)](#), [nsrmig\(8\)](#), [nsrhmck\(8\)](#), [nsradmin\(8\)](#), [nsrexecd\(8\)](#), [nwadmin\(8\)](#)

## nsr\_notification(5)

### Name

**nsr\_notification** – Legato Storage Manager resource type “NSR notification”

### Synopsis

**type: NSR notification**

### Description

A resource of type **NSR notification** is used for each combination of an event, priority, and action handled by the Legato Storage Manager notification system. A Legato Storage Manager notification consists of a single event type, a single priority, and a message. The notification system posts each message to the action of each **NSR notification** resource (by executing the command listed in the action, with the message on standard input) that includes that event type and priority. See **nsr\_resource(5)** for more information on Legato Storage Manager resources. To edit the NSR notification resources type:

```
nsradmin -c "type:NSR notification"
```

or use the **nwadmin(8)** GUI. See **nsradmin(8)** for more information on using the Legato Storage Manager administration program.

### Attributes

The following attributes are defined for resource type **NSR notification**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed by an administrator, except when the resource is created. **Read/write** means the value can be changed at any time by authorized administrators. **Choice list** means that any number of values can be chosen from the given list. **Single string** means that only a single value is allowed. **Static** attributes change values rarely, if ever. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or expert mode (**-x**) in **nwadmin(8)**. For example, an attribute marked (**create-only, static**) has a value

which is set when the attribute is created and never changes. Several additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only, static)

This attribute holds the name of the notification resource.

**event** (create-only, choice list, hidden)

Each value is a class of events that will trigger the given notification. More than one class may be selected. Valid values are: **Media** for events related to the media multiplexor subsystem, **Savegroup** for events generated by the **savegrp(8)** command (usually the nightly automatic backups), **Index** for events related to the on-line file index subsystem, **Registration** for events caused by changes in the product's registration status (for example, a license that will soon time out), and **Server** for other Legato Storage Manager server events, such as restarting.

*Example:* event: Media;

**priority** (create-only, choice list, hidden)

Each value is a priority at which the notification will be triggered. More than one priority may be selected. The valid values in increasing priority order are **Info** – supplies information about the current state of the server; **Notice** – an important piece of information; **Warning** – information about a non-fatal error; **Waiting** – the server is waiting for an operator to perform a routine task, such as mounting a tape; **Critical** – the server detected an error condition that should be fixed by a qualified operator; **Alert** – a severe error condition that demands immediate attention; **Emergency** – a condition that may cause Legato Storage Manager to fail unless corrected immediately.

*Example:* priority: Notice;

**action** (read/write, single string)

The value is a command line to be executed when the given event occurs. The command line is run (see **popen(3s)**) with the event information connected to standard input. Typical actions are to log the message with the **syslog(3)** package, or send electronic mail to a system operator.

*Example:* action: /usr/ucb/mail -s "savegroup completion" root;

## Example

A complete example follows with two resources, one for mail and one using the syslog mechanism:

```
type:                NSR notification;
name:                savegroup completion;
administrator:      root;
```

```
action:          /usr/ucb/mail -s \"savegroup completion\"
                 root;
event:           Savegroup;
priority:        Notice;

type:            NSR notification;
name:            log default;
administrator:   root;
action:          /usr/ucb/logger -p daemon.notice -f -;
event:           Media, Savegroup, Index, Server,
                 Registration;
priority:        Info, Notice, Warning, Waiting,
                 Critical, Alert, Emergency;
```

## Files

**/nsr/res/nsr.res** – this file should never be edited directly. Use **nsradmin(8)** instead.

**See Also:** [nsr\\_resource\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_device\(5\)](#), [nsr\(8\)](#), [nsrmm\(8\)](#), [syslog.conf\(5\)](#), [syslog\(3\)](#), [nsradmin\(8\)](#), [nsrmmmd\(8\)](#), [nwadmin\(8\)](#)

## nsr\_policy(5)

### Name

**nsr\_policy** – Legato Storage Manager resource type “NSR policy”

### Synopsis

**type: NSR policy**

### Description

Each Legato Storage Manager policy is described by a single resource of type **NSR policy** (see **nsr\_resource(5)**). To edit the **NSR policy** resources for a Legato Storage Manager server, type:

```
nsradmin -c "type:NSR policy"
```

See **nsradmin(8)** for more information on using the Legato Storage Manager administration program.

These resources control how long entries remain in a client’s on-line file index and when to mark a save set as recyclable. Each **NSR client** resource (see **nsr\_client(5)**) uses two policies, a browse policy and a retention policy. Policies can only be deleted if no clients are using them.

Each policy defines an amount of time. The amount of time is determined by the *period* and the *number of periods*.

### Attributes

The following attributes are defined for resource type **NSR policy**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed by an administrator once the resource is created. **Read/write** means the value can be set as well as read at any time. Several additional hidden attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (create-only)

This attribute contains the name of the policy defined by this resource. The name must be unique for this Legato Storage Manager server, but otherwise can be anything that makes sense to the administrator. This name will appear as a choice attribute of each NSR client resource. The **NSR policy** resources named "Quarter"

and "Year" may be modified, but may not be removed. The name can only be specified when the group is created.

*Example:* name: life cycle;

**period** (read/write)

The period attribute determines the base unit for this policy. It may be one of four values: **Days**, **Weeks**, **Months** or **Years**. A week is defined as 7 days, a month as 31 days, and a year as 366 days.

*Example:* period: Months;

**number of periods** (read/write)

The number of periods attribute specifies the number of base units to use.

*Example:* number of periods: 3;

## Example

The following **NSR policy** resource named "Quarter" defines a period of 3 months, or one quarter of a year:

```
type:                NSR policy;
name:                Quarter;
period:              Months;
number of periods:   3;
```

**See Also:** [nsr\(8\)](#), [nsrim\(8\)](#), [nsr\\_resource\(5\)](#), [nsr\\_client\(5\)](#),  
[nwadmin\(8\)](#), [nsradmin\(8\)](#)

## nsr\_pool(5)

### Name

**nsr\_pool** – Legato Storage Manager resource type “NSR pool”

### Synopsis

**type: NSR pool**

### Description

Each NSR pool is described by a single resource of type **NSR pool** (see **nsr\_resource(5)**). To edit the NSR pool resources for a Legato Storage Manager server type:

```
nsradmin -c "type:NSR pool"
```

Be careful to include the quotation marks and the space between “NSR” and “pool”. See the **nsradmin(8)** manual page for more information on using the Legato Storage Manager administration program.

These resources are used by Legato Storage Manager to determine what volumes save sets should reside on depending upon the characteristics (for example, Group or Level) of the save. Consult your *Legato Storage Manager Administrator's Guide* for more guidelines on using pools.

There are four types of pools. *Backup* pools accept data from **savegrp** and manual backups. *Archive* pools accept archive data. Data cloned from a backup pool can be directed to a *backup clone* pool. Likewise, archive data can be cloned to a *archive clone* pool.

There are four pools shipped pre-enabled with Legato Storage Manager. The *Default* pool is meant to collect any backup data not directed to a pool a user creates with selection criteria. Any archive data not directed to a pool with selection criteria is collected in the *Archive* pool. When cloning data, the user must select a destination pool for the operation. The *Default clone* pool is available for users to clone backup data to. The *Archive clone* pool is available for users to clone archive data to.

There are also a few pools shipped with Legato Storage Manager that are not enabled by default. The *Full* and *NonFull* pools can be used to segregate full level backups from other backups, for example, fulls versus incrementals. The *Offsite* pool can be used to generate offsite backups, because no index entries are stored for the media pool and will not be referenced during normal recovers. Note that one

can also clone media to produce copies of data to be taken offsite. Save sets that are generated without index entries can still be recovered using “Save Set Recover” feature of **nwadmin(8)** or **recover(8)**.

## Attributes

The following attributes are defined for resource type **NSR pool**. The information in parentheses describes how the attribute values are accessed. **Create-only** indicates that the value cannot be changed after the resource has been created. **Read/write** means the value can be updated by authorized administrators. **Yes/no** means only a yes or no choice is possible. **Choice** indicates that the value can only be selected from a given list. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or if the *Details View* option is selected in the *Media Pools* window in **nwadmin(8)**.

**name** (create-only)

The names of pool resources are used when labeling volumes and when determining what volumes a save set should reside on. The name can generally be chosen at the administrator’s convenience, but it must be unique for this Legato Storage Manager server. The pool resources named **Default**, **Default Clone**, **Archive**, and **Archive Clone** may not be modified or deleted. The pool resource named **Full** and **NonFull** may not be deleted. Other pools can only be deleted if no volumes still reference them.

*Example:* name: Accounting;

**groups** (read/write, choice)

What *groups* (**nsr\_group(5)**) are allowed in this pool.

*Example:* groups: Accounting;

**clients** (read/write, choice)

What *clients* (**nsr\_client(5)**) are allowed in this pool. If a *group* is specified, only clients that are members of that group are allowed to be listed.

*Example:* clients: mars;

**save sets** (read/write, choice)

What save sets (**nsr\_client(5)**) are allowed in this pool. Save sets can be matched using the regular expression matching algorithm described in **nsr\_regexp(5)**.

*Example:* save sets: /, /usr;

**levels** (read/write, choice)

What *levels* (**nsr\_schedule(5)**) are allowed in this pool.

*Example:* levels: full;

**archive only** (read/write, yes/no, hidden, create)

If **yes** is selected, only archive saves are allowed to this pool. *This hidden attribute can be modified by a user.*

*Example:* archive only: no;

**status** (read/write, hidden, choice)

If set to **enabled**, this pool is considered for determining what pools a save set should be saved to when performing backup volume selection. If set to **clone**, this pool is considered only as the destination of cloning operations. If set to **disabled**, this pool is completely ignored. *This hidden attribute can be modified by a user.*

*Example:* status: enabled;

**label template** (read/write, choice)

Determine what *label template* (**nsr\_label**(5)) is referenced when generating volume names for this pool.

*Example:* label template: Accounting;

**devices** (read/write, choice)

This attribute lists the **only** devices volumes from this pool are allowed to be mounted onto. If no devices are listed, volumes from this pool may be mounted on any device.

*Example:* devices: /dev/nrst8;

**store index entries** (read/write, yes/no, choice)

If set to **yes**, entries are made into the file indexes for the backups. Otherwise, only media database entries for the save sets are created.

*Example:* store index entries: yes;

**auto media verify** (read/write, yes/no, choice)

If set to **yes**, Legato Storage Manager verifies data written to volumes from this pool. Data is verified by re-positioning the volume to read a portion of the data previously written to the media and comparing the data read to the original data written. If the data read matches the data written, verification succeeds otherwise it fails. Media is verified whenever a volume becomes full while saving and it is necessary to continue onto another volume, or when a volume goes idle because all save sets being written to the volume are complete. When a volume fails verification it is marked full so Legato Storage Manager will not select the volume for future saves. The volume remains full until it is recycled or a user marks it not full. If a volume fails verification while attempting to switch volumes all save sets writing to the volume are terminated.

*Example:* auto media verify: yes;

**recycle to other pools** (read/write, yes/no, choice)

This attribute determines whether or not a given pool allows other pools to recycle

its recyclable volume for their use.

*Example:* recycle to other pools: yes;

**recycle from other pools** (read/write, yes/no, choice)

This attribute determines whether or not a given pool can recycle volumes from other pools when it exhausts all its writable and recyclable volumes.

*Example:* recycle from other pools: yes;

**volume type preference** (read/write, choice)

The volume selection preference attribute is used as a selection factor when a request is made for a write-able volume. The preferred type will be considered first within a priority level such as *jukebox* or *standalone device*.

*Example:* volume type preference: 4mm;

## Example

A complete NSR pool resource, named 'Default', follows:

```

type:                NSR pool;
archive only:       No;
clients:             ;
devices:             ;
groups:              ;
label template:     Default;
levels:              ;
name:                Default;
save sets:           ;
status:              Enabled;
store index entries: Yes;
auto media verify:  Yes;
recycle from other pools: Yes;
recycle from other pools: Yes;
volume type preference: 4mm;

```

**See Also:** [nsr\(5\)](#), [nsr\\_label\(5\)](#), [nsr\\_resource\(5\)](#), [nwrecover\(8\)](#), [recover\(8\)](#), [savegrp\(8\)](#), [savefs\(8\)](#), [uasm\(8\)](#)

## nsr\_regexp(5)

### Name

**nsr\_regexp** – Regular expression syntax

### Description

This manual page describes the regular expression handling used in Legato Storage Manager. The regular expressions recognized are described as follows. This description is essentially the same as that for **ed**(1).

A regular expression specifies a set of strings of characters. A member of this set of strings is said to be matched by the regular expression.

#### Form Description

1. Any character except a special character matches itself. Special characters are the regular expression delimiter plus a backslash(\), brace{ }, or period(.) and sometimes a caret(^), asterisk(\*), or dollar symbol(\$), depending upon the following rules.
2. A . matches any character.
3. A \ followed by any character except a digit or a parenthesis matches that character.
4. A nonempty string **s**, bracketed string [**s**] (or [^**s**]) matches any character in (or not in) **s**. In **s**, \ has no special meaning and ] may only appear as the first letter. A substring **a-b**, with **a** and **b** in ascending ASCII order, stands for the inclusive range of ASCII characters.
5. A regular expression of form 1 through 4 followed by \* matches a sequence of 0 or more matches of the regular expression.
6. A bracketed regular expression **x** of form 1 through 8, \(\b), matches what **x** matches.
7. A \ followed by a digit **n** matches a copy of the string that the bracketed regular expression beginning with the **n**th \(\b) matched.
8. A regular expression **x** of form 1 through 8 followed by a regular expression **y** of form 1 through 7 matches a match for **x** followed by a match for **y**, with the **x** match being as long as possible while still permitting a **y** match.

9. A regular expression of form 1 through 8 preceded by ^ (or followed by \$), is constrained to matches that begin at the left (or end at the right) end of a line.
10. A regular expression of form 1 through 9 picks out the longest among the leftmost matches in a line.
11. An empty regular expression stands for a copy of the last regular expression encountered.

**See Also:** [ed\(1\)](#), [nsr\\_client\(5\)](#)

## nsr\_repack\_schedule(5)

### Name

**nsr\_repack\_schedule** – Legato Storage Manager resource type “NSR repack schedule”

### Synopsis

**type: NSR repack schedule**

### Description

Each Legato Storage Manager repack schedule is described by a single resource of type **NSR repack schedule** (see **nsr\_resource(5)**). To edit the **NSR repack schedule** resources for a Legato Storage Manager server, type:

**nsradmin -c "type:NSR repack schedule"**

or use the **nwadmin(8)** GUI. See **nsradmin(8)** for more information on using the Legato Storage Manager administration program.

This resource describes what day(s) that a volume may be repacked (see **nsrrepack(8)** and **nsr\_pool(5)**). There is one **NSR repack schedule** resource for each Legato Storage Manager repack schedule.

### Attributes

The following attributes are defined for resource type **NSR repack schedule**. The information in parentheses describes how the attribute values are accessed.

**Read-only** indicates that the value cannot be changed by an administrator.

**Read/write** means the value can be set as well as read. Several additional hidden attributes (for example, **administrator**) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read/write)

Specifies the repack schedule’s name. The repack schedule is referred to by its name in client resources.

*Example:* name: monthly\_fulls;

**period** (read-only)

Specifies the length of the repack schedule's period. It may be either "Week" or "Month". "Week" repack schedules repeat every 7 days and start on Sunday. "Month" repack schedules start over at the first of each month. The default is "Week".

*Example:* period: Month;

**action** (read/write)

Specifies the sequence of actions making up the repack schedule. One entry is used for each day of the repack schedule. The entries must be separated by white space, for example, blanks or tabs. The valid actions are 'repack' and 'skip'. The actions repack and skip may be abbreviated 'r' and 's', respectively. 'Repack' permits repacking while 'skip' disallows it.

When the action attribute does not contain enough entries to account for every day in the period, Legato Storage Manager will repeat the list of actions when the end of the action list is reached.

*Example:* action: r s s s s s s;

**override** (read/write)

Specifies a list of actions and dates overriding the actions specified in the *action* attribute. The format of the override specification is *actiondate*. *action* must be one of 'repack' or 'skip'. *date* must be of the form 'month/day/year'. Action/date pairs are separated by a commas (.). Month and day are 2-digit numbers; year may be either 2 or 4 digits. If the year is 2 digits, numbers in the range 70-99 are assumed to be offsets from 1900; those in the range 00-69 are assumed to be offset from 2000.

*Example:* override: repack 1/1/1999, repack 6/1/1999;

## Examples

The following defines an **NSR repack schedule** resource named 'Default'. The Default repack schedule may be modified, but it may not be deleted. Each Legato Storage Manager server must have a Default repack schedule. This repack schedule has a period of one week, does a repack on Sunday, followed by 6 skips. There are no override actions specified.

```

type:      NSR repack schedule;
name:      Default;
period:    Week;
action:    r s s s s s s;
override:  ;

```

The following defines a repack schedule named 'quarterly'. It has a period of one month. The action attribute specifies repack and skips. In the override attribute, repacking is specified for the first day of each quarter. Note that there are only 7 entries in the action attribute. Upon reaching the end of the list, Legato Storage Manager will start over at the beginning of the list, performing a repack.

```
type:      NSR repack schedule;
name:      quarterly;
period:    Month;
action:    repack skip skip skip repack skip skip;
override:  repack 1/1/1998, repack 3/1/1998, repack
           6/1/1998, repack 9/1/1998, repack 1/1/1999;
```

**See Also:** [mminfo\(8\)](#), [nsr\(8\)](#), [nsr\\_pool\(5\)](#), [nsrrepack\(8\)](#),  
[nsradmin\(8\)](#), [nwadmin\(8\)](#)

## nsr\_resource(5)

### Name

**nsr\_resource** – Legato Storage Manager resource format

### Synopsis

```
resource ::= attribute list <blank line>  
attribute list ::= attribute [ ; attribute ]*  
attribute ::= name [ : value [ , value ]*  
name, value ::= <printable string>
```

### Description

The Legato Storage Manager system uses files containing *resources* to describe itself and its clients. Each resource represents a component of the Legato Storage Manager system that might need administration. Devices, schedules, and clients are examples of Legato Storage Manager resources. The system administrator manipulates resources to control the Legato Storage Manager system. The file and the resources in them are accessible through the **nwadmin(8)** and the **nsradmin(8)** programs. They can also be viewed with a normal text editor.

The files all share a common format. The same format is used by the **nsradmin(8)** program. Each resource is described by a list of attributes, and ends in a blank line. Each attribute in the attribute list has a name and an optional list of values. The attribute name is separated from the attribute values by a colon (:), attribute values are separated by commas (,), and each attribute ends in a semicolon (;). A comma, semicolon or back-slash (\) at the end of a line continues the line. A line beginning with a pound-sign (#) is a comment and the rest of the line is ignored. The back-slash character can also be used to escape the special meaning of other characters (comma, semicolon, pound-sign, and back-slash).

The attribute name and values can contain any printable character. Uppercase and lowercase is not distinguished on comparisons, and extra white space is removed from both ends but not from inside of names and values. For example,

```
Name: this is a test;
```

matches

```
name : This Is A Test ;
```

but is different from

```
Name: this is a test;
```

In the following example resource, there are eight attributes. They are **type**, **name**, **server**, **schedule**, **directive**, **group**, **save set**, and **remote access**. The **remote access** attribute has no value.

```
type:          NSR client;
name:          venus;
server:        earth;
schedule:      Default;
directive:     Unix standard directives;
group:         Default;
save set:      All;
remote access: ;
```

In the following resource, there are six attributes. The **administrator** attribute has three values: **&engineering**, **root**, and **operator**. Note that the three values are separated by commas. The **action** attribute has one value: **incr incr incr incr full incr full incr**. Note that this is a single value – it just happens to have spaces separating its words.

```
type:          NSR schedule;
action:        incr incr incr incr full incr;
administrator: &engineering, root, operator;
name:          engineering servers;
override:     ;
period:       Week;
```

## Special Attributes

Each Legato Storage Manager resource includes seven special attributes: **type**, **name**, **administrator**, **hostname**, **ONC program number**, **ONC version number**, and **ONC transport**. The **type** and **name** attributes are normally visible, but the others attributes are hidden. That an attribute is hidden indicates that it is

infrequently used and perhaps esoteric. Frequently, hidden attributes should not be changed by the user.

The **type** attribute defines which other attributes a resource can contain. For example, a resource with type **NSR client** will always include the attribute **server**, while a resource of type **NSR schedule** does not.

The **name** attribute is a descriptive name of the object that a resource represents. In the first example, the **name** attribute is the name of the Legato Storage Manager client computer. In the second example, the **name** attribute describes a schedule used to back up the servers in the engineering department.

The **administrator** attribute is the list of users that have permission to modify or delete this resource. This attribute is inherited from the **type: NSR** resource when a new resource is created. The administrator of the **NSR** resource also controls who has permission to create and delete Legato Storage Manager resources.

The **hostname** attribute specifies the hostname of the computer on which the service that controls this resource is running. It is used internally and cannot be changed by the administrator.

The remaining attributes (**ONC program number**, **ONC version number**, and **ONC transport**) specify the Open Network Computing information for this service. They should never be changed manually.

In some cases, the resource identifier will be visible. Although it may look like an attribute, it is an internal value that is set and used by the Legato Storage Manager system to provide unique identification of each resource. When new resources are created in the **edit** command of **nsradmin(8)**, the resource identifier attribute should be left off. This signals that this is a new resource and a new identifier will be assigned.

Legato Storage Manager resources are implemented by the Legato Resource Administration Platform, which is described in the **resource(5)** manual page. This flexible architecture means that in future releases of Legato Storage Manager, more resource types or attributes may be added, and the administration tools in this release will automatically be able to use them. To make this possible, each server provides *type descriptors* that are used internally to describe the attributes of each type, between the administration tools and the services. These type descriptors may cause limitation on the values, such as only allowing a single value, allowing no value, or only numeric values.

## Resource Types

This release of Legato Storage Manager defines the following types of resources:

### **NSR**

This resource describes a Legato Storage Manager server. It contains attributes that control administrator authorization, information about operations in progress, and statistics and error information about past operations. For more information see the **nsr\_service(5)** manual page.

### **NSR client**

This resource describes a Legato Storage Manager client. It includes attributes that specify the files to save, which schedule to use, and which group this client belongs to. There may be more than one client resource for a Legato Storage Manager client. This allows a client to save files on different schedules. For more information see the **nsr\_client(5)** manual page.

### **NSR device**

This resource type describes a storage device. It includes attributes that specify a particular device name (for example, */dev/nrst1*), media type (for example, 8mm), and the name of the currently mounted volume. It also provides status and statistics on current and past operations. For more information see the **nsr\_device(5)** manual page.

### **NSR directive**

This resource describes a directive. Directives control how a client's files are processed as they are being saved. For more information see the **nsr\_directive(5)**, the **nsr(5)** and the **uasm(8)** manual pages.

### **NSR group**

This resource specifies a logical grouping of Legato Storage Manager clients and a starting time. Each day, at the specified time, all members of the group will start their saves. For more information see the **nsr\_group(5)** manual page.

### **NSR jukebox**

This resource type describes a jukebox. It includes attributes such as the jukebox model, the first and last slot numbers in the jukebox, and the names of the devices within the jukebox.

### **NSR label**

This resource type specifies a template describing a sequence of names to be used when labeling volumes. For more information see the **nsr\_label(5)** manual page.

### **NSR license**

This resource contains licensing information for each feature currently enabled in this Legato Storage Manager installation. It contains various enabler and

authorization codes that are used by Legato Storage Manager to validate licensed capabilities. For more information see the **nsr\_license(5)** and **nsrcap(8)** manual pages.

### **NSR notification**

A notification specifies an action to be performed when a particular type of Legato Storage Manager event takes place. For more information see the **nsr\_notification(5)** manual page.

### **NSR policy**

Policy resources are used as part of the index management process in Legato Storage Manager. These policies control how long entries remain in a client's on-line file index and when to mark a save set as recyclable. For more information see the **nsr\_policy(5)** manual page.

### **NSR pool**

This resource type is used by Legato Storage Manager to determine what volumes save sets should reside on based on the characteristics of the save (for example, group or level). For more information see the **nsr\_pool(5)** manual page.

### **NSR schedule**

Schedule resources define a sequence of save levels and an override list. The override list is made up of pairs of levels and dates. The level controls the amount of data saved when a client is backed up. For more information see the **nsr\_schedule(5)** manual page.

### **NSR stage**

Each stage resource describes a staging policy. The resource includes attributes that define control parameters for the policy, and devices managed by the policy. For more information see the **nsr\_stage(5)** manual page.

## **Files**

### **/nsr/res/nsr.res**

Holds the Legato Storage Manager server's resources.

**See Also:** [nsr\(5\)](#), [nsr\\_client\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_directive\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_label\(5\)](#), [nsr\\_license\(5\)](#), [nsrcap\(8\)](#), [nsr\\_notification\(5\)](#), [nsr\\_policy\(5\)](#), [nsr\\_pool\(5\)](#), [nsr\\_schedule\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_stage\(5\)](#), [nsr\(8\)](#), [nwadmin\(8\)](#), [savegrp\(8\)](#), [savefs\(8\)](#), [nsradmin\(8\)](#), [uasm\(8\)](#)

## nsr\_schedule(5)

### Name

**nsr\_schedule** – Legato Storage Manager resource type “NSR schedule”

### Synopsis

**type: NSR schedule**

### Description

Each Legato Storage Manager schedule is described by a single resource of type **NSR schedule** (see **nsr\_resource(5)**). To edit the **NSR schedule** resources for a Legato Storage Manager server, type:

```
nsradmin -c "type:NSR schedule"
```

or use the **nwadmin(8)** GUI. See **nsradmin(8)** for more information on using the Legato Storage Manager administration program.

This resource describes a sequence of levels controlling the amount of data saved by Legato Storage Manager clients (see **nsr\_client(5)**). There is one **NSR schedule** resource for each Legato Storage Manager schedule.

### Attributes

The following attributes are defined for resource type **NSR schedule**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. Several additional hidden attributes (for example, **administrator**) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read/write)

This attribute specifies the schedule’s name. The schedule is referred to by its name in client resources.

*Example:* name: monthly\_fulls;

**period** (read-only)

This attribute specifies the length of the schedule's period. It may be either "Week" or "Month". "Week" schedules repeat every 7 days and start on Sunday. "Month" schedules start over at the first of each month. The default is "Week".

*Example:* period: Month;

**action** (read/write)

This attribute specifies the sequence of save levels making up the schedule. One entry is used for each day of the schedule. The entries must be separated by white space, that is, blanks or tabs. The valid levels are 'consolidate', 'full', 'incr', 'skip', and the numbers 1 through 9. The actions consolidate, full, incr, and skip may be abbreviated as 'c', 'f', 'i', and 's', respectively.

When the action attribute does not contain enough entries to account for every day in the period, Legato Storage Manager will repeat the list of actions when the end of the action list is reached.

*Example:* action: f i i i i i;

**override** (read/write)

This attribute specifies a list of actions and dates overriding the actions specified in the *action* attribute. The format of the override specification is *action date*. *action* must be one of 'full', 'incr', 'skip' or one of the numbers 1 through 9. *date* must be of the form 'month/day/year'. Action/date pairs are separated by commas (','). Month and day are 2 digit numbers, year may be either 2 or 4 digits. If the year is 2 digits, numbers in the range 70-99 are assumed to be offsets from 1900, those in the range 00-69 are assumed to be offset from 2000.

*Example:* override: full 1/1/1999, full 6/1/1999;

## Examples

The following defines a **NSR schedule** resource named 'Default'. The Default schedule may be modified, but it may not be deleted. Each Legato Storage Manager server must have a Default schedule. This schedule has a period of one week, does a full on Sunday, followed by 6 incremental saves. There are no override actions specified.

```
type:          NSR schedule;
name:          Default;
period:        Week;
action:        f i i i i i i;
override:     ;
```

The following defines a schedule named 'quarterly'. It has a period of one month. The action attribute specifies level 5, 9, and incremental saves. In the override attribute, full saves are specified for the first day of each quarter. Note that there are only 7 entries in the action attribute. Upon reaching the end of the list, Legato Storage Manager will start over at the beginning of the list, performing a level 5 save.

```
type:          NSR schedule;
name:          quarterly;
period:        Month;
action:        5 incr incr incr 9 incr incr;
override:     f 1/1/1998, f 3/1/1998, f 6/1/1998,
              f 9/1/1998, f 1/1/1999;
```

**See Also:** [nsr\(8\)](#), [savefs\(8\)](#), [mminfo\(8\)](#), [nsradmin\(8\)](#), [nsr\\_client\(5\)](#), [nsr\\_policy\(5\)](#), [nsr\\_resource\(5\)](#), [nwadmin\(8\)](#)

---

## nsr\_service(5)

### Name

**nsr\_service** – Legato Storage Manager server resource type “NSR”

### Synopsis

**type: NSR**

### Description

Each Legato Storage Manager server is described by a resource of type **NSR**. See **nsr\_resource(5)** for general information on Legato Storage Manager resources. To edit the **NSR** resource use the command:

```
nsradmin -c "type:NSR"
```

or use the **nwadmin(8)** GUI. See **nsradmin(8)** for information on using the Legato Storage Manager administration program.

### Attributes

The following attributes are defined for the **NSR** resource. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Choice list** means that any number of values can be chosen from the given list. **Static** attributes change values rarely, if ever. **Dynamic** attributes have values which can change rapidly. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(8)** or details option is set in **nwadmin(8)**. For example, an attribute marked (**read-only, static**) has a value which is set when the resource is created and never changes, or is changed only by the server.

**name** (read-only, static)

This attribute specifies the hostname of this Legato Storage Manager server.

*Example name:* mars;

**version** (read-only, dynamic)

This is the software version of the Legato Storage Manager server daemon, **nsrd(8)**. This includes a slash and the number of clients currently licensed.

*Example:* version: Legato Storage Manager 4.1 Turbo/110;

**save totals** (read-only, dynamic, hidden)

Save statistics. A string containing the total number of save sessions, the number of saves with errors (if any) and the total number of bytes saved (if any). This attribute is updated after each save session completes.

*Example:* save totals: "37 sessions, 457 MB total";

**recover totals** (read-only, dynamic, hidden)

Recovery statistics. A string containing the total number of recover sessions, the number of recovers with errors (if any) and the total number of bytes recovered (if any). This attribute is updated after each recover session completes.

*Example:* recover totals: "347 sessions, 48 MB total";

**totals since** (read-only, dynamic)

The time statistics collection started. This is usually the last time the Legato Storage Manager server was rebooted.

*Example:* totals since: "Fri Jun 1 09:35:02 1999";

**NSR operation** (read-only, choice list, hidden)

This attribute is currently unused and is provided for backward compatibility.

**parallelism** (read/write, static)

This attribute sets the number of concurrent save sessions that this server will allow. The value can be set by an administrator. Use higher values for better performance on a fast system with lots of main memory and swap space, and use lower values to avoid overloading a slow system, or systems with little main memory and/or swap space. **Warning:** due to bugs in some versions of UNIX, high values of parallelism may cause the system to lock up.

*Example:* parallelism: 4;

**session statistics** (read-only, dynamic, hidden)

This attribute reports the statistics of each active session. There are 14 values for each set of statistics, namely, **id** (session's unique identifier), **name** (session's name), **mode** (read, write, browse), **pool** (current pool), **volume** (current volume), **rate kb** (current data transfer rate for save session), **amount kb** (current amount read/written by session), **total kb** (total amount to be read by session), **amount files** (current number of files recovered; to be implemented in a future release), **total files** (current number of files to recover; to be implemented in a future release), **connect time** (time session has been connected), **num volumes** (number of volumes to be used by recover session), **used volumes** (number of volumes processed by recover session), and **completion** (running, complete, or continued)

*Example:* sessions statistics: ;

**manual saves** (read/write, hidden)

This attribute allows the administrator to disable manual backups to the server. Scheduled backups continue to work normally.

**volume priority** (read/write)

If a Legato Storage Manager server has volumes in locally managed jukeboxes and volumes being managed by SmartMedia, this attribute allows the administrator to set a priority for selecting volumes to be used while saving data. Determines whether the server has a preference for volumes being managed by SmartMedia, **SmartMedia Priority**, or whether the server has a preference for volumes in a locally managed jukebox, **NearLine Priority**, when selecting a volume to be used to record data while saving. The default value is **NearLine Priority**.

*Example:* volume priority: NearLine Priority;

**SmartMedia save mount** (read/write)

This attribute controls the form of the request made to SmartMedia to mount a volume for saving data. Setting this attribute's value to **volume by characteristics**, causes Legato Storage Manager to request a volume meeting specified criteria, and lets SmartMedia select an appropriate volume from all media which satisfy the criteria specified. When this attribute's value is set to **volume by name**, Legato Storage Manager will request the volume by name and SmartMedia mounts the volume requested. The default value is **volume by characteristics**.

*Example:* SmartMedia save mount: volume by characteristics;

**license server** (read/write, hidden)

The name of the server on which a Legato license manager is installed and running. This attribute used to be called "GEMS server". You can set the value to a GEMStation where a GEMS license manager is running or to a computer where a Legato license manager is running.

*Example:* license server: maruthi;

**message** (read-only, dynamic, hidden)

The last message of any kind logged. A time stamp is included at the start of the string.

*Example:* message: "Mon 12:25:51 Tape full, mount volume mars.001 on /dev/nrst1";

**message list** (read-only, dynamic, hidden)

A list of recent messages, with a time stamp and a string message for each value.

*Example:* message: "Mon 12:25:51 Tape full, mount volume mars.001 on /dev/nrst1";

**session** (read-only, dynamic, hidden)

The value of this attribute is a list of session output strings. Each string includes the Legato Storage Manager client name, type of operation (saving, browsing, or recovering) and information about the save set, including name, number of bytes,

and number of files. All sizes and rates are in bytes per second, Kilobytes (1024), Megabytes (a thousand Kilobytes), and so on.

*Example:*

```
session: "venus:/usr saving to mars.001 20MB",  
        "mars:/usr/src done saving 24MB";
```

**pending** (read-only, dynamic, hidden)

A list of events pending with the Legato Storage Manager event notification system (see **nsr\_notification**(5)). The first three fields are the time, priority, and event name.

*Example:* pending: "Fri 14:40:15 alert: media mount of mars.001 suggested on /dev/nrst1";

**status** (read-only, dynamic, hidden)

A list of status flags for the Legato Storage Manager server. These flags are only for use by Legato Storage Manager server-side programs (for example, **savegrp**) and list various features enabled in the running server. The format is currently *name=boolean* (true or false). The features listed, and their states, can change at any time.

**statistics** (read-only, dynamic, hidden)

A list of strings of the form *name=number* that give a number of server statistics.

**types created** (read-only, static)

A list of all the other resource types this Legato Storage Manager server can create and about which clients can query.

*Example:* types created: NSR device, NSR group;

**administrator** (read/write, static)

This is a list of names (or netgroups) of users who are allowed to administer Legato Storage Manager. Normally this list is inherited by all other resources on this server, although each administrator attribute can be explicitly changed if desired.

Administrators can change the values of attributes for the resource which lists them.

The administrator list also determines who can add and delete resources of the

other NSR types. The user "root" on the local host of the server is always an

administrator. Entries specifying other administrators are of the form, user,

user@host, host/user, or &netgroup. The username and/or host name may be a

wild card, "\*" (any user and/or host). Entering just a username allows that user to administer Legato Storage Manager from any host (equivalent to user@\* or \*/user).

Netgroup names are always preceded by an "&". The following example grants

Legato Storage Manager administrative privileges to, "root" from any host, the user

"operator" from the hosts "jupiter" and "mars", the user "admin" from any host, and

all <username, user's hostname, server's domain> in the netgroup "netadmins".

*Example:* administrator: root, operator@jupiter, mars/operator, admin@\*,

&netadmins;

**contact name** (read/write, static)

This attribute is used for product licensing/registration purposes. It must be specified before printing the registration information from the registration window.

*Example:* contact name: contact\_name;

**company** (read/write, static)

This attribute is used for product licensing/registration purposes. Your company name must be specified before printing the registration information from the registration window.

*Example:* company: Legato;

**street address** (read/write, static)

This attribute is used for product licensing/registration mailing purposes. Specify your mailing street address.

*Example:* street address: 3145 Porter Drive;

**city/town** (read/write, static)

This attribute is used for product licensing/registration mailing purposes.

*Example:* city/town: Palo Alto;

**state/province** (read/write, static)

This attribute is used for product licensing/registration mailing purposes.

*Example:* state/province: CA;

**zip/postal code** (read/write, static)

This attribute is used for product licensing/registration mailing purposes.

*Example:* zip/postal code: 94304;

**country** (read/write, static)

This attribute is used for product licensing/registration mailing purposes.

*Example:* country: USA;

**phone** (read/write, static)

This attribute is used for product licensing/registration information purposes. This attribute must to be specified before printing the registration information from the registration window.

*Example:* phone: 650-812-6100;

**fax** (read/write, static)

This attribute is used for product licensing/registration purposes.

*Example:* fax: 650-812-6031;

**email address** (read/write, static)

This attribute is used for product licensing/registration purposes.

*Example:* email address: support@us.oracle.com;

**server OS type** (read/write, static)

This attribute is used for product licensing/registration purposes.

*Example:* server OS type: Solaris;

**purchase date** (read/write, static)

This attribute is used for product licensing/registration purposes. It specifies the purchase date of the product enabler code. This attribute must be specified before printing the registration information from the registration window.

**product serial number** (read/write, static)

This attribute is used for product licensing/registration purposes. It must be specified before printing the registration information from the registration window.

**mm op message** (read/write, dynamic, hidden)

This attribute lists the descriptive message for the most recently completed media database operation. The Legato Storage Manager program (such as **nsrmm**(8)) that requested the operation clears this attribute as soon as it has read the result. An administrator should never change this attribute manually.

**mm operation value** (read/write, dynamic, hidden)

This attribute is used by programs such as **nsrmm**(8) to pass the desired media database operation location or flags to the Legato Storage Manager server. The value is automatically cleared when the operation completes. An administrator should never change this attribute manually.

**mm operation** (read/write, choice list, dynamic, hidden)

This attribute is used by programs such as **nsrmm**(8) to pass the media database operation type currently desired to the Legato Storage Manager server. The possible choices are: purge volume, purge save set, delete volume, delete save set, mark volume, mark save set, unmark volume, unmark save set, specify volume location, specify volume flags, and specify save set flags. The server serializes such operations and performs the appropriate queries on **nsrmmdbd**(8). The value is automatically cleared when the operation completes. An administrator should never change this attribute manually.

**mm operation id** (read/write, dynamic, hidden)

This attribute is used by programs such as **nsrmm**(8) to pass the desired media database operation identifier to the Legato Storage Manager server. The value is automatically cleared when the operation completes. An administrator should never change this attribute manually.

**nsrmon info** (read/write, dynamic, hidden)

This attribute is used by programs such as **nsrmon**(8) to pass information about remote daemon requests to the Legato Storage Manager server. The value is automatically cleared when the request completes. An administrator should never

change this attribute manually. See **nsr\_storage\_node(5)** for a description of storage nodes and remote daemons.

**nsrmmmd count** (read-only, dynamic, hidden)

This attribute is used by programs such as **nsrd(8)** to track the number and location of the media daemons, **nsrmmmd(8)**.

**nsrmmmd polling interval** (read/write, hidden)

This attribute specifies the number of minutes between polling events of a remote **nsrmmmd(8)**. **nsrd(8)** polls a remote **nsrmmmd(8)** at this interval, to determine whether it is running. If it determines from this poll that the daemon is no longer running, it will restart the **nsrmmmd(8)**, with a delay set by the 'nsrmmmd restart interval', see the following description. See **nsr\_storage\_node(5)** for additional details on this attribute and storage nodes.

**nsrmmmd restart interval** (read/write, hidden)

This attribute specifies the number of minutes between restart attempts of a remote **nsrmmmd(8)**. When **nsrd(8)** determines that a remote **nsrmmmd(8)** has terminated, it periodically attempts to restart the remote daemon. A value of zero for this attribute means the daemon should be restarted immediately. See **nsr\_storage\_node(5)** for additional details on this attribute and storage nodes.

**nsrmmmd control timeout** (read/write, hidden)

This attribute specifies the number of minutes **nsrd(8)** waits for storage node requests.

**enabler code** (read/write, dynamic, hidden)

This attribute specifies the enabler code for the base enabler of the server software.

**SS cutoff size** (read/write, hidden)

This attribute sets the default "save set cut off size" to be used when saving. A blank value uses the built in default value. A non blank value for this attribute consists of a number followed by KB, MB, or GB signifying Kilobytes, Megabytes, or Gigabytes.

## Example

A complete example follows:

```

type:                NSR;
name:                mars;
version:             Legato Storage Manager 4.1
                    Turbo/110;
save totals:         "84 sessions, 3597 MB total";
recover totals:     1 session;

```

```
totals since:                "Fri Oct 14 12:41:31 1999";
NSR operation:              Idle Write Read Verify label Label;
parallelism:                4;
manual saves:              Enabled Disabled ;
message:                    \
                             "Mon 14:37:25 media alert event:
                             recover waiting for 8mm tape
                             mars.001";
message list:               \
                             "Mon 07:10:12 media info: loading
                             volume man.001 into /dev/nrst11",
                             "Mon 07:10:33 /dev/nrst11 mount
                             operation in progress",
                             "Mon 07:11:15 /dev/nrst11 mounted
                             8mm 5GB tape man.001";
session:                    "mars:george browsing",
                             "mars:/home/mars starting recovery
                             of 9K bytes";
session statistics:         ;
pending:                    \
                             "Mon 14:40:15 media alert: recover
                             waiting for 8mm tape mars.001";
status:                     disabled=false, jukebox=true,
                             dm=true,
                             archive=true, cds=true, turbo=true,
                             single=false;
statistics:                 elapsed = 257415, saves = 1176,
                             recovers = 12, save KB = 12050007,
                             recover KB = 28272839, bad saves
                             = 0, bad recovers = 0, current
                             saves = 1, current recovers = 0,
                             max saves = 12, max recovers = 1,
                             mounts = 0, recover delays = 0,
                             saving daemons = 0, recovering
                             daemons = 0, idle daemons = 0;
types created:              NSR device, NSR group, NSR
                             directive, NSR notification, NSR
                             client, NSR policy, NSR schedule,
                             NSR pool, NSR label, NSR jukebox,
                             NSR license, NSR archive client,
                             NSR archive list;
administrator:              root;
contact name:               Technical Support;
company:                    Oracle;
street address:             3145, Porter Drive;
```

```
city/town:                Palo Alto;
state/province:           CA;
zip/postal code:         94304;
country:                  USA;
phone:                    650-812-6100;
fax:                      650-812-6031;
email address:            support@us.oracle.com;
CompuServe address:      76044,3423;
purchase date:           ;
product serial number:   ;
mm op message:           ;
mm operation value:      ;
mm operation:            ;
mm operation id:         ;
nsrmon info:             ;
nsrmm count:             "mars:2";
nsrmm polling interval:  3;
nsrmm restart interval:  2;
nsrmm control timeout:   5;
enabler code:            ;
SS cutoff size:         ;
```

## Files

**/nsr/res/nsr.res** – This file should never be edited directly. Use **nwadmin(8)** or **nsradmin(8)** instead.

**See Also:** [netgroup\(5\)](#), [nsr\(5\)](#), [nsr\(8\)](#), [nsr\\_device\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_notification\(5\)](#), [nsr\\_resource\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [nsradmin\(8\)](#), [nsrd\(8\)](#), [nsrmm\(8\)](#), [nsrmmdbd\(8\)](#), [nsrmon\(8\)](#), [nwadmin\(8\)](#), [recover\(8\)](#), [save\(8\)](#)

## nsr\_shutdown(8)

### Name

**nsr\_shutdown** – Stop a Legato Storage Manager server’s processes

### Synopsis

**nsr\_shutdown** [ **-a** ] [ **-A** ] [ **-d** ] [ **-n** ] [ **-q** ] [ **-s** ] [ **-v** ]

### Description

**nsr\_shutdown** kills Legato Storage Manager processes on a Legato Storage Manager server. This command is simpler than the procedure of using **ps**(1), **grep**(1), and **kill**(1).

### Options

**-a**

Kill all daemons. This is the same as using the **-A**, **-d**, and **-s** options.

**-A**

Kill any **nsralist**(8) processes.

**-d**

This is the default option; it kills the server daemons. These may include **nsrd**(8), **nsrindexd**(8), **nsrexecd**(8), **nsrib**(8), **nsrmmd**(8), and **nsrmmdbd**(8). Since **savegrp**(8), **nsrexec**(8), and **nsralist**(8) processes depend on the service daemons, they are also killed.

**-n**

Not really. Echoes the kill command without actually invoking it.

**-q**

Perform the shutdown quietly; don’t prompt for confirmation.

**-s**

Kill any **savegrp**(8) (and **nsrexec**(8)) processes.

**-v**

Verbose: Instruct the shell to print commands and their arguments as they are executed.

**See Also:** [nsralist\(8\)](#), [nsrd\(8\)](#), [nsrexec\(8\)](#), [nsrexecd\(8\)](#), [nsrindexd\(8\)](#), [nsrmmd\(8\)](#), [nsrmmdbd\(8\)](#), [savegrp\(8\)](#)

## nsr\_stage(5)

### Name

**nsr\_stage** – Legato Storage Manager resource type “NSR stage”

### Synopsis

**type: NSR stage**

### Description

Each staging policy used by a Legato Storage Manager server is described by a single resource of type **NSR stage**. See **nsr\_resource(5)** for information on Legato Storage Manager resources. To edit the **NSR stage** resources run:

```
nsradmin -c "type:NSR stage"
```

Be careful to include the space between “NSR” and “stage” and the surrounding quotation marks. See **nsradmin(1m)** for information on using the Legato Storage Manager administration program.

### Attributes

The following attributes are defined for resource type **NSR stage**. The information in parentheses describes how the attribute values are accessed. **Read-only** indicates that the value cannot be changed by an administrator. **Read/write** means the value can be set as well as read. **Hidden** means it is an attribute of interest only to programs or experts, and these attributes can only be seen when the hidden option is turned on in **nsradmin(1m)** or if the *Details View* option is selected in the *Stage* window in **nwadmin(1m)**. **Static** attributes change values rarely, if ever. **Dynamic** attributes have values which change rapidly. For example, an attribute marked **(read-only, static)** has a value which is set when the attribute is created and may never change. Additional attributes (for example, administrator) are common to all resources, and are described in **nsr\_resource(5)**.

**name** (read-only, single string)

The name attribute specifies the staging policy name.

**enabled** (read/write, choice)

The **enabled** attribute determines whether or not save sets are automatically staged from devices associated with this policy. It also enables and disables the periodic recover space operations. It may be one of two values: *Yes*, *No*

**max storage period** (read/write)

Specifies the maximum number of *days* for a save set in a given volume before it is *staged* to a different volume.

**high water mark (%)** (read/write)

The point at which save sets should be *staged*, measured as the percentage of available space used on the file system. Staging will continue until the lower mark is reached.

*Example:* high water mark (%): 90;

**low water mark (%)** (read/write)

The point at which the staging process should *stop*, measured as the percentage of available space used on the file system.

*Example:* low water mark (%): 80;

**Save set selection** (read/write)

Save set selection criteria for staging. It may be one of four values:

*largest save set*

*smallest save set*

*oldest save set*

or

*youngest save set.*

**Destination pool** (read/write)

The pool to which save sets should be sent (see **nsr\_pool(5)**).

**Devices** (read/write, multiple choice)

This attribute lists the *file type* devices associated with this policy.

**Recover space interval** (read/write, hidden)

The number of *hours* between recover space operations for save sets with no entries in the media database form file devices.

**Fs check interval** (read/write, hidden)

The number of *hours* between file system check operations.

**Start now** (read/write)

Updating this attribute will cause the selected operation to be triggered immediately on all devices associated with this policy. The attribute value will not actually change. Operation can be one of the following:

*Check fs* check file system and stage data if necessary.

*Recover space* recover space for save sets with no entries in the media database (garbage collection).

*Stage all save sets* stage all save sets to the destination pool.

## Examples

**Note:** The hidden options are not shown in the first example.

The following example shows a resource that defines a stage policy called 'test stage1'. Save sets will be staged from device '/disk/fd0' to pool 'Default Clone' when the file system is 90% full or 7 days after the date of the backup, whichever comes first. The largest save set will be the first to stage to the destination pool:

```
type:                NSR stage;
name:                test stage1;
autostart:           Enabled;
max storage period:  7;
high water mark (%): 90;
low water mark (%):  85;
save set selection:  largest save set;
destination pool:    Default Clone;
devices:              /disk/fd0;
start now:           ;
```

The following example shows a resource that defines a stage policy called 'test stage2'. Save sets will be staged from device '/disk/fd2' to pool 'Default' when the file system is 95% full or 14 days after the date of the backup, whichever comes first. The smallest save set will be the first to stage to the destination pool. The file system will be checked every 1 hour and a staging operation will be triggered if necessary. A recover-space operation (for save sets with no entries in the media database) will be triggered every 3 hours on all devices associated with the policy:

```
type:                NSR stage;
name:                test stage2;
autostart:           Enabled;
max storage period:  14;
high water mark (%): 95;
low water mark (%):  80;
save set selection:  smallest save set;
destination pool:    Default;
devices:              /disk/fd2;
recover space interval: 3;
fs check interval:   1;
start now:            ;
administrator:        root@omni;
hostname:              omni;
```

**See Also:**    [nsrclone\(8\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#)

## nsr\_storage\_node(5)

### Name

**storage node** – Description of the storage node feature

### Synopsis

The *storage node* feature provides central server control of distributed devices for saving and recovering client data.

### Description

A *storage node* is a host that has directly attached devices that are used and controlled by a Legato Storage Manager server. These devices are called *remote devices*, since they are remote from the server. Clients may save and recover to these *remote devices* by altering their "storage nodes" attribute (see **nsr\_client(5)**). A *storage node* may also be a client of the server, and may save to its own devices.

The main advantages provided by this feature are central control of remote devices, reduction of network traffic, use of faster local saves and recovers on a storage node, and support of heterogeneous server and storage node architectures.

There are several attributes which affect this function. Within the NSR resource (see **nsr\_service(5)**) there are the "nsrmmmd polling interval", "nsrmmmd restart interval" and "nsrmmmd control timeout" attributes. These attributes control how often the remote media daemons (see **nsrmmmd(8)**) are polled, how long between restart attempts, and how long to wait for remote requests to complete, respectively.

Within the "NSR device" resource (see **nsr\_device(5)**) the resource's name will accept the "rd=hostname:dev\_path" format when defining a *remote device*. The "hostname" is the hostname of the storage node and "dev\_path" is the device path of the device attached to that host. There are also hidden attributes called "save mount timeout" and "save lockout," which allow a pending save mount request to timeout, and a storage node to be locked out for upcoming save requests.

Within the "NSR client" resource (see **nsr\_client(5)**), there are the "storage nodes" and "clone storage nodes" attributes. The former is used by the server in selecting a storage node when the client is saving data. The latter is used during cloning, to direct cloned data from a volume on the storage node (the node represented by this client resource).

## Install and Configure

In order to install a storage node, choose the client and storage node packages, where given the choice. For those platforms that do not have a choice, the storage node binaries are included in the client package. In addition, install any appropriate device driver packages. If not running in evaluation mode, a storage node enabler must be configured on the server for each node.

As with a client, ensure that the **nsrexecd(8)** daemon is started on the storage node. To define a device on a storage node, from the controlling server define a device with the previously mentioned "rd=" syntax. For a remote jukebox (on a storage node), run **jbconfig(8)** from the node, after adding root@storage\_node to the server's administrator list, (where root is the user running **jbconfig** and storage\_node is the hostname of the storage node). This administrator list entry may be removed after **jbconfig(8)** completes. Note that on a Windows NT storage node, use **jbconfig** rather than **jbconfig(8)**.

In addition to **jbconfig(8)**, when running **scanner(8)** on a storage node, root@storage\_node must be on the administrator list.

When a device is defined (or enabled) on a storage node, the server will attempt to start a media daemon (see **nsrmmd(8)**) on the node. In order for the server to know whether the node is active, it polls the node every "nsrmmd polling interval" minutes. When the server detects a problem with the node's daemon or the node itself, it attempts to restart the daemon every "nsrmmd restart interval" minutes, until either the daemon is restarted or the device is disabled (by setting "enabled" to "no" in the device's "enabled" attribute).

In addition to needing a storage node enabler for each storage node, each remote jukebox will need its own jukebox enabler, as is needed for local jukeboxes.

## Operation

A storage node is assignable for work when it is considered functional by the server (nsrexecd running, device enabled, nsrmmd running, and responding to the server's polls). When a client save starts, the client's "storage nodes" attribute is used to select a storage node. This attribute is a list of storage node hostnames, which are considered in order, for assignment to the request.

The exception to this node assignment approach is when the server's index or bootstrap is being saved - these save sets are always directed to the server's local devices, regardless of the server's "storage nodes" attribute. Hence, the server will always need a local device to backup such data, at a minimum. Note that these save sets can later be cloned to a storage node, as can any save set.

If a storage node is created first (by defining a device on the host), and a client resource for that host is then added, that hostname is added to its "storage nodes" attribute. This addition means the client will back up to its own devices. However, if a client resource already exists, and a device is defined on that host, then one must manually update the client's "storage nodes" attribute, with the hostname for that computer. Since this attribute is an ordered list of hostnames, add its own name as the first entry.

Note that the volume's location field is used to determine the host location of an unmounted volume. The server looks for a device or jukebox name in this field, as would be added when a volume resides in a jukebox. Volumes in a jukebox are considered to be located on the host to which the jukebox is connected. The location field can be used to bind a standalone volume to a particular node by manually setting this field to any device on that node (using the "rd=" syntax).

There are several commands that interact directly with a device, and so must run on a storage node. These include **jbconfig(8)**, **nsrjb()** and **scanner(8)**, in addition to those in the device driver package. When invoking these commands directly, do so on the storage node rather than on the server, and use the server option ("-s server\_hostname", where server\_hostname is the controlling server's hostname).

## Cloning Function

A single clone request may be divided into multiple sub-requests, one for each different source computer (i. e., the host from which save sets will be read). For example, suppose a clone request must read data from volumeA and volumeB, which are located on storage nodes A and B, respectively. Such a request would be divided into two sub-requests, one to read volumeA from storage node A and another to read volumeB from storage node B.

Before the "clone storage nodes" attribute was introduced (see **nsr\_client(5)**) the target location for a clone request (the storage node to which cloned data is written) was determined by the "storage nodes" attribute of the server's client resource. In servers where the "clone storage nodes" attribute is defined, the target node is determined by examining the "clone storage nodes" attribute of the client resource of the storage node, from which the clone volume will be read. If this attribute has no value, the "clone storage nodes" attribute of the server's client resource is consulted. Finally, if this attribute has no value, the "storage nodes" attribute of the server's client resource is used.

## Limitations

Some limitations of this feature are that a server cannot be a storage node of another server, and a storage node can only be controlled by one server.

**See Also:** [mmlocate\(8\)](#), [nsr\\_client\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_service\(5\)](#), [nsrclone\(8\)](#), [nsrexecd\(8\)](#), [nsrmmd\(8\)](#), [nsrmon\(8\)](#), [scanner\(8\)](#)

## nsradmin(8)

### Name

**nsradmin** – Legato Storage Manager administrative program

### Synopsis

```
nsradmin [ -c ] [ -i file ] [ -s server ] [ -p prognum ] [ -v version ] [ query ]
```

```
nsradmin [ -c ] [ -i file ] [ -f resfile ... ] [ -t typefile ] [ query ]
```

### Description

The **nsradmin** command is a command-line based administrative program for the Legato Storage Manager system. Normally **nsradmin** monitors and modifies Legato Storage Manager resources over the network. Commands are entered on standard input, and information is produced on standard output.

If **nsradmin** is started without a query it uses a default query that selects all resources involved in Legato Storage Manager products.

### Options

**-c**

Use the **termcap**(5) and **curses**(3) packages to implement a full-screen display mode, just like the **visual** command described as follows.

**-f *resfile***

Use the Legato Storage Manager resource file *resfile* instead of opening a network connection. This should generally never be used, except when the Legato Storage Manager server is not running. Multiple **-f** and *resfile* arguments can be used to start **nsradmin** with access to more than one file at a time.

**-i *file***

Take input commands from *file* instead of from command line. In this mode, the interactive prompt will not be printed.

**-s *server***

Open a connection to the named Legato Storage Manager server instead of allowing administration of all servers. Useful to limit the number of resources if there are many servers, or to administer when the RAP location service is not working.

**-p** *program*

Use the given RPC program number instead of the standard program number. The standard number is 390109. This option is generally used only for debugging.

**-t** *typefile*

Use the alternate file *typefile* to define RAP types.

**-v** *version*

Bind to the Legato Storage Manager RAP service with the given version number. The default is 2. This option is generally used only for debugging.

*query*

If a query is specified (in the form of an attribute list), the edit operation (see the following description) is performed on the results of the query. See the following section, **Commands**, for more information on how the edit command works.

## Resources

Each Legato Storage Manager resource is made up of a list of named attributes. Each attribute can have zero or more values. The attribute names and values are all represented by printable strings. Uppercase and lowercase is not distinguished on comparisons, and spaces are ignored except inside the names and values.

The format for specifying attributes and attribute lists is:

*attribute ::= name [ : value [ , value ]\* ]*

An attribute is a name optionally followed by a colon, followed by zero or more values, with values separated by commas. A comma at the end of a line continues the line.

*attribute list ::= attribute [ ; attribute ]\**

An attribute list is one or more attributes separated by semicolons. A semicolon at the end of a line continues the line. The list is ended by a newline that is not preceded by a comma or semi-colon.

Here is an example of an attribute list:

```
name: mars;
type: NSR client;
remote access: mars, venus, jupiter;
```

For more information on attributes, attribute lists, and the Legato Storage Manager resource types, see the **resource(5)**, **nsr\_resource(5)**, and **rap(8)** manual pages.

## Commands

At each input prompt, **nsradmin** expects a command name and some optional arguments. Command names can be shortened to the smallest unique string (for example, **p** for **print**). Command arguments are always specified in the form of an attribute list. Most commands operate on a set of resources returned by a *query*. The query is specified as an attribute list which is used to match resources with the following rules:

1. The resource must match all the given attributes.
2. If more than one value is specified the resource can match any one of the values.
3. Values in a query may be in the form of **ed(1)** style regular expressions. A pattern match is attempted against all resources that contain the specified attribute.
4. If an attribute is specified with no value the resource must contain an attribute of that name.

Thus, a query:

```
type:NSR device;  
name:mars, venus;  
test
```

will match all resources that have a **type** attribute with the value **NSR device** and a **name** attribute with a value of either **mars** or **venus**, and an attribute **test** with any value.

If the query has only one name and no values (for example, if there is no semi-colon or colon in it), then the program tries to guess a more reasonable query. If the name is a host name, then the query will select all the resources on the given host. Otherwise, the name will be interpreted as a type name, and all resources of that given type will be selected.

### **bind** [*query*]

Bind to the service that owns the resource described by *query*. If no query is specified, queries are sent to the RAP Resource Directory, and update, create, and delete commands to the service that owns the resource being changed. On failure, the previous service will continue to be used.

### **create attribute list**

Create a resource with the given attributes. One of the attributes must be **type** to specify a Legato Storage Manager type that can be created. The **types** command can be used to find out which types a server supports.

**delete** [*query*]

Delete the resources that match the current query. If a *query* is specified, it becomes the current query.

**edit** [*query*]

Edit the resources that match the current query. If a *query* is specified, it becomes the current query. If the environment variable EDITOR is set, then that editor will be invoked; otherwise **vi**(1) will be started. When the editor exits, **nsradmin** applies update, delete and create operations based on the changes to the resources. Be careful to not edit the resource identifier attribute, and to write the file out before exiting the editor.

**help** [*command*]

Print a message describing a command. If no command name is given a synopsis of all of the commands is printed.

**print** [*query*]

Print the resources that match the current query. If a *query* is specified, it becomes the current query. If the current show list is not empty only the attributes named in the show list will be displayed.

**server** [*servername*]

Bind to the given Legato Storage Manager server name. If no server is specified, the RAP location service will be used. On failure, the previous server will continue to be used.

**show** [*name; ...*]

If a name list (really an attribute list with no values) is specified, add those names to the show list. Only these attributes will be displayed in subsequent **print** commands. If no name list is given the show list is cleared, resulting in all attributes being shown.

**types**

Print a list of all known types.

**update** *attributes*

Update the resources given by the current query to match *attributes*.

**quit**

Exit.

**visual** [*query*]

Enter a full-screen mode using the **curses**(3) package to step through commands in a perhaps more user-friendly manner than the command-line interface. You can get this mode directly using the **-c** command line argument.

**option** [*list*]

This command enables some options to change the display of resources. With no arguments it displays the current options; with a list of options it turns the specified ones on. The options are: **Dynamic** displays all dynamic attributes, even the normally hidden ones. **Hidden** displays all attributes, even the normally hidden ones. **Resource ID** displays the resource identifier on each resource, a number that is used internally to provide sequencing and uniqueness.

**unset** [*list*]

This command turns off the specified option.

**.[query]**

If a *query* is specified, this command will set the current query without printing the results of the query. Otherwise, it will display the current query, show list, server binding, and options.

**? [command]**

Same as the **help** command mentioned previously.

## Examples

```
print type:NSR device
```

Print all resources of type **NSR device** and make this the current query.

```
show type; name
```

Set the show list to only display the attributes **type** and **name**.

```
delete
```

Delete all resources that match the current query.

```
delete type:NSR device; hostname: mars
```

Delete the resource with attributes: **type: NSR device** and **hostname: mars**.

```
edit type:NSR notification
```

Edit all resources of type **NSR notification**.

**See Also:** [ed\(1\)](#), [vi\(1\)](#), [curses\(3\)](#), [nsr\\_resource\(5\)](#), [termcap\(5\)](#), [nsr\(8\)](#), [rap\(8\)](#), [rapd\(8\)](#)

## Diagnostics

The following exit status values are meaningful:

- 0 Interactive mode exited normally.
- 1 There was a usage or other non-query related error.

- 2 When reading input from a file (*-i file*), one or more RAP operations failed. This status is never returned interactively.

## nsralist(8)

### Name

**nsralist** – Legato Storage Manager archive request executor

### Synopsis

**nsralist** **-R** *archive request name*

### Description

The **nsralist** command is used to run an archive request (see **nsr\_archive\_request(5)**). The **nsralist** command is normally run automatically by **nsrd(8)**, as specified by each archive request resource.

The **nsralist** command will set up an RPC connection to **nsrexecd(8)** to run **nsrarchive(8)** on the specified client. If **nsrexecd** is unavailable, **nsralist** will fall back on using the **rcmd(3)** protocol and the client-side **rshd(8)**.

The **nsralist** monitors the execution of the archive command and stores any information in the log of the archive request. The **nsrarchive** command running on the client updates the server with its progress, including whether or not optional verification and cloning operations have completed successfully. See **nsrclone(8)** for more information on cloning.

### Options

**-R** *archive request name*

This option specifies which archive request is supposed to be run.

### Files

**/nsr/tmp/al.request\_name**

A lock file to keep multiple runs of the same archive list from running simultaneously.

**See Also:** [nsrarchive\(8\)](#), [nsrclone\(8\)](#), [nsrexecd\(8\)](#), [nsr\\_archive\\_request\(5\)](#)

## nsrarchive(8)

### Name

**nsrarchive** – Archive files to long term storage with Legato Storage Manager

### Synopsis

```
nsrarchive [ -BinpqvxVy ] [ -b pool ] [ -C clone pool ] [ -f directive filename ]
[ -G remove ] [ -N name ] [ -R name ] [ -s server ] [ -T annotation ] [ -W width ]
[ path ... ]
```

### Description

**nsrarchive** archives files, including directories or entire file systems, to the Legato Storage Manager server (see **nsr**(8)). The progress of an archive can be monitored using the X Window System based **nwadmin**(8) program or the **curses**(3X) based **nsrwatch**(8) program for other terminal types. Use of **nsrarchive** is restricted to users on the administrator and archive users lists.

If no *path* arguments are specified, the current directory will be archived. **nsrarchive** will archive a directory by archiving all the files and subdirectories it contains, but it will not cross mount points, nor will it follow symbolic links.

The directive files (see **nsr**(5)) encountered in each directory will be read by default, and they contain special instructions directing how particular files are to be archived (compressed, skipped, and so on.). These files are named *.nsr*.

Each file in the subdirectory structures specified by the *path* arguments will be encapsulated in a Legato Storage Manager archive stream. This stream of data is sent to a receiving process (see **nsrd**(8)) on the Legato Storage Manager server, which will process the data, adding entries to the media database for the archive save set, with the data finally ending up on some long term storage media (see **nsrmmmd**(8)).

Details about handling media are discussed in **nsrmm**(8) and **nsr\_device**(5).

If the grooming option *remove* is requested, all files and directories archived are removed. If verification is requested, the files will not be removed if the verification failed. Likewise, the files will not be removed if a requested cloning operation fails. The user is prompted for confirmation before the files are removed unless the *-y* option is supplied.

If the user does not supply a **-T** option on the command line, they will be prompted to enter an annotation for the archive.

## Options

### **-b** *pool*

Specify a destination pool for the archive save set. This option overrides the automatic pool selection which normally occurs on the server.

### **-B**

Force archive of all connecting directory information from root (“/”) down to the point of invocation. Note that the connecting directory information is always archived even without this option if client file index is generated.

### **-C** *clone pool*

Generate a clone of this archive save set to the specified *clone pool*.

### **-E**

Estimate the amount of data which will be generated by the archive, then perform the actual archive. Note that the estimate is generated from the inode information, and thus the data is only actually read once.

### **-f** *filename*

The file from which to read default directives (see **nsr(5)**). A *filename* of **-** causes the default directives to be read from standard input.

### **-i**

Ignore directive files as they are encountered in the subdirectory structures being archived.

### **-G** *remove*

Groom the files after they have been archived. If cloning or verification is requested, no grooming is performed until those operations have completed successfully. The user is prompted for removal of top-level directories unless the **y** option is supplied. **nsrarchive** creates a temporary file which contains a list of all files and directories to be removed. The temporary file is placed in */tmp* unless the environment variable **TMPDIR** is set.

### **-n**

No archive. Estimate the amount of data which will be generated by the archive, but do not perform the actual archive.

### **-N** *name*

The symbolic name of this archive save set. By default, the first *path* argument is used as the name.

- v**  
Verbose. Cause the **nsrarchive** program to tell you in great detail what it is doing as it proceeds.
- p**  
Exit with status 0. Used by server to determine if client installed properly.
- q**  
Quiet. Display only summary information and error messages.
- R name**  
This option should only be used by the **nsralist** program, which handles executing archive requests. Updates to the named archive request resource occur when this option is specified.
- s server**  
Specify which computer to use as the Legato Storage Manager server.
- T annotation**  
Archive save sets can be annotated with arbitrary text (up to 1023 characters). This option specifies an annotation for the archive save set being generated.
- V**  
Verify the archive save set after it completes.
- W width**  
The width used when formatting summary information output.
- x**  
Cross mount points.
- y**  
Answer yes to any questions.

**See Also:** [curses\(3X\)](#), [nsr\\_getdate\(3\)](#), [nsr\(5\)](#), [nsr\(8\)](#), [nsr\\_service\(5\)](#), [nsr\\_device\(5\)](#), [nsrmm\(8\)](#), [nsrmmmd\(8\)](#), [nsrd\(8\)](#), [nsrwatch\(8\)](#), [nsrretrieve\(8\)](#), [nwadmin\(8\)](#)

## Diagnostics

Exit codes:

- 0** Normal exit.
- 1** Abnormal exit.

## nsrkap(8)

### Name

**nsrkap** – Update the capabilities of a Legato Storage Manager installation

### Synopsis

**nsrkap** [ **-vn** ] { **-c** | **-u** | **-d** } **enabler-code**

### Description

The **nsrkap** program is primarily used to enable new features on Legato Storage Manager. You can also use **nsrkap** to upgrade or downgrade Legato Storage Manager software features that are currently being used (upgrades and downgrades should be performed carefully; read the following descriptions). Enablers are separate from the Legato Storage Manager software, and are specified by an 18-digit code, usually displayed as 3 groups of 6 digits each. To enable a new feature, the **nsrd**(8) program must be running on the system where the Legato Storage Manager server software is installed. To enable a new feature you must be logged in to the Legato Storage Manager server as administrator or root. The **nsrkap** program is run once for each feature you want to enable by specifying the 18-digit code each time. If no errors occur, no output is printed. You can inspect the enablers currently loaded by viewing the NSR license resources using **nwadmin**(8) or **nsradmin**(8).

### Options

**-c**

Causes **nsrkap** to enable a feature that is not currently installed, using the specified enabler code. You can only load a feature once; an error is returned if you attempt to load the enabler more than once. You can only specify one of the **-c**, **-d**, or **-u** options.

**-d**

Causes **nsrkap** to downgrade an existing Base or Jukebox enabler. After you downgrade the enabler, you cannot return to the previous level enabled on the system. *Do not* use the **-u** option unless instructed to do so by Oracle Technical Support. You must specify one of the **-c**, **-d**, or **-u** options.

**-u**

Causes **nsrkap** to enter an enabler that upgrades an existing Base or Jukebox enabler. After you upgrade the enabler, you cannot return to the previous level enabled on the system. *Do not* use the **-u** option unless instructed to do so by Oracle Technical Support.

**-v**

Causes **nsrkap** to display more verbose output, describing the enabler being loaded. You must specify one of the **-c**, **-d**, or **-u** options.

**-n**

No load. Causes **nsrkap** to inspect the enabler code for validity. When you specify the **-n** option, the enabler code you enter on the command line is inspected and verified, but is not entered into the Legato Storage Manager server's **nsr\_license** resource.

**See Also:** [nwadmin\(8\)](#), [nsradmin\(8\)](#), [nsrd\(8\)](#)

## Diagnostics

**enabler-code is too long**

Enabler codes must be 18 digits in length. The code entered is longer than 18 digits, and is invalid.

**invalid code: xxxxxx-xxxxxx-xxxxxx**

The 18-digit code entered on the command line is invalid. Re-check the enabler-code on your enabler sheet.

**cannot find a jukebox resource to enable**

The code word entered is a jukebox license enabler, but there are no jukebox resources to enable. You need to run **jbconfig(8)** to complete the jukebox installation before running **nsrkap**.

**found a jukebox, but it had more than N slots.**

Jukebox enablers can only enable jukeboxes with at most N physical slots, where N is the type of jukebox enabler. Either the jukebox was installed incorrectly, or you need to obtain a larger jukebox enabler.

**this enabler-code is already assigned**

The enabler-code entered is already loaded onto the system and cannot be used again for an upgrade.

**no appropriate jukeboxes to upgrade**

An upgrade was attempted, but no jukebox resources were found. Only use the **-u** option for jukeboxes when upgrading from one jukebox level to another, not on the initial installation. You also need to run **jbconfig(8)** before running **nsrkap**.

**this enabler-code previously loaded**

The enabler-code entered has been loaded onto the system previously and cannot be used again. You need to purchase a new enabler-code for the upgrade.

**don't know how to upgrade this enabler****don't know how to downgrade this enabler**

The enabler-code entered is not for a base or jukebox enabler. These are the only types of enablers that you can currently upgrade or downgrade.

**base enabler must be loaded before upgrading****base enabler must be loaded before downgrading**

You cannot perform an upgrade or downgrade until a base product has been installed. Install a base enabler, and then perform the upgrade or downgrade.

**cannot find the enabler to upgrade**

A jukebox upgrade was attempted, but the license enabler for the jukebox is not currently loaded. You must use the **-c** option for the initial installation of a jukebox enabler, not the **-u** option.

**RPC error, Program not registered**

The **nsrkap** program requires that the Legato Storage Manager daemons be running. Start your Legato Storage Manager daemons (`cd /; nsrd`) and re-run the **nsrkap** program. If **nsrd** is already running, you have probably reached a resource limit on the server (for example, not enough memory, or no more processes).

**RAP error, user *login name* needs to be of the type:NSR administrator list.**

Your login name is not listed in the administrator's list for the server. You need to be a valid administrator to run **nsrkap**.

**RAP error, ...**

Various other errors can be returned from **nsrd** if the enabler is invalid. For example, if you try to load a base enabler onto a system that already has a base enabler loaded, or if you attempt to load a jukebox enabler before the jukebox has been completely installed. The specific problem will follow the **RAP error** prefix.

## nsrcat(8)

### Name

**nsrcat** – Legato Storage Manager notification redirector for tty devices

### Synopsis

**nsrcat** [ -n ]

### Description

The **nsrcat** command appends a carriage return to all newlines so that Legato Storage Manager notification messages can be redirected to the */dev/console* or */dev/tty* directory on systems with tty drivers that do not append a carriage return to output lines. This command reads text messages from the command line, appends a carriage return to the newline character, and writes the message to the screen.

### Options

**-n**

Indicates that the codeset is to be converted from UTF-8 to the user's native character encoding.

### Example

```
type:          NSR notification;
name:          Log default;
action:        nsrcat > /dev/console;
```

**See Also:** [console\(4\)](#), [tty\(4\)](#), [nsr\\_notification\(5\)](#), [nsr\(5\)](#)

## nsrck(8)

### Name

**nsrck** – Legato Storage Manager index consistency check and repair program

### Synopsis

```
nsrck [ -qM ] | [ -T tempdir ] [ -X [ -x percent ] | -C | -F | -m | -c ]  
[ clientname ... ]
```

### Description

**Nsrck** is used to check the consistency of the Legato Storage Manager on-line index of clients' save records. Normally, **nsrck** is started automatically and synchronously by the **nsrindexd**(8) program when **nsrindexd** starts.

Index consistency checking is finished in up to four phases.

Phase zero determines if any further checking of a client's index is necessary. This phase checks certain internal state of the index database, and if that state is consistent, avoids further passes. Phase zero also reports any suspicious-looking index names (indexes whose names cannot be mapped into network addresses).

Phase one fixes all errors in the database record file, **db**, and, if necessary, rebuilds the b-tree indexes for that database.

In phase two, if the database has been flagged for cross-checking with the media database, the records in the file index are checked against the media database, and any records that do not match existing, browsable save sets are deleted.

Finally, in phase three, if the database requires compression, either due to space freed by the previous phases or due to state flagged by a previous run, the index is compressed.

Index compression is a two or three step process. First, the records of the database are copied to a temporary database, **db.CMP**. When that operation completes, a flag file, **db.SVC**, is created, the old, uncompressed database is removed, and the compressed database is renamed to **db**. Finally, the **db.SVC** file is removed. If there is not enough room on the file system containing the **db** file to also include the temporary database, **nsrck** will create a temporary file on another writable file system, and stores a pointer to this file in a file named **db.PTR**. In this case, an extra copy of the data is required, since the uncompressed database must first be

removed before the data can be copied back to the correct place. After all of these steps are completed, the **db.PTR** file is removed.

The **nsrck** program is restartable at any time during its execution. Therefore, it can survive system crashes or exhaustion of resources without losing data.

## Options

**-C**

Force index compression on the named clients, or all clients if none are specified. Other phases of checking are only performed if an error in a database is detected.

**-c**

Create indexes databases for the named clients, or all clients if none are specified. Do not check the index databases if they already exist.

**-F**

Force a check on any given client names. If no names are given, forced checks are performed for all client indexes. This option forces all phases of index checking. For backward compatibility, the **-F** option implies index compression, and may be used to force the index to be compressed. This option is typically only necessary when the *browse policy* (see **nsr\_policy**(5)) is reduced, for example, changing the browse policy from 1 year to 6 months.

**-T tempdir**

Specify the directory to use to hold the temporary database during compression if there is not enough room in the file system containing the **db** file. If this option is used and there is insufficient space in the specified directory, **nsrck** will fail. This argument is ignored if there is sufficient space in the file system containing the **db** file.

**-M**

Master mode (not advised for manual operation). This advises **nsrck** that it is being run by **nsrd**(8) or another Legato Storage Manager daemon and should log messages with timestamps, and perform any other behavior expected by **nsrd**.

**-m**

Forces **nsrck** to check and rebuild the media database b-tree indexes, instead of checking a client's online file index.

**-q**

Quiet mode. All advisory messages are suppressed.

**-X**

Do not check the index databases (unless an error in phase zero occurs), rather cross-check the save set ids in the index records with save sets found in the media database. Records that have no corresponding media save sets are discarded. If any clients are listed, the cross-check is limited to those indexes.

**-x percent**

After a database has been cross-checked, if the database uses less than *percent* of the UNIX file, then the database is compressed to release unused pages back to the file system. The percent default is 30.

## Files

**/nsr/tmp/.nsrck**

**Nsrck** locks this file thereby insuring that only one copy of the command thrashes the Legato Storage Manager server.

**/nsr/index/clientname**

**/nsr/index/clientname/db**

**/nsr/index/clientname/db.CMP**

**/nsr/index/clientname/db.PTR**

**/nsr/index/clientname/db.SCV**

**/nsr/index/clientname/db.RCV**

**/anyfilesystem/nsrckXXXXXX**

Temporary database created while compressing a db file if there is not enough room to do compression in **/nsr/index/clientname/db.CMP**.

**See Also:** [nsr\\_layout\(5\)](#), [nsr\\_policy\(5\)](#), [hosts\(5\)](#), [nsrd\(8\)](#), [nsrindexd\(8\)](#), [nsrmmdbd\(8\)](#), [nsrim\(8\)](#), [savegrp\(8\)](#)

## Diagnostics

**checking index for *clientname***

Informative message that the files associated with the named client are being inspected.

**more space needed to compress *clientname* index, size required**

The program cannot find enough disk space to hold the temporary file **db.CMP**. The operator should free some disk space on any local file system and retry the command. The **df(8)** command may be used to see how much free space is available on any file system.

**cannot lock flag file for *clientname*: reason**

The flag file signifying the end of the first part of index compression is already in

use by another instance of this program, or by the **nsrindexd** daemon. Since disaster will ensue if two processes access the same index at the same time, the program will refuse to act on the named file.

**WARNING no valid savetimes - cross-check not performed for *clientname***

During a cross-check, no save sets were found for this client. Since this situation can occur during disaster recovery, **nsrck** avoids deleting the entire contents client index and instead does nothing.

**cross-checking index for *clientname***

Displayed when the **-X** option is in effect.

**rolling forward index compression for *clientname***

After a restart, if index compression completed its first copy, the compression is rolled forward.

**compressing index for *clientname***

Displayed when the **-x** or **-C** option has taken effect.

**completed checking *count* clients**

Displayed as the program finishes, provided some form of checking was accomplished.

## nsrclone(8)

### Name

**nsrclone** – Legato Storage Manager save set cloning command

### Synopsis

**nsrclone** [ **-v** ] [ **-s** *server* ] [ **-b** *pool* ] { **-f** *file* | *volname...* }

**nsrclone** [ **-v** ] [ **-s** *server* ] [ **-b** *pool* ] **-S** { **-f** *file* | *ssid...* }

**nsrclone** [ **-v** ] [ **-s** *server* ] [ **-b** *pool* ] **-V** { **-f** *file* | *valid...* }

**and on HSM enabled server:**

**nsrclone** [ **-v** ] [ **-s** *server* ] [ **-b** *pool* ] **-c** *client* **-N** *saveset*

### Description

The **nsrclone** program is used to make new copies of existing save sets. These copies are indistinguishable from the original, except for the volume(s) storing the copies. The copies are placed on different media volumes, allowing for higher reliability than a single copy provides. The copies may be made onto any kind of media (for example, save sets on an 8mm tape may be copied to a set of optical disks). However, all media that will be used as the destination of an **nsrclone** operation must be in a *clone pool*. See **nsr\_pool(8)** for a description of the various pool types.

Although the command line parameters allow you to specify volume names or volume identifiers, **nsrclone** always copies complete save sets. Save sets that are only partially contained on a specified volume will be completely copied, so volumes may be requested during the cloning operation in addition to those specified on the command line. Note that **nsrclone** does not perform simple *volume duplication*, but rather, copies full save sets to a set of destination volumes in a given pool. If the first destination volume chosen cannot hold all of the save sets to be copied, another volume will be chosen. This allows one to use different kinds of media for each copy, and allows for variable sized volumes, such as tapes.

If you use the **-c** and **-N** options together, **nsrclone** will create a *super-full* clone for the given client save set. A super-full clone is a feature that is supported only with HSM (most useful on HSM enabled servers). It automatically creates a clone of the most recent complete full backup of the named client and save set, along with any HSM migration save sets referenced to by the backup. Super-full clones should be

cloned to a volume from a pool of type *Migration Clone*. If no migration save sets are referenced by the most recent backup, only the full save set is cloned.

The **nsrclone** program, in conjunction with **nsrmmmd(8)**, guarantees that each save set will have at most one clone on a given volume. When you specify a volume name or identifier, the copy of the save sets on that volume are used as the source. When save sets are specified explicitly, those with existing multiple copies are automatically chosen (copies of save sets that exist on volumes in a jukebox are chosen over those that require operator intervention). You can also specify which copy (clone) of a save set to use as the source; see the **-S** option, described as follows.

Cloning between storage nodes is accomplished by an **nsrmmmd(8)** on the source node reading from a volume, and another **nsrmmmd(8)** on the target node writing to a volume. The source node is determined by the location of a source volume, which is given by where the volume is currently mounted or by its "location" field if unmounted (see **mmlocate(8)**). The target node of a clone is determined by either the "clone storage nodes" attribute of the storage node's client resource, that of the server's, or the "storage nodes" attribute of the server. See **nsr\_storage\_node(5)** and **nsr\_client(5)** for additional detail on how these attributes are used and on other storage node information.

## Options

### **-b** *pool*

Specifies the media pool to which the destination clones should be sent. The pool may be any pool currently registered with **nsrd(8)** that has its status set to *clone*. The possible values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin(8)**. If you omit this option, the cloned save sets are automatically sent to the *Default Clone* pool.

### **-c** *client*

Specifies a client whose save sets should be considered in the creation of a super-full clone. This option must always be specified in conjunction with the **-N** option.

### **-f** *file*

Instructs **nsrclone** to read the volume names, volume identifiers or save set identifiers from the file specified, instead of listing them on the command line. The values must be listed one on each line in the input file. The *file* may be "-", in which case the values are read from standard input.

### **-N** *saveset*

Specifies which save set name is used to create a super-full clone. This option must always be specified in conjunction with the **-c** option.

**-s** *server*

Specifies a Legato Storage Manager server to migrate save sets from. See **nsr(8)** for a description of server selection. The default is the current system.

**-v**

Enable verbose operation. In this mode, additional messages are displayed about the operation of **nsrclone**, such as save sets that cross volumes, or save set series expansions.

**-S**

Causes **nsrclone** to treat subsequent command line parameters as save set identifiers, not volume names. Save set identifiers are unsigned numbers. You can find out the save set identifier of a save set using the **mminfo -v** command (see **mminfo(8)**). The **-S** option is useful when you want to copy individual save sets from a volume or all save sets matching an **mminfo** query (see the following examples). The save set identifiers may also specify exactly which copy of a save set with multiple copies to use as the source. To specify exact copies, use the *ssid/cloneid* format for each save set identifier. In this case, the *ssid* and the *cloneid* are unsigned numbers, separated by a single slash (/). You can find out the *cloneid* for a particular copy by using the **mminfo -S** report, or a custom report.

**-V**

Causes **nsrclone** to treat subsequent command line parameters as volume identifiers, not volume names. Volume identifiers can be found using the **mminfo -mv** report, for example. This option cannot be used in conjunction with the **-S** option.

## Examples

Copy all save sets on the volume **mars.001** to a volume in the **Offsite Clone** pool:  
**nsrclone -b 'Offsite Clone' mars.001**

Copy all complete save sets created during the previous weekend (recall that **nsr\_getdate(3)** dates without time-of-day match midnight at the beginning of that day). Only complete save sets can be copied by **nsrclone(8)**:

```
nsrclone -S 'mminfo -r ssid \  
-q '!incomplete,savetime>last saturday,savetime<last monday''
```

Copy a specific clone of a specific save set:

```
nsrclone -S 1538800517/770700786
```

**See Also:**    [nsr\\_getdate\(3\)](#), [nsr\\_pool\(5\)](#), [nsr\\_storage\\_node\(5\)](#),  
[mminfo\(8\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nsrmmd\(8\)](#), [nwadmin\(8\)](#)

## Diagnostics

The exit status is zero if all of the requested save sets were cloned successfully, nonzero otherwise.

Several messages are printed signalling that **nsrd(8)** is unavailable for cloning data; these are self-explanatory. You may also see a message from the following list.

### **adding save set series which includes *ssid***

If running in verbose mode, this message is printed when **nsrclone** notices that a requested save set is continued, requiring the entire series to be cloned (even if only part of the series was specified in the command line parameters).

### **Cannot contact media database**

The media database (and most likely other Legato Storage Manager services as well) on the named server is not answering queries. The server may need to be started, or if it was just started, it must finish its startup checks before answering queries.

### **cannot clone save set *number*, series is corrupt**

The given save set is part of a save set series (used for saving very large files or file systems), but not all of the save sets in the series were found in the media database. This can happen if, for example, you relabel a tape that contains part of a save set series.

### **cannot clone backup and archive data together**

Archive and backup data is fundamentally different and cannot be cloned to the same pool. You need to run **nsrclone** twice, once to clone the backup save sets and once more for the archive save sets.

### **cannot open nsrclone session with *server***

This message is printed when the server does not accept clone sessions.

### **cloning not supported; upgrade required**

Another enabler is required to use this feature.

### **cloning requires at least 2 devices**

Cloning requires at least one read/write device and one read-only or read/write device, since data is copied from one volume directly to another.

### ***server* does not support cloning**

The named server is not capable of cloning.

### **error, no valid clones of *ssid number***

The listed save set exists, but cannot be cloned because there are no complete copies

of the save set. The save set was either aborted or is in progress. Only complete save sets can be copied.

**error, user *username* needs to be on administrator list**

**error, user *username* needs to be on archive users list**

Only Legato Storage Manager administrators are allowed to make clones of backup save sets. Legato Storage Manager administrators are listed in the NSR server resource, see **nsr\_service(5)** for more information. For servers with archive capability, users listed in the NSR archive client's user list are allowed to clone archive save sets.

**no complete, full backup of *client:saveset***

During an attempt to create a super-full clone, **nsrclone** could not find a complete, full backup of the requested save set.

**no complete save sets to clone**

Only complete save sets can be copied, and no complete save sets were found matching the requested command line parameters.

***number* is not a valid save set**

The given save set identifier is not valid. Two forms are understood: simple save set identifiers and those with a cloneid specified. Simple save sets are unsigned numbers. The save set with the cloneid form is specified as two unsigned numbers separated by a single slash (/).

**pool is not a cloning pool**

The pool specified with the **-b *pool*** option is not a clone pool. You must always use a pool with a type of "Backup Clone" or "Archive Clone" for the **-b** option.

**save set *number* does not exist**

The given save set (from a **-S** save set list) does not exist. Verify your save set identifiers using **mminfo(8)**.

**save set *number* crosses volumes; requesting additional volumes**

This message is printed in verbose mode when volume names or IDs were specified, but the given save set is only partially resident on the listed volumes. Since only complete save sets can be cloned, **nsrclone** automatically requests additional volumes.

**save set clone *number/cloneid* does not exist**

A specific clone of a save set was specified, but that save set has no clones with that clone identifier. Verify your save set identifiers using **mminfo(8)**.

**volume *name-or-number* does not exist**

The given volume (either a volume name or a volume id specified in the **-V** option) does not exist in the media database.

**waiting 30 seconds then retrying**

A temporary error occurred and **nsrclone** will automatically retry the request until the condition is cleared. For example, an error will occur if all devices are busy saving or recovering and **nsrclone** must wait for these devices become available.

**Warning: No candidate migration save sets of *client:saveset***

If you are using **nsrclone** to create a super-full clone, and the most recent full backup of the named client and save set does not refer to any migration save sets, this warning is printed as nsrclone begins cloning the full backup.

## nsrd(8)

### Name

**nsrd** – Daemon providing the Legato Storage Manager service

### Synopsis

**nsrd**

### Description

The **nsrd** daemon provides an RPC-based save and recover service. This service allows users to save, query for, and recover their files across a network. The RPC program number provided by **nsrd** is 390103.

Normally **nsrd** is invoked from a startup shell script (for example *rc.local*, *rc.boot*) at boot-time, and should never need to be started directly by a user. After it is started, **nsrd** starts up the other daemons it must provide the Legato Storage Manager service.

The **nsrd** command must be run on a computer with appropriate resources. These resources include such things as devices (for example, tape drives) which are under the control of the media multiplexor software, **nsrmmd**(8), and sufficient disk space for the index daemons, **nsrindexd**(8), and **nsrmmdbd**(8), to maintain the index of user files that have been saved and which volumes contain which files.

Each time a backup, recover or various other sessions begin, **nsrd** starts another program, **ansrd**, to process the requested session. The **ansrd** program is called an *agent*. The agent is in charge of monitoring that backup, recover, or other session, and automatically exits when a session completes. Using **ps**(1) or another process monitoring tool, one can inspect the subsequent parameters of **ansrd** to see what kind of session it is monitoring. If necessary, agents can be forcibly terminated to stop a backup or recover session. Agents cannot be run directly; they can only be started by **nsrd**.

## Files

### **/nsr/logs/daemon.log**

The file to which **nsrd** and other Legato Storage Manager daemons send information about various error conditions that cannot otherwise be logged using the Legato Storage Manager event mechanism.

### **/nsr/res/nsr.res**

Attributes describing the Legato Storage Manager service and its resources (See **nsr\_service(5)**).

### **/nsr/res/nsrjb.res**

Attributes describing the jukebox resources of the Legato Storage Manager service.

**See Also:** [nsr\(8\)](#), [nsr\\_service\(5\)](#), [nsrmmd\(8\)](#), [nsrmmdbd\(8\)](#), [nsrindexd\(8\)](#), [ps\(1\)](#), [rc\(8\)](#)

## nsrexec(8)

### Name

**nsrexec** – Remotely run Legato Storage Manager commands on Legato Storage Manager clients

### Synopsis

```
nsrexec [ -a auth ] [ -vR ] [ -T hangseconds ] [ -c client ] [ -f file / - ] [ -- ... ]
```

### Description

The **nsrexec** command is run only by other Legato Storage Manager commands. It is used to remotely run commands on Legato Storage Manager clients running **nsrexecd** and monitor the progress of those commands.

### Options

- a**  
The authentication type to use.
- T**  
The inactivity timeout.
- c**  
The client on which to run the command.
- R**  
A file system probe is being performed.
- v**  
Verbose output has been requested.
- f**  
Data is read from a file or the standard input.
- (double dash)  
Information after the double dash is used for descriptive purposes only.

**See Also:** [nsr\(5\)](#), [nsr\(8\)](#), [nsrexecd\(8\)](#), [savegrp\(8\)](#)

## nsrexecd(8)

### Name

**nsrexecd** – Legato Storage Manager client execution service

### Synopsis

**nsrexecd** [ **-s** *server* [ **-s** *server ...* ] ] [ **-f** *serverfile* ] [ **-p** *savepath* ] [ **-i** ]

### Description

**nsrexecd** is used by Legato Storage Manager servers to perform automatic operations on Legato Storage Manager clients. It is currently used by **savegrp**(8) to start saves, pre-migrations and storage node functions on Legato Storage Manager client computers. When migration is enabled on a Legato Storage Manager client, **nsrexecd** monitors disk usage, starts migration runs using **nsrmig**(8), and also manages file recall using **nsrib**(8). When storage node functions are in use, **nsrexecd** starts **nsrmmd**(8) daemons and **nsrjb**() commands on the host, and responds to polling requests from the server. See **nsr\_storage\_node**(5) for additional detail on storage nodes. The **nsrexecd** service is normally started at startup time on each Legato Storage Manager client computer. Since Legato Storage Manager servers are usually expected to be clients of themselves, **nsrexecd** runs on all Legato Storage Manager servers as well.

The **nsrexecd** service exports an RPC-based service to remotely perform Legato Storage Manager operations. All requests must be authenticated, and can optionally be restricted to specific Legato Storage Manager servers. Only **save** requests (for example, **save**(8) or **savefs**(8)), migration requests (for example, **nsrpmig**(8) or **nsrmig**(8)), and storage node requests are allowed.

When command execution is requested, **nsrexecd** first verifies that the request is authenticated, and that it comes from a valid Legato Storage Manager server. Next, **nsrexecd** verifies that the command is a save or migration command (for example, **save**(8) or **nsrpmig**(8)). It then executes the specified command from the Legato Storage Manager binary directory. This directory is normally determined by the location of the **nsrexecd** executable, but can be specified on the command line.

## Options

**-i**

As part of the Legato Storage Manager server authentication, the server's network address is mapped to a name. The name is then reverse-mapped to a network address. The server is authenticated if and only if the original network address matches the reverse-mapped address. The **-i** flag skips the address comparison thereby allowing workarounds to misconfigured or misfeatured (that is, Solaris) naming systems. This option should be used with care since it may allow the Legato Storage Manager client to send its data to an unauthorized computer.

**-f serverfile**

Specifies a file containing a list of Legato Storage Manager servers which can start saves. This file should list one server name on each line. If no **-f** or **-s** options are specified, **nsrexecd** will look for a default file in this same format (or Mac preferences on the Mac client). The location of this default file is listed in the following **Files** section.

**-p savepath**

Tells **nsrexecd** to look for save commands in the *savepath* directory, rather than the default (the directory in which **nsrexecd** exists).

**-s server**

Only allow save requests to be initiated by the given Legato Storage Manager server. Multiple **-s** options may be given to allow access by several Legato Storage Manager servers. If a Legato Storage Manager server has multiple network interfaces, it is often best to list the hostname corresponding to each network interface, to avoid failed saves. See also the **-f** option as follows for more information.

## Files

**/nsr/res/nsrla.res**

Attributes describing the Legato Storage Manager **nsrexecd** service and its resources (see **nsr\_service(5)**).

**/nsr/res/servers**

The file containing the default list of servers that can back up or migrate the Legato Storage Manager client.

**See Also:** [nsr\\_service\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [nsrib\(8\)](#), [nsrmig\(8\)](#), [nsrpmig\(8\)](#), [save\(8\)](#), [savefs\(8\)](#), [savegrp\(8\)](#)

## nsrfile(8)

### Name

**nsrfile** – Legato Storage Manager module to aid in writing shell script ASMs

### Synopsis

**nsrfile -s** [-N *asmname*] [-C *command*] <ASM args> file...

**nsrfile -r** [-F] [-N *asmname*] [-C *command*] <ASM args> file...

### Description

The **nsrfile** command is an external ASM (Application Specific Module) that makes it easy to implement shell script based ASMs for saving and recovering databases, raw devices, and other site-specific special data.

It is intended that **nsrfile** be invoked by a shell script which is in turn invoked by **uasm(8)** during **save(8)** or **recover(8)** operations.

In save mode, **nsrfile** creates a Legato Storage Manager protocol save record on standard-out for each file given on the command line. A command (see **-C** option as follows) is executed for each filename and the data produced by that command is converted into Legato Storage Manager protocol save record data and emitted on standard-out. In this way, **nsrfile** can be used to create a save record from the data produced by any UNIX command. Save mode looks like:

```
save-command | nsrfile -s ==> save record
```

In recover mode, **nsrfile** reads a Legato Storage Manager protocol save stream from the incoming information and runs a command for each save record in the stream. The save data for each save record is written to the standard input of the running command. Recover mode looks like:

```
save record ==> nsrfile -r | recover-command
```

### Options

See **uasm(8)** for a general description of ASMs and the <ASM args>.

**-s**

Save mode.

**-r**

Recover mode.

**-F**

Don't create the target file. This flag is only valid in recover mode (with the **-r** flag). This flag is used when the data being recovered contains the information necessary to create the target file (for example, tar or cpio data).

**-C command**

Execute the given command for each file in either save or recover mode. In save mode (**-s** flag), the **command** is executed once for each file given on the command line. The stdout of the command is piped to the stdin of **nsrfile** which produces a Legato Storage Manager save record on stdout. In recover mode (**-r** flag), the command is executed once for each file in the recover data stream. The stdout of **nsrfile** is connect through a pipe to the stdin of the **command**. The **command** can include an argument **%** which will be replaced with the current path name on each invocation of **command**.

**-N asmname**

Pretend to be the ASM called **asmname**. This option allows **nsrfile** to take the name of the shell script ASM that invoked it so that warnings and error messages are prepended with **asmname** instead of **nsrfile**.

**file ...**

List of files and directories for command, see option **-C**.

## Implementing ASM Shell Scripts

Several support procedures are necessary to implement an ASM shell script. The **hideasm(8)** command is implemented as a shell script ASM and it can be used as the source of these shell procedures.

**asm\_echo**

Works like the shell **echo** command, but it writes to stderr instead of stdout. This procedure should be used instead of **echo** throughout the ASM script because stdout is used for the save record output in save mode and to pipe to a command in recover mode.

**asm\_error**

Like **asm\_echo** mentioned previously, but it prepends the name of the ASM script to the message.

**asm\_setup**

Parses the ASM script arguments and breaks them into several shell variables. This procedure expects as arguments all of the arguments to the ASM shell script.

The shell variables defined by **asm\_setup** are:

**Nsrfile\_args**

The arguments that **nsrfile** should be invoked with. These arguments are derived from the shell script arguments.

**Files**

The list of files and directories to operate on.

**Mode**

The ASM mode. The value of this variable can be **SAVE**, **RECOVER**, or **COUNT**.

**Myname**

The name that the ASM shell script was invoked with. The directory path is stripped off.

**Exec\_path**

The directory path name that should be used to explicitly call **nsrfile** or other ASM executables.

## Example

To implement a shell script ASM that uses **tar(1)** to back up and recover files and directories the following shell script (plus the shell procedures described previously) would be used:

```
asm_setup $0 $*
if [ $Mode = SAVE ]; then
    $Exec_path/nsrfile -C "tar cf - %" $Nsrfile_args $Files
elif [ $Mode = RECOVER ]; then
    $Exec_path/nsrfile -F -C "tar xBf -" $Nsrfile_args $Files
else # $Mode = COUNT
    $Exec_path/nsrfile $Nsrfile_args $Files
fi
```

If this **tarasm** script were executed with the command line:

**tarasm -s foo**

the Mode variable would be set to SAVE and **nsrfile** would run as though the following command line had been typed:

**tar cf - foo | nsrfile -N tarasm -s foo**

**See Also:** [nsr\(5\)](#), [save\(8\)](#), [recover\(8\)](#), [uasm\(8\)](#)

## nsrhmck(8)

### Name

**nsrhmck** – Check and correct consistency for an HSM-managed file system

### Synopsis

**nsrhmck** [ *-cdfMv* ] [ *-s server* ] *path*

### Description

The **nsrhmck** command checks and correct consistency between HSM migrated files stored in Legato Storage Manager and disk files. There are four situations handled by **nsrhmck**.

The first situation occurs when the stub for a migrated file is renamed. This is known as the rename case. In this situation, the stub with the original filename no longer exists. The corrective action taken by **nsrhmck** in this instance is to update the HSM file index to reflect the new name.

The second situation occurs when a symbolic link is created that points to the same name in the Legato Storage Manager IB namespace as another symbolic link. This is known as the duplicated link case. The corrective action for a duplicated link is to replace the duplicate with a symbolic link that points to the original symbolic link, and not directly into the Legato Storage Manager IB namespace.

The third situation occurs when the stub pointing to a migrated file has been deleted. This is known as the possible delete case. The term "possible" is used here, because the stub may reappear at some later point in time, possibly if a recover of the stub is performed using Legato Storage Manager. The corrective action for a possible deletion is to mark the index entry for the migrated file as possibly deleted. Note that if a file marked as possibly deleted is detected on disk before the index entry is later deleted, the index entry will be unmarked as a possible deletion.

The fourth situation handled by **nsrhmck** is an index entry that is marked as a possible deletion that has passed the 60 day expiration time. This is known as the expired delete case. The corrective action for this case is to remove the expired entries from the HSM file index. Before an entry is deleted from the HSM file index, a final check is made to make sure the file doesn't exist on disk.

A path must be specified on the command line when **nsrhmck** is run. Only files and index entries that fall under the path specified will be examined for consistency.

## Options

**-c**

Walk the HSM file index, and delete index entries that are marked as possibly deleted and that have passed the 60 day expiration time.

**-d**

Walk the HSM file index, and mark any possible deletions that are detected.

**-f**

Walk the file system on disk, and search for duplicated links and renames.

**-M**

Master mode (not advised for manual operation). This advises **nsrhmck** that it is being run by **nsrexecd(8)** or another Legato Storage Manager daemon and should log messages with timestamps, and perform any other behavior expected by **nsrexecd**.

**-n**

Don't correct inconsistencies found, but just report them.

**-s server**

Use **server** as the Legato Storage Manager server.

**-v**

Increment the verbosity level. This flag may be specified up to three times to achieve the maximum verbosity. Note that verbose mode can produce an extremely large quantity of output, and is not recommended for use in most situations.

**See Also:** [nsr\(5\)](#), [nsr\(8\)](#), [nsr\\_migration\(5\)](#), [nsrexecd\(8\)](#), [nsrmig\(8\)](#), [nsrpmig\(8\)](#)

## Diagnostics

Exit codes:

**0** Normal exit.

**1** Abnormal exit.

## nsrib(8)

### Name

**nsrib** – Legato Storage Manager index browser daemon

### Synopsis

```
nsrib [ -s server ] [ -t timeout ] [ -v ] [ -M ] [ -i # ] [ -C # ] [ -D # ] [ -R # ] [ -T rdir ]  
[ dir ]
```

```
nsriba [ -s server ] [ -c client ] [ -p path ] [ -v ] [ -t browse_date ] [ -I index_type ]  
[ -N session_name ] [ -i # ] [ -C # ] [ -D # ] [ -R # ] [ -T rdir ] [ dir ]
```

### Description

The **nsrib** (index browser) and **nsriba** (index browser agent) daemons provide a convenient NFS interface in which to view Legato Storage Manager indexes. Using **nsrib** is the preferred method as it will launch and manage the appropriate **nsriba** processes as needed. The **nsriba** daemon gives you an NFS file system view of a particular Legato Storage Manager client's index as of a given time and can be used directly for situations where the flexibility provided by **nsrib** is not required. The **nsrib** and **nsriba** daemons appear to be an NFS server to the local kernel in a manner similar to **automount**(8).

The **nsrib** daemon will interpret names referenced in *dir* without an '@' as a Legato Storage Manager client index to browse. You can also construct names of the form *client@date* to browse a particular index as of a particular time. You can also use the name of the form *@date* to browse the index for the local computer. *date* is interpreted as a **nsr\_getdate**(3) style string after replacing any underscores ('\_') with a space (' ') and all dashes ('-') with a slash ('/'). When **nsrib** gets such a name request, it will launch and manage appropriate **nsriba** process on a mount point that it builds in *dir* automatically. If the **nsriba** file system is not accessed within an appropriate interval, **nsrib** will attempt an unmount of the **nsriba** file system. If successful, the symbolic link and mount directory created in *dir* is removed.

Below the *dir/client@date* directory for **nsrib** (or within the *dir* directory for **nsriba**), a read-only file system consisting of the entire Legato Storage Manager index for the specified client can be seen. If the local computer does not have Legato Storage Manager **recover access** rights for the specified client (see **nsr\_client**(5)), or if there are no entries in the Legato Storage Manager index for the specified client at the appropriate time, then the directory will be empty (**nsrib**) or the command will fail

(**nsriba**). The files and directories within the **nsriba** file system will appear as normal UNIX files just as in **recover(8)** except that the access time (atime) of all files will be the "save time" of the file, not the access time of the file as stored in the index. Thus running **ls -lu** within an **nsriba** directory will show all the file save times. If an file within an **nsriba** file system is read, then **nsriba** will either recover the file and then return the resultant file as needed or return an NFSERR\_OPNOTSUPP ("Operation not supported") error. The actual behavior is dependent on the **-R** and **-C** flags described as follows and whether the file appears to be currently "on-line" to Legato Storage Manager. If a file is to be recovered, the actual operation may take a long time depending primarily at the speed and location of the underlying media that will be needed to recover the file. A separate process is used to do the actual file recovery so that the **nsriba** process can still respond to new NFS operations.

Within **nsriba** file systems, hidden directories can be referenced for each file or directory to give information similar to the **recover(8)** **version** command. These hidden directories are named *file.V*. Since these hidden directories names are never returned for things like **readdir(2)**, programs like **find(1)** that traverse the file system will never see these directories. The files/directories within these hidden directories are built up using the Legato Storage Manager file location information. The hidden directories for directories can either be named as ".V" within the directory or as *dirname.V* from above the directory. But when using **nsrib**, you can only use *dir/client@date/.V* to see all the versions for "/". Files within the hidden directories can be read (recovered) as any other file within the **nsriba** file system.

**nsrib** and **nsriba** must not be terminated with the SIGKILL signal (kill -9). Without an opportunity to unmount itself and clean up properly, the **nsrib** and **nsriba** mount points will appear to the kernel as a non-responding NFS server. The recommended way to terminate an **nsrib** or **nsriba** process is to send a SIGTERM (kill -15) signal to the daemon. When **nsriba** receives a SIGTERM signal, it will attempt to unmount itself and will exit if the unmount is successful (the file system is not currently busy). When **nsrib** receives a SIGTERM signal, it will try to signal any child **nsriba** processes to exit that it started. If all child **nsriba** processes exit, **nsrib** will then attempt to unmount *dir* itself and will exit if the unmount is successful (the file system is not currently busy).

## Options

Common **nsrib** and **nsriba** options:

**-s** *server*

Legato Storage Manager server to use.

**-i #**

Specifies "in place" mode if the corresponding file in the system is a symlink whose target string has the filename at the end.

0 - never to "in place" recovers

1 - do "in place" recovers only for exact matches with names of "file@date"

2 - do "in place" recovers on any matching symlink target

Default value is 1.

**-v**

Run in verbose mode. This should only be used for debugging purposes.

**-C #**

Sets an upper limit on the number of concurrent file recovers that can be going at once. A value of 0 will disable all recovers (independent of the **-R** value described as follows). Default value is 2.

**-D #**

Specifies the debug level for messages. Using a number from 1 - 3 to get various (reasonable) levels of output. When running in a debugging mode, **nsrib** will not automatically run itself in the background. Default value is 0.

**-R #**

Specifies recover mode on read.

0 - never recover the file on NFS read.

1 - recover the file on NFS read if "on-line".

2 - always attempt a recovery of a file on NFS read.

Default value is 2.

**-T rdir**

Temporary directory to used to cache recovered files. Default value is `"/usr/tmp/nsrib/Rtmp.client"`.

The **-i**, **-s**, **-v**, **-C**, **-D**, **-R**, and **-T** options to **nsrib** are passed through to each **nsriba** program started.

The following options apply only to **nsrib**:

**-t timeout**

Time in minutes to attempt unmounts of **nsriba** browsing directories. Default is 30 minutes.

**-M**

The **nsrib** is being monitored by another process (such as **nsrexecd(8)**), and should not run in the background.

The following options apply only to **nsriba**:

**-c** *client*

Legato Storage Manager client index name to browse.

**-p** *path*

Legato Storage Manager index path to browse.

**-t** *browse\_date*

A **nsr\_getdate(3)** string giving the "browse as of" time. Default value is **now**.

**-I** *index\_type*

Type of index that is being browsed. The default is a backup index.

**-N** *session\_name*

Name to used to generate the Legato Storage Manager session name. Default value is the mount directory *dir*.

## Examples

These examples assume that **nsrib** has been started on the **/ib** directory.

### *Finding files*

To find all versions of a file named *foo* that were owned by *user* last week. Note that when using **find(1)**, you should **cd(2)** to the directory first to avoid using the symbolic link instead of the resultant directory.

```
cd /ib/@last_week; find . -name foo -user user -ls
```

### *Seeing saved versions*

To see all the saved versions of **/var/adm/messages** for a Legato Storage Manager client *clientname*. Note that by using the **-u** flag to **ls(1)**, the file save times will be displayed in the **ls** date field.

```
ls -lu /ib/clientname/var/adm/messages.V
```

### *Recovering files*

To recover **/etc/fstab** as of yesterday into the **/tmp** directory.

```
cp /ib/@yesterday/etc/fstab /tmp/fstab
```

## Files

### **/etc/mtab**

Updated on SunOS 4.1.x as **nsrib** and **nsriba** processes are mounted and umounted.

### **/etc/mnttab**

Updated on Solaris 2.x as **nsrib** and **nsriba** processes are mounted and umounted.

### **/ib**

Directory on which **nsrib** will mount itself.

### **/usr/tmp/nsrib**

Default file cache directory tree.

**See Also:** [mount\(2V\)](#), [umount\(2V\)](#), [signal\(3\)](#), [nsr\\_getdate\(3\)](#), [nsr\(5\)](#), [nsr\\_client\(5\)](#), [automount\(8\)](#), [nsrindexd\(8\)](#), [nsrexecd\(8\)](#), [recover\(8\)](#)

## Bugs

The **pwd(1)** command fails from within a hidden ".V" directory.

The file system statistics returned to programs like **df(1)** aren't terribly useful.

An unreliable heuristic is used to determine if a file is currently "on-line" for recovery if the Legato Storage Manager server version is 3.x or earlier. In particular, if a volume on a pre-4.0 Legato Storage Manager server is marked to be at some location using the **mmlocate(8)** command, **nsriba** always believes that the volume is "on-line".

## nsrim(8)

### Name

**nsrim** – Legato Storage Manager index management program

### Synopsis

**nsrim** [ **-b** *browse* ] [ **-c** *client* ] [ **-N** *save set* ] [ **-r** *retention* ] [ **-x** *percent* ] [ **-lnqvMX** ]

### Description

The **nsrim** program is used to manage Legato Storage Manager's online file and media indexes. Normally, **nsrim** is invoked by **nsrmmdbd**(8) at startup, by the **savegrp**(8) command on completion, and by **nsrd**(8) when *Remove oldest cycle* is selected from the Legato Storage Manager Administrator program. **nsrim** is not normally run manually.

**nsrim** uses *policies* to determine how to manage on-line entries (see **nsr\_policy**(5), **nsr\_client**(5), and the *Legato Storage Manager Administrator's Guide* for an explanation of index policies). Entries that have been in an on-line file index longer than the period specified by the respective client's browse policy are removed. Save sets that have existed longer than the period specified by a client's retention policy are marked as *recyclable* in the media index. When all of the save sets on a volume have been marked recyclable, then the volume is considered recyclable. Recyclable volumes may be selected (and automatically relabeled by a jukebox) by Legato Storage Manager when a writable volume is needed to hold new backups. When a recyclable volume is re-used, the old data is erased and is no longer recoverable.

Unless the **-q** option is used, **nsrim** prints header and trailer information for each group of save sets. The header lists the save set type, the client name, the save set name, and the applicable browse and retention policies that apply to the save set (see the following example). There are four types of save sets:

#### *Normal*

Includes all save sets backed up automatically using **savegrp** that are associated with a schedule, a browse policy, and a retention policy.

#### *Ad hocs*

User initiated save sets are designated by appending *ad hocs* to the header line.

### *Archives*

Save sets that never expire automatically are designated by appending *archives* to the save set line.

### *Migrations*

Save sets that never expire automatically and were created by a file migration application are designated by appending *migrations* to the save set line.

The trailer lists four utilization statistics of the save set after **nsrim** has applied the policies to it. The four statistics are the total number of browsable files remaining in the online index, the grand total of files currently associated with the save set, and the amount of recoverable data out of the grand total of data associated with the save set. For example, **nsrim** may print the following output for a single save set name:

```
mars:/usr, retention policy: Year, browse policy: Month, ad hocs
8481 browsable files of 16481 total, 89 MB recoverable of 179 MB total
mars:/usr, retention policy: Year, browse policy: Month, ad hocs
0 browsable files of 13896 total, 163 MB recoverable of 163 MB total
mars:/usr, retention policy: Year, browse policy: Month 43835
browsable files of 427566 total, 6946 MB recoverable of 7114 MB total
```

When the **-v** option is used, the following information is also printed for each save set: the save set id, creation date, level, file count, size, and status. A save set's status is one of the following:

#### *browse*

The file entries for the save set are browsable (the save set files still exist in the online index). These files are easily restored using Legato Storage Manager's recover mechanisms.

#### *recover*

The age of the save set's does not exceed its retention policy for the save set, but its entries have been purged from Legato Storage Manager's online index. This means that save set is recoverable from the back-up media using **recover** (see **recover(8)**); **scanner(8)** may be also be used to recover the save set, but users should use **recover** first).

*recycle*

The save set is older than its associated retention policy and may be overwritten (deleted) once its backup media is recycled. Until the media is recycled, the save set is also recoverable from the back-up media.

*delete*

The save set will be deleted from the media database. **nsrim** deletes only recyclable save sets that have zero files.

The save set status may be followed by any of the following modifiers:

*(expires mm/dd/yy)*

The save set has an explicit expiration date in the future, and is exempt from any status change.

*(archive)*

The save set never expires, and is exempt from any status change.

*(migration)*

The save set was created by a file migration application and never expires, and is exempt from any status change.

*(scanned in)*

The save set was restored using the **scanner** command, and is exempt from any status change.

*(aborted)*

A save set of questionable size consuming backup media space.

If **nsrim** changes the status of a save set, then it prints the transition symbol -> followed by the new status. For example:

```
17221062 3/05/92 f 23115 files 158 MB recycle
17212499 3/19/92 f 625 files 26 MB recover(aborted)->recycle
17224025 5/23/92 i 0 files 0 KB recover->recycle->delete
17226063 6/05/92 f 3115 files 58 MB recover
17226963 6/09/92 f 3197 files 114 MB browse->recover
17227141 6/10/92 f 3197 files 115 MB browse
```

Once **nsrim** has processed all of the save sets, it flags the file index for cross-checking in **nsrindexd**(8). If the **-l** flag is specified, the cross-check is attempted synchronously, otherwise, it is simply scheduled and **nsrindexd** performs the cross-check when the index is idle. Concurrently, **nsrim** processes the status of any affected Legato Storage Manager volumes. With the absence of the **-q** flag, a line is printed for each volume. The line includes the volume name, the

amount of space used, the total number of save sets, and the status. The status will be one of the following:

*appendable*

More save sets may be appended to the volume. The status may also be modified with *(currently mounted)* which signifies that the volume could transition to the *recyclable* state if it was not mounted for writing.

*read-only, full*

No more save sets can be appended to the volume, nor can the volume be re-used, since it contains some valuable save sets.

*recyclable*

No more save sets can be appended to the volume, and all save sets on the volume have expired.

In addition, the following modifier applies to all four of the previously described states:

*(manual recyclable)*

The volume will not be automatically eligible for recycling when all of its save sets have expired. Instead, the volume may only be recycled by a manual relabel operation. Note that a *read-only* volume can still be recycled unless the *manual-recyclable* flag is also set. The manual-recyclable flag can be set using the Legato Storage Manager administration GUI (**nwadmin(8)**) or the **nsrmm(8)** commands when volumes are labeled or at any time thereafter. This flag is never set automatically.

If the volume status changes, then **nsrim** appends **->recyclable** to the status. If the volume contains some browsable save sets, then this is noted; recoverable save sets are also noted. The odd case where an appendable volume has only recyclable save sets is also noted. For example:

```
jupiter.20: 3474 MB used, 398 save sets, full->recyclable
jupiter.21: 4680 MB used, 440 save sets, full, 249 recoverable
jupiter.22: 4689 MB used, 351 save sets, full, 351 browsable
jupiter.24: 1488 MB used, 141 save sets, appendable, 141 browsable
```

## Retention and Browse Policies

Under normal circumstances, the association between browse or retention policies and client save sets is obvious. However, since a save set may be listed by more than one client resource with the same name, and each client resource may specify different browse and retention policies, determining the policies applicable to a save set is not always straight forward. **nsrim(8)**, uses the following steps to select an

---

instance of a client resource with the client's name. Once the client resource is selected, its browse or retention policy is used for managing information about the save set.

1. Locate all the client resources which belong to the same group that the saveset belongs to. Within this set of client resources, apply the following rules to get the best match. If no client resource belongs to the saveset's group, or if the group no longer exists, or if the saveset is from a pre-version 5 backup when group info was not recorded in the saveset, apply the following rules to all the client resources to get the best match.
2. Locate a client resource explicitly listing the save set. If more than one client resource lists the save set, choose the client resource with the longest policy.
3. Search for a client resource listing the save set "All". If more than one client resource lists the save set "All", choose the client resource with the longest policy.
4. Find the client resource listing a save set with the most common prefix (longest) of the target save set. If more than one client resource lists the save set with the most common prefix, choose the client resource with the longest policy.
5. Among all of the client resources, choose the client resource with the longest policy.

Note that if two or more client resources with the same name exist, it is possible that the browse policy from one instance of the client resource and the retention policy of another instance of the client resource may be used for managing save set information.

Save sets that have no corresponding Legato Storage Manager client resource use the Legato Storage Manager client resources of the server to determine the browse or retention policies.

A save set cannot be purged from the index or marked for recycling until all of its dependent save sets are also eligible for purging or recycling. See the *Legato Storage Manager Administrator's Guide* for an explanation of dependent save sets.

The last (and only) Full save set will not be purged from the on-line index until it is also marked for recycling. In this case, the header line of the save set omits the browse policy and prints a message stating that only one browsable cycle exists.

With the exception of the `-I` option, manual hoc save sets are treated as full save sets that have no dependents. However, unlike true full save sets, the last manual save set is not given any special consideration with regard to index purging.

The expiration time applied to save sets is rounded up to midnight when the elapsed time implied by the policies is greater than or equal to a day. Therefore, **nsrim** should produce the same results whether it is run at 8 a.m. or 5 p.m. on the same day.

## Options

### **-b** *browse*

Uses the specified policy rather than the browse policy found on the client's resource. This is very useful when combined with the **-n** option to see how a new or modified policy will affect the indexes.

### **-c** *client*

Only processes the online file index for the specified client. Normally, all client indexes are processed. This option may be repeated to process multiple clients.

### **-l**

Removes the oldest full save and all save sets dependent on it from the online index. Browse and retention policies are ignored. The save set header information will print the number of browsable full cycles currently in the online index. *Archive* and *Migration* save sets are ignored. With this option, *manual* save sets are treated as normal *incremental* level save sets. This option also sets the utilization threshold to 30 percent.

### **-M**

Master mode (not advised for manual operation). Advises **nsrim** that it is being run by **nsrd(8)** or another Legato Storage Manager daemon and that it should log messages with timestamps, and perform any other behavior expected by **nsrd**.

### **-N** *save set*

Processes only save sets named; all others are skipped. This option can be repeated to process multiple save sets.

### **-n**

Do nothing. Instead, emulate the actions of this command without the index cross-check. Note that trailer statistics reflect current (and not emulated) results.

### **-q**

Run quietly. This option will not generate header, trailer or save set messages.

### **-r** *retention*

Uses the specified policy rather than the retention policy found on the client's resource. This is very useful when combined with the **-n** option to see how a new or modified policy will affect the volumes.

**-v**

Produces a more detailed report. This may produce a large amount of output. When both **-v** and **-q** are issued, they cancel each other.

**-X**

Consistency check the data structures of the save set with the data structures of the volume. This is only required after a Legato Storage Manager failure. This option also sets the utilization threshold to 30 percent.

**-x percent**

Sets the utilization threshold. If, after removing entries, the utilization of an online file index is less than the specified amount, the index is compressed automatically by passing this percentage to **nsrindexd** when requesting a cross-check. The default value is 50 (percent). Note that specifying **-X** or **-I** changes the default to 30 (percent).

## Files

**/nsr/tmp/.nsrim**

**nsrim** locks this file, preventing more than one copy of itself from thrashing the media database.

**See Also:** [nsr\\_client\(5\)](#), [nsr\\_layout\(5\)](#), [nsr\\_policy\(5\)](#), [nsrd\(8\)](#), [nsrindexd\(8\)](#), [nsrmm\(8\)](#), [nsrmmdbd\(8\)](#), [nwadmin\(8\)](#), [recover\(8\)](#), [savegrp\(8\)](#), [scanner\(8\)](#)

## Diagnostics

### **You are not authorized to run this command**

Only root or Legato Storage Manager administrators may run **nsrim** to modify the online indexes. However, any user may run the command with the **-n** option.

### **Cannot fetch client resource for *clientname***

The named client has no resource to extract the browse and retention policies from. In this case, the resources of the Legato Storage Manager server are used.

### **nsrim has finished (cross) checking the media db**

This notification messages appears in the Legato Storage Manager messages window when **nsrim** completes and the command was invoked with the **-q** option, but without the **-c** and **-N** options.

## nsrindexasm(8)

### Name

**nsrindexasm** – Legato Storage Manager module for saving and recovering indexes

### Synopsis

**nsrindexasm** [ *standard-asm-arguments* ]

### Description

The **nsrindexasm** is a standard, external ASM (Application Specific Module) that assists in the saving and recovery of Legato Storage Manager on-line save record index files.

See **uasm**(8) for a general description of ASM's and the [*standard-asm-arguments*]. It is intended that **nsrindexasm** be invoked by **uasm** during **save**(8) or **recover**(8) operations.

Actions performed by **nsrindexasm**, specific to the Legato Storage Manager application during a **save** or **recover**, are:

Locking:

To get a consistent copy of the index, this ASM must coordinate with the **nsrindexd**(8) which reads and writes save records from and to the index.

Architecture independence:

The high speed access methods and data structures implemented by the index code can be computer dependent. This ASM saves only the save records (and not access indexes) in an architecture independent manner. Therefore, Legato Storage Manager indexes may be saved from one computer architecture and recovered to another.

Conservation:

Since only save records themselves are saved, and internal indexes are not saved, considerable network bandwidth and tape space are conserved. Furthermore, the saving of records within the database adhere to rules similar to saving of files in a file system. Therefore, incremental or differential saves of the database consume minimal network bandwidth and tape space.

## Files

***/nsr/index/clientname/db***

The file whose data is saved and recovered by this ASM.

***/nsr/index/clientname/db.RCV***

A temporary file that holds an array of valid save times; this file only exists while the index is being recovered.

**See Also:** [nsr\\_layout\(5\)](#), [nsrindexd\(8\)](#), [recover\(8\)](#), [mmrecov\(8\)](#),  
[save\(8\)](#), [savegrp\(8\)](#), [uasm\(8\)](#)

## nsrindexd(8)

### Name

**nsrindexd** – Legato Storage Manager file index daemon

### Synopsis

**nsrindexd**

### Description

The **nsrindexd** daemon is started by the server **nsrd(8)** daemon. It should not be started by hand. The daemon provides an RPC-based service to the server **nsrd(8)** daemon; direct network access to this service is not allowed. The RPC program and version numbers provided by **nsrindexd** are 390105 and 4, respectively.

The service provided to the Legato Storage Manager system is designed for high performance insertion and deletion of save records into indexes. This performance is obtained by keeping information cached in the **nsrindexd** process address space. When the Legato Storage Manager system wishes to commit a save session's records, it notifies the **nsrindexd** daemon (using a remote procedure call) to flush its volatile state to its file(s).

Since the daemon (or the server computer) may fail at any time, the index files may be left in an inconsistent state. Therefore, the maintenance program, **nsrck(8)** is run automatically by the **nsrd** daemon before the Legato Storage Manager service is started.

Since **nsrindexd** and **nsrck** are run at the same time, both programs use an advisory file locking mechanism on the file **db.SCAVENGE** to synchronize their access to an index.

### Files

***/nsr/index/clientname/db***

Where the client's index records are stored and accessed.

***/nsr/index/clientname/db.SCAVENGE***

When this file exists and **nsrindexd** is not running, the **nsrck** program must be run before **nsrindexd** is restarted.

**See Also:** [nsr\\_layout\(5\)](#), [nsr\(8\)](#), [nsrck\(8\)](#), [nsrd\(8\)](#), [nsrim\(8\)](#), [nsrindexasm\(8\)](#), [nsrls\(8\)](#), [nsrmm\(8\)](#), [nwadmin\(8\)](#)

## Diagnostics

**waiting for lock on *filename*.**

This message indicates that another program is accessing the same file that is required by this daemon. The daemon will wait for the advisory lock to be cleared.

**lock on *filename* acquired.**

Informative message that will eventually follow the previous message.

## nsrinfo(8)

### Name

**nsrinfo** – Legato Storage Manager file index reporting command

### Synopsis

```
nsrinfo [ -vV ] [ -s server | -L ] [ -n namespace ] [ -N filename ] [ -t time ]  
[ -X application ] client
```

### Description

The **nsrinfo** command generates reports about the contents of a client file index. Given a required Legato Storage Manager client name and no options, **nsrinfo** will produce a report of all files and objects, one on each line, in the *backup* name space for that client. It can also generate reports for a specific file index name space, for all name spaces at once, for a particular XBSA application, and can be restricted to a single time (the time at which the entry was entered into the file index, called the *savetime*).

For example, to generate a report of all files backed up in the most recent backup of the */usr* file system for the client *mars*, use the following sequence of commands (assuming the following % character is the shell prompt):

```
% mminfo -r nsavetime -v -N /usr -c pegasus -ot | tail -1 809753754  
% nsrinfo -t 809753754 mars
```

**Note:** The time used in the query is obtained by running the **mminfo(8)** command with a custom report to print the save time for the most recent save set for */usr*. The time printed is passed to **nsrinfo** along with the name of the client (*mars*).

Unless you use the **-L** option (see the following description), you must have Legato Storage Manager Administrator privilege to use this command. In the case of **-L**, you must be a system administrator (*root* on a UNIX system).

### Options

**-v**

Verbose mode. In addition to the filename, the type of the file (see the following description for more information on file types), the internal file index identifier (if any), the size (if it is a UNIX file), and the savetime are printed. This option may be combined with the **-V** option described as follows.

**-V**

Alternate verbose mode. In addition to the filename, the offset within the save set containing the file, the size within the save set, the application namespace (see the **-n** option described as follows for a list of values), and the savetime, are printed. This option may be combined with the **-v** option described previously.

**-s server**

The name of the Legato Storage Manager system to be queried. By default, the server on the local system is queried.

**-L**

Open a file index directly without using the server. This option is used for debugging, or to query the file index while Legato Storage Manager is not running.

**-n namespace**

The file index name space to query. By default the *backup* name space is used. The other recognized values are: *migrated*, *archive* (reserved for future use), *nsr* (for internal use), *informix* (for informix data), *sybase* (for sybase data), *msexch* (for exchange data), *mssql* (for SQL Server data), *notes* (for Notes data), *db2* (for DB/2 data), and *all*. The name space field is case sensitive.

**-N filename**

An exact filename to look for in the file index. Only index entries matching this name exactly will be printed. Note that for some clients, such as NetWare, the name stored in the file index is often not made up of printable ASCII characters, giving this option limited use.

**-t time**

Restricts the query to a single, exact, save time. The time can be in any of the Legato Storage Manager **nsr\_getdate(3)** formats. Every save set created by Legato Storage Manager has a unique save time; these times can be determined by using the **mminfo(8)** command.

**-X application**

Restricts the query to list information for only a specific X/Open Backup Services (XBSA) application. Valid application types are **All**, **Informix**, and **None**. The application type is not case sensitive. See the following section, **Application Types**, for more information.

## File Types

The file index can store entries for all types of clients. Each index entry includes an index entry type to differentiate between the types of entries. In general, only the client that created the index entry will be able to decode the entry.

Index entry types recognized by **nsrinfo** are listed as follows. However, even though these types are recognized, **nsrinfo** can only completely decode one entry type: the UNIX version decodes UNIX entry types, and the NT version decodes NT entry types. For other recognized types, some information may be incomplete.

old UNIX

Clients running pre-Legato Storage Manager 3.0 for UNIX.

UNIX

Clients running pre-Legato Storage Manager 4.0 for UNIX.

UNIX ASDF

Index entries including extended ASM Structured Data Format (ASDF) information for clients running Legato Storage Manager 4.1 and later for UNIX.

UNIX ASDF v2

Index entries from agentless saves for clients running Legato Storage Manager 4.2 and later for UNIX.

UNIX ASDF v3

Index entries for large files (files > 2 gigabytes) for clients running Legato Storage Manager 5.1 for UNIX and beyond.

old DOS

DOS clients running pre-Legato Storage Manager for DOS 2.0.

DOS

DOS, Windows or OS/2 clients running version 2.0 of Legato Storage Manager for DOS, Windows or OS/2.

DOS old ASDF

DOS, Windows or OS/2 clients running version 2.0 of Legato Storage Manager for DOS, Windows or OS/2.

WIN ASDF

Windows or NT clients running Legato Storage Manager for Windows NT 4.2 and later.

WIN ASDF v2

Windows or NT clients running Legato Storage Manager for Windows NT 4.2 and beyond, created by using agentless saves.

old NetWare

NetWare clients running pre-Legato Storage Manager for NetWare 3.0.

NetWare

NetWare clients running Legato Storage Manager for NetWare 3.0 and later.

**OSF 64bit**

A client running OSF/1 with 64bit file sizes and offsets.

**continuation**

A special, internal index entry, generated when a file crosses save set boundaries in a save set series.

## Application Types

**All**

This application type prints out all of the X/Open Backup Services API (XBSA) information available for each object; only XBSA objects are printed. The **-v** and **-V** flags have the same effect here as they do on files.

**Informix**

This application type prints out only those objects recognized as Informix Database objects (XBSA ObjectOwner.bsaObjectOwner is INFORMIX). The **-v** flag behaves as it does with files, while the **-V** flag prints out all the XBSA information about the object (see All, described previously), including the normal **-V** information.

**None**

This application type prints out objects that are *not* XBSA objects, but match the given criteria. For example, this option can be used to print a list of files backed up from a client.

## Files

**/nsr/index/client/db**

**See Also:** [nsr\\_getdate\(3\)](#), [mminfo\(8\)](#), [nsrck\(8\)](#), [nsrindexd\(8\)](#)

## Diagnostics

**bad time value 'time'**

The time value specified in the **-t** option is not in a valid [nsr\\_getdate\(3\)](#) format.

**cannot open index for client *client*: *reason***

The file could not be opened using the **-L** option. The specific reason is printed, although there may be several. The most likely reasons are *permission denied* if the user is not the superuser, and *service busy, try again* if the file index is already locked (by [nsrindexd\(8\)](#), for example).

**cannot create db scan on client**

An internal error occurred while attempting to query the file index. Contact Oracle Technical Support.

**number bad records for client**

This diagnostic is printed at the end of a report if any bad index records were detected. This is a sign that the index is damaged, and may need to be recovered.

**cannot connect to server**

The index server is not available for one of many reasons. For example, the Legato Storage Manager server may be down, or **nsrinfo** may not be able to connect to a running server due to either a resource shortage, or a network problem.

**cannot start session with server**

The index server is running, but refused the connection. The exact reason is printed on the subsequent line of output. The most likely reasons are *permission denied* if the user is not a Legato Storage Manager administrator, and *service busy, try again* if the file index is locked (by **nsrck(8)**,forexample).

**lookup failed to server**

The index server is running, but was unable to process the query. The exact reason is printed on the subsequent line of output.

## Limitations

The command line options should be made as powerful as those of **mminfo(8)**.

The **-v** and **-V** reports are not formatted into columns.

A query for a specific time can take a very long time due to the schema of the file index.

The queries are limited due to the lack of a cross platform browser.

## nsrlic(8)

### Name

**nsrlic** – Legato Storage Manager license reporting command

### Synopsis

**nsrlic** [ -vi ] [ -s server ]

### Description

The **nsrlic** program generates reports about all license information currently active on a Legato Storage Manager server. This command queries the Legato Storage Manager resource database, and formats and displays the results to standard output.

The **nsrlic** program reports: the number of server or universal client licenses, the number of server client licenses used, the number of server client licenses borrowed by workstation clients, the number of remaining server client licenses, the list of server clients connected to the named Legato Storage Manager server, the list of server clients defined in the specified Legato Storage Manager server, the number of workstation client licenses, the number of workstation licenses needed or used, the number of remaining workstation client licenses, the list of workstation clients connected to the specified Legato Storage Manager server, the list of workstation clients defined on the specified Legato Storage Manager server, the number of server clients listed by platform, the number of workstation clients listed by platform, the list of valid ClientPaks installed on the system, and the list of client types allowed by ClientPaks and server enablers installed on the system.

When applications exist which require licensing, **nsrlic** also reports them in the same manner. In this case, though, the output will not contain any references unless either there are licenses available or a connected client is using a license count for such applications.

### Options

**-i**

Interactive Mode. In this mode, you can request different reports, refresh the information, or switch to a different server. The information is requested once and cached until another **connect** command is issued.

**-s** *server*

Selects which Legato Storage Manager server to query. By default, the server on the local system is queried.

**-v**

Verbose mode. In addition to the number of licenses or the number of clients, a list of connected and defined clients is gathered and displayed.

## Usage

The following commands are supported in the interactive mode:

**connect** [ *server* ]

Connects to the named *server*. By default, this is the server on the local system.

**detail**

Produces a detailed report. A list of connected clients (clients that saved to Legato Storage Manager) and a list of defined clients (clients that are defined in the Legato Storage Manager server but not yet saved) are displayed.

**help**

Displays a list of available commands.

**summary**

Displays a summary report.

**?**

Same as help.

**quit**

Immediately exit from **nsrlic**.

**See Also:** [nsrd\(8\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#)

## Diagnostics

Nsrlic will display a "usage" message describing the available options, when characters are used that are not valid for this command.

**command not found**

Indicates that the command is not a supported command.

**RPC error, Remote system error RPC error, Program not registered**

Indicates that some problems were encountered while connecting to the Legato Storage Manager server on the specified system. The **nsrlic** command requires that the Legato Storage Manager daemons be running. Start your Legato Storage

Manager daemons (nsrd) and rerun **nsrlic**. If **nsrd** is already running, you have probably reached a resource limit on the server (for example, not enough memory or no more processes).

**Server xxx does NOT have Self-Identifying capability**

Indicates that the specified Legato Storage Manager server does not have the capability to report license information.

## nsrls(8)

### Name

**nsrls** – List statistics of Legato Storage Manager index files

### Synopsis

```
nsrls [ { clientname ... | -f filename ... } ]
```

### Description

When **nsrls** is used without any options being specified, the number of files in an online index, the number of kilobytes that the online index currently requires, and the usage of the online index with respect to the number of kilobytes allocated to its UNIX file is printed. Administrators can use this command to establish how many files have been saved by a client.

### Options

When invoked with the **-f** option, **nsrls** takes a list of filenames rather than a list of Legato Storage Manager client names. For each legitimate index filename, **nsrls** prints an internal "volume id" number and the filename, then a statistics banner followed by statistics associated with each internal file in the index.

Each internal file has the following four statistics associated with it: an internal file id (**Fid**), the size of the file (**Size**), the number of logical records in the file (**Count**), and a descriptive name for the internal file (**Name**).

The internal files are interpreted as follows:

#### **record files**

These are the internal record files which store the actual data (for example, sr).

#### **index files**

These internal b-tree index files hold the index records used for optimizing database queries. The names of these files contain the extension "\_i\*" (for example, sr\_i0).

#### **temporary files**

These files (with the filename extension "\_t\*") contain temporary records used during sorting. Temporary files are present only while a database is currently being modified.

**transaction log files**

These files (with the extension "\_x\*") contain lists of records to be deleted if a saveset has been interrupted (for example, sr\_x0).

The number, name, function and interpretation of the internal files can change at any time.

An empty argument list prints the statistics for all known clients.

**Example****% nsrls monsoon**

```
monsoon: 318497 records requiring 78 MB
/nsr/index/monsoon/db is currently 91% utilized
```

**% nsrls -f /nsr/index/monsoon/db**

```
Database id 0: /nsr/index/monsoon/db
```

| Fid | Size   | Count  | Name  |
|-----|--------|--------|-------|
| 0   | 60 MB  | 318497 | sr    |
| 1   | 5.8 MB | 318497 | sr_i0 |
| 2   | 4.4 MB | 31510  | sr_i1 |

**See Also:** [nsr\\_layout\(5\)](#), [nsrindexd\(8\)](#)

**Diagnostics****Host ... is not a valid client**

The *clientname* specified is not a client of the Legato Storage Manager server.

**bad database header****bad database version**

These messages indicate that **nsrls** was run against a file that is not a legitimate database file (often a database file created by a different hardware architecture).

**No such file or directory**

The database file name, whether specified explicitly using the **-f** option or specified indirectly by a *clientname*, does not exist.

## nsrmig(8)

### Name

**nsrmig** – Migrate files for long-term storage with Legato Storage Manager HSM

### Synopsis

**nsrmig** [ **-nvx** ] [ **-l percent** ] **-s server** [ **-t savetime** ] [ **-W width** ] [ *path* ]

### Description

**nsrmig** migrates files to the Legato Storage Manager server. Migration means replacing a file with a "stub" that points to a copy of the file made from pre-migration. If the stub is accessed at some later time, the file will automatically be recalled to disk by the Legato Storage Manager server.

Criteria specified in the Legato Storage Manager migration client resource are used to select files for migration. Currently, only regular files are pre-migrated and migrated.

If no *path* argument is specified, the current directory will be migrated. **nsrmig** will not cross mount points, nor will it follow symbolic links.

### Options

#### **-l percent**

Specify a goal percentage for **nsrmig**. Migration will stop when the goal percentage is reached. If the goal percentage is already reached when **nsrmig** is run, then **nsrmig** will do nothing and exit. If the **-l** option is not specified, the goal percentage is read from the appropriate migration client resource.

#### **-n**

No stub replacement. Estimate the total number of files and the total size that will be freed by stub replacement, but don't actually replace qualified files with stubs.

#### **-s server**

Use the Legato Storage Manager server named **server**.

#### **-t savetime**

Migrate files that were pre-migrated at **savetime**.

**-v**

Verbose. Cause the nsrmig program to tell you in detail what it is doing as it proceeds. Specifying multiple **-v** options causes the verbosity level to increase.

**-W width**

The width used when formatting summary information output.

**-x**

Cross mount points that are encountered.

**See Also:** [nsr\\_getdate\(3\)](#), [hosts\(5\)](#), [nsr\(5\)](#), [nsr\(8\)](#), [nsr\\_client\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_service\(5\)](#), [nsrd\(8\)](#), [nsrhmck\(8\)](#), [nsrindexd\(8\)](#), [nsrpmig\(8\)](#), [nsrmm\(8\)](#), [nsrmmmd\(8\)](#), [nsrwatch\(8\)](#), [savefs\(8\)](#), [savegrp\(8\)](#)

## Diagnostics

Exit codes:

**0** Normal exit.

**-1** Abnormal exit.

## nsrmm(8)

### Name

**nsrmm** – Legato Storage Manager media management command

### Synopsis

**nsrmm** [ **-C** ] [ **-v** | **-q** ] [ **-s** *server* ] [ **-f** *device* ]

**nsrmm** **-m** [ **-v** | **-q** ] [ **-s** *server* ] [ **-f** *device* ] [ **-r** ] [ *volume* ]

**nsrmm** **-l** [ **-v** | **-q** ] [ **-s** *server* ] [ **-f** *device* ] [ **-myB** ] [ **-e** *expiration* ] [ **-c** *capacity* ] [ **-o** *mode* ] [ **-b** *pool* ] [ **-R** | *volume* ]

**nsrmm** { **-u** | **-j** } [ **-v** | **-q** ] [ **-s** *server* ] [ **-y** ] [ **-f** *device* | *volume..* ]

**nsrmm** **-p** [ **-v** | **-q** ] [ **-s** *server* ] [ **-f** *device* ]

**nsrmm** { **-d** | **-o** *mode* } [ **-v** | **-q** ] [ **-s** *server* ] [ **-Py** ] [ **-S** *ssid/cloneid* ] | **-V** *valid* | *volume...* ]

### Description

The **nsrmm** program provides a command-line interface to manage the media and devices (tapes, disks, and files) used by Legato Storage Manager servers and storage nodes.

A *volume* is a physical piece of media, for example, a tape or disk cartridge. When dealing with file type devices, *volume* refers to a directory on a file system. Legato Storage Manager must have exclusive use of this directory, as files will be created and removed. The Legato Storage Manager system keeps track of which user files have been saved on which volumes, so they can be more easily recovered. Every volume managed by Legato Storage Manager has a *volume name* (also known as a *volume label*) selected by an operator. A volume name is specified when the volume is first introduced to the system. It can only be changed when a volume is relabeled. The volume should have an external label displaying its volume name for future reference. Legato Storage Manager refers to volumes by their volume names, for example, when requesting a volume for recovery.

The Legato Storage Manager system automatically manages an index mapping saved user files to volumes. Legato Storage Manager also keeps other attributes associated with a volume, including the volume's expiration date and the expected capacity of the volume.

The Legato Storage Manager server requests that specific volumes be mounted by their name for recoveries, or any writable volumes for saves. These requests are submitted through the **nsr\_notification(5)** mechanism. The **nwadmin(8)** console window or the **nsrwatch(8)** command can be used to monitor pending mount requests. Typically, the requests will also be written to the system console, or logged in a file. The same requests can be used as input for software that controls a *jukebox* (a device that automatically loads and unloads volumes).

Before the **nsrmm** command can be used (before any data can be saved or recovered), at least one device must be configured for the Legato Storage Manager server. A device is usually configured with the **nsr\_ize(8)** command when Legato Storage Manager is installed. The Legato Storage Manager configuration may be modified with the **nwadmin(8)** administration menus or the **nsradmin(8)** command after Legato Storage Manager has been installed.

## Options

### **-B**

Verifies that the volume you want to label does not have a readable Legato Storage Manager label. Before labeling the volume, an attempt is made to read any existing label the volume may already possess. If you specify this option and the volume has a valid Legato Storage Manager label that is readable by the device currently being used, the label operation is canceled and an error message is displayed. If the volume does not contain a label that is readable by the current device, the volume may be labeled. This option is used by **nsrd(8)** when automatically labeling volumes on behalf of **nsrmm(8)** requests.

### **-b pool**

Specifies the pool to which the volume belongs. The pool can name any pool currently registered with **nsrd**. The possible values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin(8)** or using the **nsradmin(8)** command. The pool name is referenced by **nsrd** when determining what save sets can reside on the volume. If you omit this option, the volume is automatically assigned to the *Default* pool. If you specify a pool name without specifying a volume name, the next volume name associated with the pool's *label template* resource is used.

### **-C**

Displays a list of Legato Storage Manager configured devices and the volumes currently mounted in them. This list displays only the devices and volumes assigned to the server, not the actual devices and volumes. The **-p** option (described as follows) verifies the volume label. This is the default option.

**-c capacity**

Overrides the default capacity of a volume. Legato Storage Manager normally uses built-in default capacities based on the device type. This option overrides these defaults. The format of the specification is *number multiplier*. *Multiplier* can be one of 'K' (1024 bytes), 'M' (1000 KB), or 'G' (1000 MB). Lower-case letters are also accepted, as are extra characters like spaces, or an extra 'B' after 'K', 'M', or 'G'. *Number* may be any value, including an integer or real number, with up to three decimal places.

**-d**

Deletes the client file indexes and media database entries from the Legato Storage Manager databases. The action does not destroy the volume: instead, it removes all references used by Legato Storage Manager to the volume and the user files contained on it. This option can be used to control the size of the Legato Storage Manager databases.

**-e expiration**

Sets the expiration date for relabeling volumes. This option overrides the default expiration date for the label, which is two years. The value for *Expiration* is entered in `nsr_getdate(3)` format, with a special value of **forever** that is used for migration and archive volumes means that the volume label never expires.

**-f device**

Specifies a device explicitly. When more than one device has been configured, **nsrmm** will select the first device by default. This option overrides the selection made by **nsrmm**.

**-j**

Ejects a volume from the device. This option is similar to performing an unmount operation, except that the volume is also physically ejected from the device, if possible. This feature is not supported by some device types, disk devices, and tapes.

**-l**

Labels (initializes) a volume for Legato Storage Manager to use and recognize. Labeling must be performed *after* the desired volume is physically loaded into the device, either by an operator or a jukebox.

**-m**

Mounts a volume into a device. Mounting is performed *after* a volume is placed into a device and labeled. Only labeled volumes can be mounted. The labeling and mounting operations can be combined into a single command line (see the **Examples** section).

**-o mode**

Sets the mode of a volume, save set, or save set instance (clone). The *mode* can be one of the following: **[not]recyclable**, **[not]readonly**, **[not]full**, **[not]manual** or **[not]suspect**. The **[not]recyclable** modes apply to both volumes or save sets, but not clones. The **[not]readonly**, **[not]full** and **[not]manual** modes apply only to volumes. The **[not]manual** modes are the only valid modes when used with the **-I** option. The **[not]suspect** modes apply only to save set instances, meaning it must be specified along with **-S ssid/cloneid**, not just **-S ssid** by itself (remember that every instance of a save set has a clone id, even the original). See **nsrim(8)** for a discussion of the volume flags. The **suspect** flag is set automatically when a **recover(8)** encounters a media error recovering data from a particular save set clone.

**-P**

When used in conjunction with the **-d** option the corresponding file index entries are purged, without deleting the entries in the media database. The **scanner(8)** command can then be used to recover the file index entries.

**-p**

Verifies and prints a volume's label. To confirm that the external volume label matches the internal label, load a volume into a drive and use this option to display the volume name in the label. Verifying a label unmounts mounted volumes.

**-q**

Quiet mode. This option tells **nsrmm** to print out as little information as possible while performing the requested operation. Generally, only error messages are printed.

**-R**

Relabels a volume. This option rewrites the volume label and purges the Legato Storage Manager indexes of all user files previously saved on the volume. Some of the volume usage information is maintained.

**-r**

Mounts a volume as read-only. To prevent Legato Storage Manager from writing to a volume, specify the read-only flag when mounting the volume. Volumes marked as full and those in the read-only mode (**-o readonly**) are automatically mounted read-only.

**-s server**

Specifies the Legato Storage Manager server to perform the **nsrmm** operation on. See **nsr(8)** for a description of server selection.

**-S *ssid***

Changes (with **-o**) or removes (with **-d**) a save set from the Legato Storage Manager databases. The save set is identified by a save set identifier, *ssid*. A save set instance, or clone, can be specified using the format *ssid/cloneid*. The **mminfo(8)** program may be used to determine save set and clone identifiers.

**-u**

Unmounts a volume. A volume should always unmount a volume before you unload it from a device.

**-V *valid***

Removes a volume from the Legato Storage Manager databases when used in conjunction with the **-d** option. The volume is identified by a volume identifier, or *valid*. The **mminfo(8)** command can be used to determine volume identifiers.

**-v**

Verbose mode. This option polls the Legato Storage Manager server to print out more information as the operation proceeds.

**-y**

Do not confirm (potentially destructive) operations before performing them. This option must be used with extreme care.

## Examples

*Labeling new tapes:*

To introduce a new tape, named **mars.001**, to the Legato Storage Manager system, load the tape in an empty drive, then use the command:

```
nsrmm -l mars.001
```

The tape is labeled with **mars.001** and an entry is made in the appropriate Legato Storage Manager indexes. The **mminfo(8)** command may be used to inspect the volume database and display information about the volumes:

```
mminfo -m
```

*Mounting a tape:*

To mount a Legato Storage Manager volume, use the **-m** option. Note that the volume must have been labeled previously and loaded in the drive:

```
nsrmm -m
```

When mounting, a volume name can also be specified:

```
nsrmm -m mars.001
```

The mount will fail unless the given volume name matches the one read from the media.

Mounting a volume makes the volume available to Legato Storage Manager. When **nsrmm**(8) needs the volume, the label will be read again and confirmed, preventing accidental data loss. Volumes are also verified and mounted automatically if the server recovers after a failure.

*Labeling and Mounting a tape:*

A volume may be labeled and mounted with a single **nsrmm** command by combining the **-m** and **-l** options. The following example, labels a volume **mars.003** and mount it on device **/dev/nrst0**:

```
nsrmm -m -l -f /dev/nrst0 mars.003
```

*Unmounting or ejecting a volume:*

When a volume must be unmounted, use either the **-u** or **-j** option, depending on whether or not the device can physically eject a volume.

```
nsrmm -u
```

When more than one volume is mounted, either the volume name or device can be specified to select the desired volume. For example,

```
nsrmm -j mars.003
```

ejects the volume named **mars.003**.

*Displaying the current volumes:*

The **-C** option displays the configured devices and the mounted volumes. This is the default option.

```
nsrmm -C
```

*Deleting a volume:*

To remove references to a volume and the user files saved on it from the Legato Storage Manager indexes, use the **-d** option. This option does not modify the physical volume, and should only be used when the physical volume is destroyed. Deleting a volume frees up space in the Legato Storage Manager file index and the

Legato Storage Manager media index, but not much more than purging it. The amount of space released depends on the number of user files saved on the volume. The following example deletes the volume **mars.003**:

```
nsrmm -d mars.003
```

The **scanner(8)** command can be used to rebuild the database entries.

*Purging file index entries:*

The file index contains information about each file saved by Legato Storage Manager. Due to size constraints, it may be necessary to purge information from the file index. When a volume or save set is deleted, the corresponding file index entries are also removed. It is also possible to preserve the media database entries of a volume while purging the file index by specifying the **-P** option when deleting.

The following example purges all of the file index entries for volume **mars.001**:

```
nsrmm -d -P mars.001
```

The **scanner(8)** command can be used to recover the file index.

**See Also:** [nsr\(8\)](#), [nsr\\_getdate\(3\)](#), [nsr\\_layout\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_notification\(5\)](#), [mminfo\(8\)](#), [nwadmin\(8\)](#), [nsrmmmd\(8\)](#), [nsradmin\(8\)](#), [nsrim\(8\)](#), [recover\(8\)](#), [scanner\(8\)](#)

## Diagnostics

***type family volume mounted on device, write enabled***

Message indicating that the **-m** (mount) option was successfully performed on a device with the given media *type* and media *family*, for example, 8mm tape.

***saveset is not a valid save set id***

The given save set identifier is not in the valid format. The format is either a single number, for the save set without reference to its instances, or two numbers separated by a slash (/), representing a save set and clone (instance) identifier pair.

***duplicate name; pick new name or delete old one***

It is not permitted to label two tapes with the same name. If you want to reuse a name, remove that volume from the index using the **-d** option.

***Are you sure you want to over-write volume with a new label?***

An attempt is being made to relabel a volume. A positive confirmation will overwrite the existing data on that tape.

***Purge file index entries for type family volume? ...***

After confirmation, the file index entries are removed.

***volume not in media index***

The media index has no entry associated with *volume*, so the **-m** command cannot be used. This problem may be caused by mistyping the volume name when the tape was originally labeled, or deleting it.

**No valid *family label***

The tape or disk in the named device does not have a valid Legato Storage Manager label.

## nsrmmd(8)

### Name

**nsrmmd** – Legato Storage Manager media multiplexor daemon

### Synopsis

**nsrmmd** [ *-v* ] [ *-s server* ] [ *-r system* ] *-n number*

### Description

The **nsrmmd** daemon provides an RPC-based media multiplexing and demultiplexing service. The RPC program and version numbers provided by **nsrmmd** are 390104 and 5, respectively. However, to support multiple instances of the protocol (if the Concurrent Device Support feature is enabled), the version numbers used have the daemon number times one hundred added to them. The daemon numbers always start at one, so the first version registered will be 105, then 205, and so on. One **nsrmmd** for each enabled device is started automatically by **nsrd**. Additional **nsrmmd** daemons may be started when a mount request is pending. To change the number of daemons, alter the number of enabled devices.

### Options

*-n number*

Specify the daemon number.

*-s server*

Specify the controlling server. This option is used on a storage node (see **nsr\_storage\_node(5)**).

*-r system*

Some **nsrmmd** programs run on the server but are controlling a device attached to a *Networker Data Management Protocol (NDMP)* system. Such instances of **nsrmmd** have an optional *-r* argument specifying the system that is being controlled.

*-v*

Verbose: Print out messages about what the daemon is doing.

**See Also:** [nsr\(8\)](#), [nsr\\_layout\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [nsrd\(8\)](#), [nsrmm\(8\)](#), [mm\\_data\(5\)](#)

## nsrmmdbasm(8)

### Name

**nsrmmdbasm** – Legato Storage Manager module for saving and recovering media databases

### Synopsis

**nsrmmdbasm** [ *standard-asm-arguments* ]

### Description

The **nsrmmdbasm** is a standard, external ASM (Application Specific Module) that assists in the saving and recovering of the Legato Storage Manager media multiplexor's database files.

See **uasm**(8) for a general description of ASM's and the [*standard-asm-arguments*]. It is intended that **nsrmmdbasm** only be invoked by **savegrp**(8) or **mmrecov**(8) operations.

Actions performed by **nsrmmdbasm**, specific to the Legato Storage Manager application during a **save** are:

Architecture independence:

The high speed access methods and data structures implemented by the database code are computer dependent. This ASM saves only the records (and not access indexes) in an architecture independent manner. Therefore, Legato Storage Manager media databases may be saved from one computer architecture and recovered to another.

Conservation:

Since only changed records are saved, and not internal indexes, considerable network bandwidth and tape space are conserved.

The **recover** operation of this ASM is the inverse of the **save** operation.

### Files

**/nsr/mm/.nsr**

This directive file causes most files in the directory to be skipped during normal save operations. **nsrmmdbasm** ignores this directive.

**/nsr/mm/mmvolume**

The file which is saved and recovered by this ASM.

**/nsr/mm/mmvolume.r**

A temporary file that stores the contents of a recovered media database until **nsrmmdbd(8)** has completed building a new media database.

**/nsr/mm/mmvolume.s**

A temporary file that this ASM reads when backing up data.

**/nsr/mm/volume.tmp**

A temporary file created when converting an older media database schema to the present schema during recovery.

**See Also:** [nsr\(5\)](#), [nsr\\_layout\(5\)](#), [mmrecov\(8\)](#), [nsrmmmd\(8\)](#),  
[nsrmmdbd\(8\)](#), [nsrindexasm\(8\)](#), [recover\(8\)](#), [savegrp\(8\)](#), [uasm\(8\)](#)

---

# nsrmmdbd(8)

## Name

**nsrmmdbd** – Legato Storage Manager media (volume) management database daemon

## Synopsis

**nsrmmdbd**

## Description

The **nsrmmdbd** daemon provides an RPC-based database service to the local **nsrd(8)** and **nsrmm(8)** daemons, and query-only network access to Legato Storage Manager clients. The RPC program number provided by **nsrmmdbd** is 390107. The RPC version numbers provided by **nsrmmdbd** are 3, 4, and 5. **Nsrmmdbd** is normally started by **nsrd(8)**.

The daemon manages a “media and save set database” located in the file **/nsr/mm/mmvolume**. The primary purpose of the database is to remember which save sets reside on which backup volumes. Numerous access methods are provided to both save set and volume records within the database.

## Files

**/nsr/mm/mmvolume**

File containing the volume database.

**/nsr/mm/cvt**

A temporary file created when converting an older media database schema to the present schema.

**/nsr/mm/.cmprssd**

For performance and space reasons, the database is periodically rebuilt (or compressed). This file is created each time the database is rebuilt; its associated ctime is used to determine when to rebuild the database again. To forcibly compress the database, remove this file and run **nsrim**.

**/nsr/mm/mmvolume.s**

This temporary file is created to hold the media database information that will be saved to tape by **nsrmmdbasm(8)**.

**/nsr/mm/mmvolume.r**

The file (created by **nsrmmdbasm**) that is read when the media database information is being recovered.

**See Also:** [mmrecov\(8\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nsrim\(8\)](#), [nsrmm\(8\)](#), [nsrmmdbasm\(8\)](#), [nsrmm\(8\)](#), [mminfo\(8\)](#)

## Diagnostics

The **nsrmmdbd** diagnostic messages will normally be logged to the **/nsr/logs/daemon.log** file.

Besides the messages listed below, **nsrmmdbd** may generate other diagnostics. Any diagnostics other than those listed below indicate a serious problem with the media database. It may be necessary to recover your media database using **mmrecov(8)** if that occurs.

**media db is converting *path* to version 5****media converting to version 5**

Any media databases created prior to the Legato Storage Manager 4.2 release have to be converted (once) to the new database format. Plan on allowing one second for every 100 save sets.

**media conversion done**

Printed when the conversion is completed successfully.

**media conversion failed! *reason***

Printed when the conversion is terminates unsuccessfully. A more detailed reason may be appended to the message. Legato Storage Manager cannot work until the media database is converted successfully.

**media db is converting *count* volumes**

This is printed after the volumes' data has been dumped from the old database, but before it has been loaded into the new database.

**media db is converting *count* save sets**

This is printed after the save sets' data has been dumped from the old database, but before it has been loaded into the new database.

**media db is saving its data, this may take a while**

Printed when the daemon is dumping its records to a temporary file when the database is being backed up. The service is unavailable while the database is dumping.

**media db is recovering, this may take a while**

Printed when the daemon is reloading its database. The service is unavailable while the data is being reloaded.

**media db is recovering old data, this may take a while**

Similar to the previous message, except that a pre-4.0 database is being recovered and it will have to be converted before service resumes.

**media db is cross checking the save sets**

Printed each time the daemon is restarted. Upon start-up, the daemon sanity checks its records before providing its service.

**media db is open for business**

Printed after any of the previous messages are printed to indicate that the service is once again available.

**A copy of this process is already running!**

Another copy of **nsrmmdbd(8)** is currently running and has exclusive access to the media database. Only one **nsrmmdbd** process should be running on a given computer at a time. This can happen if the previous **nsrmmdbd** was not properly terminated. Use **nsr\_shutdown(8)** or **ps(1)** and **kill(1)** to identify and kill off all the Legato Storage Manager daemons before restarting **nsrd(8)** again.

**Cannot open lock file**

An internal error, check the permissions on the **/nsr/tmp** and **/nsr/mm** directories.

## nsrmon(8)

### Name

**nsrmon** – Remotely control Legato Storage Manager commands and daemons

### Synopsis

**nsrmon**

### Description

The **nsrmon** command is run only by Legato Storage Manager daemons. The command is started by **nsrd(8)** to remotely control commands and daemons on Legato Storage Manager storage nodes running **nsrexecd(8)**. Commands and daemons started remotely include **nsrmmd(8)**. See **nsr\_storage\_node(5)** for additional detail on storage nodes.

**See Also:** [nsr\(5\)](#), [nsr\\_storage\\_node\(5\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nsrexecd\(8\)](#), [nsrmmd\(8\)](#)

# nsrpmig(8)

## Name

**nsrpmig** – Pre-migrate files for long-term storage with Legato Storage Manager HSM

## Synopsis

**nsrpmig** [ **-BEiLnpqv**x ] [ **-s** *server* ] [ **-N** *name* ] [ **-f** *dirfile* ] [ **-b** *pool* ] [ **-g** *group* ] [ **-m** *masquerade* ] [ **-W** *width* ] [ **-C** *clone pool* ] [ **-I** *input file* ] *path*

## Description

**nsrpmig** pre-migrates files to the Legato Storage Manager server. Pre-migration means making a copy of a file on Legato Storage Manager storage in preparation for migration. When a file is later migrated, the disk copy of the file is replaced with a reference to the pre-migrated copy in Legato Storage Manager.

Currently, only regular files are pre-migrated. Criteria specified in the Legato Storage Manager migration client resource are used to select files for pre-migration. The progress of a **nsrpmig** session can be monitored using the X Window System based **nwadmin**(8) program or the **curses**(3X) based **nsrwatch**(8) program for other terminal types.

The **nsrpmig** command will not cross mount points, nor will it follow symbolic links. If the path to be saved is mounted from a network file server, **nsrpmig** will instruct the user to run the save on the remote computer or use the **-L** option.

The directive files (see **nsr**(5)) encountered in each directory will be read by default, and they contain special instructions directing how particular files are to be saved (compressed, skipped, and so on.). These files are named *.nsr<sub>hsm</sub>*.

Note that the directive files used by Legato Storage Manager for save and recover named *.nsr* are ignored by **nsrpmig**.

Each file in the subdirectory structures specified by the *path* arguments will be encapsulated in a Legato Storage Manager save stream. This stream of data is sent to a receiving process (see **nsrd**(8)) on the Legato Storage Manager server, which will process the data, adding entries to the on-line index (see **nsrindexd**(8)) for each file in the stream, with the data finally ending up on some long term storage media (see **nsrmm**(8)).

Details about handling media are discussed in **nsrmm**(8) and **nsr\_device**(5).

## Options

**-E**

Estimate the amount of data which will be generated by the save, then perform the actual save. Note that the estimate is generated from the inode information, and thus the data is only actually read once.

**-i**

Ignore any .nsrshm directive files as they are encountered in the subdirectory structures being saved.

**-L**

Local. Saves will be performed from the local Legato Storage Manager client, even when files are from a network file server. To recover these files, run **recover(8)** with the **-c *client*** arguments, where *client* is the name of the Legato Storage Manager client that did the **save**.

**-LL**

In addition to treating the backup as a local backup, cause an extra line to be printed at the end of the completion output of the form “complete savetime=number”, where number is the savetime of the save set created by this backup. This option is meant to be used by the **savegrp(8)** command in performing automatic cloning.

**-m *masquerade***

Specifies the tag to precede the summary line with. This option is used by **savegrp(8)** and **savefs(8)** to aid in savegrp summary notifications.

**-n**

No save. Estimate the amount of data which will be generated by the save, but do not perform the actual save.

**-v**

Verbose. Cause the **save** program to tell you in great detail what it is doing as it proceeds.

**-p**

Exit with status 0. Used by server to determine if client installed properly.

**-q**

Quiet. Display only summary information and error messages.

**-s *server***

Specify which computer to use as the Legato Storage Manager server.

**-N** *name*

The symbolic name of this save set. By default, the *path* argument is used as the save set name.

**-f** *dirfile*

The file from which to read prototype default directives (see **nsr(5)**). A *dirfile* of - causes the default directives to be read from standard input.

**-b** *pool*

Specifies a particular destination pool for the save.

**-g** *group*

This option is used by **savegrp(8)** and **savefs(8)** to denote the group of the save (see **nsr\_client(5)** and **nsr\_group(5)**) and is used by the Legato Storage Manager server to select the specific media pool.

**-C** *clone pool*

Generate a clone of this archive save set to the specified *clone pool*.

**-I** *input\_file*

In addition to taking the paths to save from the command line, read paths to save from the named file. The paths must be listed one on each line. If no paths are specified on the command line, then only those paths specified in the file will be saved.

**-W** *width*

The width used when formatting summary information output.

**-x**

Cross mount points.

**-B**

Force save of all connecting directory information from root (“/”) down to the point of invocation.

**See Also:** **curses(3X)**, **nsr\_getdate(3)**, **nsr(5)**, **nsr(8)**, **nsr\_client(5)**, **nsr\_device(5)**, **nsr\_group(5)**, **nsr\_service(5)**, **nsrd(8)**, **nsrhmck(8)**, **nsrindexd(8)**, **nsrmig(8)**, **nsrmm(8)**, **nsrmmmd(8)**, **nsrwatch(8)**, **nwadmin(8)**, **recover(8)**, **save(8)**, **savefs(8)**, **savegrp(8)**

## Diagnostics

Exit codes:

**0** Normal exit.

-1 Abnormal exit.

# nsrports(8)

## Name

**nsrports** – Port configuration tool

## Synopsis

```
nsrports [ -sserver ] [ -a auth_server ] [ -S | -C ] [ range ]
```

## Description

The **nsrports** command is used to display and set ranges of ports used by the Legato Storage Manager software. The port ranges are stored by **nsrexecd(8)** in the *NSR system port ranges* resource. When **nsrports** is executed without any options, the program displays the configured ranges for the system on which the command is being run.

Only users executing a tool on the system can change the ports used by a system. This behavior can only be modified if by using **nsradmin(8)** to modify the administrator attribute for the resource storing the ranges.

There are also two additional options for viewing and setting the port ranges. The first is through the graphical user interface, **nwadmin(8)**. The second is by using the **nsradmin(8)** tool. Execute the program as follows:

```
# nsradmin -s server -p nsrexec
```

where *server* is the system to display ports for.

## Options

**-s** server

Specifies the system to contact.

**-a** auth\_server

Specifies a Legato Storage Manager server. This option is required if **nsrports** is connecting to a remote system that is located on a different platform than the system on which the command is being executed.

**-S**

Sets the system's service ports range to the specified range.

**-C**

Sets the system's connection ports range to the specified range.

**See Also:** [nsrexecd\(8\)](#), [nsradmin\(8\)](#), [nwadmin\(8\)](#)

# nsrrepack(8)

## Name

**nsrrepack** – Volume repacking command

## Synopsis

**nsrrepack** [-v] [-n] [-F] [-b *pool*] { -f *file* | *volname...* }

**nsrrepack** [-v] [-n] [-F] [-b *pool*] -V { -f *file* | *valid...* }

**nsrrepack** -a [-v] [-n] [-c *Threshold-for-repacking* ]

[-N *Desired-number-of-free-volumes* ] [-d *MaxDuration* ] [-C *Repacking-Schedule* ]  
-B *pool*

## Description

Data is stored in save sets on Legato Storage Manager volumes. From time to time, save sets will expire according to administrative configurations. This may result in volumes that contain both expired and nonexpired save sets. The space associated with the expired save sets is wasted space. In other words, the volumes that have wasted space are not being used efficiently.

This wasted space cannot be reused because saving to a tape involves appending to the tape. The amount of waste can be determined by executing the command **mminfo -a -r 'volume,%recyc'** .

The command **nsrrepack** reclaims the wasted space by repacking candidate volumes, which may be explicitly specified by the user, or the command will identify them from membership in a specific media pool. In the repacking process, nonexpired save sets of candidate volumes are cloned to other volumes. ("Cloning" means duplicating.) After this has been completed, both the expired and nonexpired save sets of the source candidate volumes are no longer needed. If a volume no longer contains save sets, it will be marked as recyclable and can be reused.

**nsrrepack** can perform either a manual repack or an automatic repack. In a manual repack (no **-a** option), the user explicitly specifies which volumes to repack. In autorepack (**-a** option), the utility will determine which volumes to repack based on certain autorepack criteria. In either case, Legato Storage Manager will move data to the next available volume (for example, whose mode is appendable). The user cannot select target volumes, but the user may specify the target pool.

**nsrrepack** relocates *complete* save sets. If there are save sets that are contained on multiple volumes, then all pertinent volumes will also be involved in the repacking operation.

By default, **nsrrepack** can repack any volume *except* those that are still in the append mode. The reason is because these volumes cannot be recycled. The user, however, can override this restriction with the **-F** option.

With the **-b** option, the system administrator can use **nsrrepack** to move save sets onto a different media pool. However, all media that will be used as the destination of a **nsrrepack** operation must be in the same pool type. (See **nsr\_pool(8)** for a description of the various pool types.) With this option, **nsrrepack** can be used to migrate all save sets from one media type to another. Suppose the system administrator wants to transfer all the save sets from 8mm tape to a set of DLT tapes. In this case, the administrator must remember to use the **-F** option, otherwise save sets on append-mode volumes will not be repacked.

In autorepack (**-a** option), the utility will determine which volumes to repack based on the following criteria:

1. which volumes are candidates for repacking (**-c** option),
2. how far to repack before stopping (**-N** option),
3. the maximum time duration to repack (**-d** option), and
4. which day to perform repacking (**-C** option).

**nsrrepack** will repack one volume at a time, starting with volumes with the greatest amount of expired save sets. It keeps on repacking data until it has either reached the autorepack criteria or until there are no more volumes to repack; or if the **-d** option was defined, it repacks data until it has reached the maximum duration time. The **-B** option *must* be specified in order to indicate which pool to perform autorepacking. See the following **Options** section for more information on each respective autorepack related options. Also, look at the following **Examples** section for more information on autorepack.

After the save sets have been repacked, **nsrrepack** will backup the index files of a set of clients and the index file and the bootstrap file of the backup server. The set of clients will include those clients whose save sets were repacked. To backup the clients' index files, **nsrrepack** will run the command **save(1m)**; to backup the server's files, **nsrrepack** will execute the command **savegrp(1m)**. In both cases, **nsrrepack** will query the nsr.res database for the group affiliation of each client and server. If there is no registered group affiliation, then the group **Default** will be used in determining the destination pool for the index files and the bootstrap.

## Options

### **-b** *pool*

Specifies the destination media pool for manual repacking. The pool may be any pool currently registered with **nsrd**(8); the selected pool *must* be the same pool type as the source volume. Possible pool values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin**(8). Pool values are also listed in the **NSRpool** resource (see **nsr\_pool**(5)).

If this option is omitted, then the save sets are automatically repacked to volumes whose media pool is the same as that of the source volumes. This option is only used in manual repack.

### **-B** *pool*

Specifies the destination media pool for autorepacking. The pool must be any pool currently registered with **nsrd**(8) that has the same pool type as that of the source volumes. The possible values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin**(8). This option *must* be specified with **-a** option.

### **-c** *Threshold-for-repacking*

Used in autorepacking, this parameter specifies the percentage value that identifies potential volumes as repacking candidates. The percentage value measures the amount of recyclable space on the volume. Any volume whose recyclable percentage is greater than (or equal to) *Threshold-for-repacking* is a repack candidate. You can use the **mminfo -a -r 'volume,%recyc'** command (see **mminfo**(8)) to determine the volumes' recyclable percentage values. The default value is 50%.

### **-C** *Repacking-Schedule*

The repacking schedule that specifies on what days a pool's volumes will be repacked. Possible repacking schedule values can be viewed by selecting the Schedule menu item from the Administration menu of **nwadmin**(8). Repacking schedules are also listed in the **NSR repack schedule** resource (see **nsr\_repack\_schedule**(5)). This option is only available with autorepacking. If this option is not used, the **nsrrepack** will perform its operation without checking against any configured schedule.

### **-d** *Duration*

The duration time specifies a window for starting the repacking of a volume. The duration is specified in an *hh:mm* format. Say that the **nsrrepack** command was initiated at 11:00 pm with the duration length of 4:30, **nsrrepack** will repack a volume as long as the current time is between 11:00 pm and 3:30 am. If not specified, then **nsrrepack** will repack with no regard to time.

It is important to note that the actual duration of repacking is subject to the completion of a saveset. For example, suppose the duration is set to 4 hours. Going into 3 hour 45 minutes of repacking, **nsrrepack** starts to repack the next save set which will take 1 hour. In this case, repacking will not stop at 4 hours, but will continue until this last save set has been repacked successfully.

This option is only available with autorepacking.

**-f file**

Reads the volume names or volume identifiers from the named file instead of listing them on the command line. The values must be listed one on each line in the input file. The *file* may be "-", in which case the values are read from standard input.

**-F**

Forces **nsrrepack** to override any default restrictions and to perform repacking. The restrictions are: 1) Cannot repack volumes in append mode; 2) Cannot repack volumes in Read-Only mode; 3) Cannot repack volumes that were read in using **scanner(8)**. The default is no override.

**-n**

Causes no actual repacking to occur. It merely simulates the repacking activity. This option is used to display which volumes are involved in the repacking process. For autorepacking, it also displays how much it can repack and determines, in advance, if the autorepack criteria is achievable or not.

**-N *Desired-number-of-free-volumes***

This parameter indicates the desired number of free/recyclable volumes needed in a pool. This option is only used in autorepacking. If the desired number of free volumes is already met, then repacking will not take place.

The actual number may be smaller because there are fewer volumes to repack, or the duration period ( **-d** option) is specified.

The default is repack as much as possible.

**-v**

Enables verbose operation. In this mode, additional messages are displayed about the operation of **nsrrepack**, such as save sets that cross volumes.

**-V**

Causes **nsrrepack** to treat subsequent command line parameters as volume identifiers, not volume names. Volume identifiers can be found using the **mminfo -mv** report.

## Examples

Repack all save sets on the volume **mars.001** to a volume in the **Silo** pool:

```
nsrrepack -b Silo mars.001
```

Perform autorepack in the **Silo1** pool. Attempt to repack as many volumes as necessary to achieve 30 recyclable (or free) volumes. Do not attempt to repack any volume that has least 25% of expired save sets:

```
nsrrepack -a -B Silo1 -N 30 -c 25
```

The administrator plans to upgrade the existing pool of 3480 tapes to 3490E tapes. The administrator must move all existing data from 3480 tapes onto 3490E tapes. All 3480 tapes have been identified and listed in a text file called **3480tapes**. All 3490E tapes are registered into the **jupiter** pool:

```
nsrrepack -F -b jupiter -f 3480tapes
```

The administrator wants to get a preview of an auto repacking sessions for **Silo1** pool. He wants to see all the source volumes involved in the repacking process:

```
nsrrepack -a -n -B Silo1
```

**See Also:** [mminfo\(8\)](#), [nsr\(8\)](#), [nsr\\_pool\(5\)](#), [nsr\\_repack\\_schedule\(5\)](#), [nsrd\(8\)](#), [nsrmmd\(8\)](#), [nwadmin\(8\)](#)

## Caveat

Legato Storage Manager Version 4.2.5.A implemented a command-line option, **-s server**, which allowed the administrator to repack volumes on a remote Legato Storage Manager server. This option is no longer available because **nsrrepack** now executes the commands **save(1m)** and **savegrp(1m)**, which run locally.

## Diagnostics

The exit status is zero if all of the requested save sets were repacked successfully; nonzero otherwise.

Several messages are printed which denote a temporary unavailability if the **nsrd(8)** for repacking data. They are self-explanatory. In addition, you may see one of the following messages:

**Candidate threshold value *value* must be a positive integer**

The given candidate threshold value is in the wrong format. It must be a positive integer value."

**Cannot contact media database**

The media database (and most likely, other Legato Storage Manager services as well) on the server is not answering queries. The server may need to be started, or if it was just started, it must finish its startup checks before answering queries.

**Cannot open nsrrepack session with server**

This message is printed when the server is not accepting repack sessions. A more detailed reason is printed on the previous line.

**Cannot read file: error message.**

Unable to read the file. The error message will indicate the reason.

**Cannot repack backup and archive data together**

Archive and backup data are fundamentally different and cannot be repacked to the same pool. You need to run nsrrepack twice, once to repack the backup save sets and once more for the archive save sets.

**Cannot repack volume *volume* because its media index was built using scanner(8)**

nsrrepack will not repack any volume that was scanned in using scanner(8). Information gathered from scanner(8) is not complete enough to perform a successful repack operation.

**Corruption detected in media database for ssid**

Inconsistent information was detected for ssid. Check to make sure that the media database was not tampered.

**Failed to repack the following save sets.**

The following save sets did not get repacked. This message should also be accompanied by other messages explaining a more detailed reason of the problem.

**Cloneid *cloneid* is in incomplete state. Can not repack it.**

The cloneid is detected to be invalid. This cloneid will not be repacked.

**Cloneid *cloneid* is in suspect state (bad read). Can not repack it.**

The cloneid is detected to be invalid. This cloneid will not be repacked.

**Failed occurred during repacking**

A failure occurred. This message should also be accompanied by other messages explaining a more detailed reason of the problem.

**Invalid duration value. Must be in hh:mm format**

The given duration value is in the wrong format.

**Media pool *pool* does not exist.**

The given media pool does not exist.

**Must be root, or the group 'operator', to run this command.**

You do not have the necessary privilege to run this command.

**no complete save sets to repack**

No complete save sets were found that matched the requested command line parameters.

**Pool must be defined**

nsrrepack cannot continue unless an existing media pool is defined.

**Recyclable criteria value *value* must be a positive integer**

The given recyclable criteria value is in the wrong format. It must be a positive integer value."

**There are no free volumes to repack to. Cannot repack.**

nsrrepack cannot repack because there are no free volume to repack to. Try adding in more volumes into the media pool.

**There is on-going activity in volume *volume*. Will skip repacking this volume**

nsrrepack will not repack any volume in which there are other activity that uses it.

**Unable to find any free volume to repack to. Cannot repack.**

nsrrepack cannot find any free volume to repack to. Either add more free volume or expire more save sets.

**Unable to query media pool information from nsr.res**

Error occurred while attempting to get media pool information in nsr.res. Look at nsr.res and determine if it got corrupted.

**Unable to query volume information for *valid***

nsrrepack has trouble getting volume information while attempting to communicate to the media database.

**Volume *volume* has a capacity of zero. Cannot**

Repacking must know the capacity before it can be repacked. For example, this message will appear for optical volume that has no default capacity defined.

**Volume *volume* cannot repack to pool *pool* because**

Repacking cannot occur across two different pool types. For example, you cannot repack files from non-clone pool type to a clone pool type.

**waiting 30 seconds then retrying**

nsrd is busy and so **nsrrepack** will automatically retry its request until the condition is cleared. For example, if all of the active devices are busy saving or recovering, **nsrrepack** cannot use those devices and must wait for two of them to become free.

## nsrretrieve(8)

### Name

**nsrretrieve** – Retrieve Legato Storage Manager archive save sets

### Synopsis

```
nsrretrieve [ -f ] [ -n ] [ -q ] [ -i {nNyYrR} ] [ -d destination ] -s server  
[ -S ssid/cloneid ] ... [ -A annotation ] ... [ path ] ...
```

### Description

**nsrretrieve** is used to restore archive save sets from the Legato Storage Manager server. No browsing is available using **nsrretrieve**. Use of **nsrretrieve** is restricted to administrators and users on the list of archive users for an archive client resource, see the **nsr\_client**(5) man page for further details. When not running as root, only files the user owns can be recovered.

### Options

**-A *annotation***

*Annotation* is a regular expression which uniquely identifies a single archive save set (see **nsrarchive**(8)). The regular expression is of the form used by **grep**(1).

**-S *ssid/cloneid***

*Ssid* specifies the save set id's for the save set(s) to be retrieved. When there are multiple clone instances for an archive save set, the *cloneid* can be also be specified to select the particular clone instance to be retrieved from. When no *path* arguments are specified, the entire save set contents will be retrieved. To restrict the archive save set retrieve to only particular directories or files matching a given path prefix, exact matching *path*'s can be specified to limit which directories and files are retrieved.

**-d *destination***

Specifies the destination directory to relocate retrieved files to. Using this option is equivalent to using the **relocate** command when in interactive mode (discussed as follows).

**-s *server***

Selects which Legato Storage Manager server to use.

**-q**

The **nsrretrieve** command normally runs with verbose output. This flag turns off the verbose output.

**-f**

Indicates that retrieved files will overwrite existing files whenever a name conflict occurs.

**-n**

When retrieving, do not actually create any directories or files.

**-i {nNyYrR}**

Specifies the initial default overwrite response to use when recovering files and the file already exists. Only one letter may be specified. This option is the same as the **uasm -i** option when running in recover mode. See the **uasm(8)** man page for a detailed explanation of this option.

**See Also:** [grep\(1\)](#), [uasm\(8\)](#), [nsrarchive\(8\)](#), [nsr\\_service\(5\)](#), [nsr\\_client\(5\)](#), [nsr\(8\)](#), [nsrd\(8\)](#)

## Diagnostics

**nsrretrieve** complains about bad option characters by printing a “usage” message describing the available options.

### **Cannot open retrieve session with *server***

This message indicates that some problem was encountered connecting to the Legato Storage Manager server on the named computer.

### **cannot retrieve backup save sets**

**nsrretrieve** can only be used to restore archive save set data

## nsrssc(8)

### Name

**nsrssc** – Legato Storage Manager save set consolidation program

### Synopsis

```
nsrssc -c client -N saveset [ -p pool ] [ -r ] [ -vq ]
```

### Description

**nsrssc** consolidates the most recent level 1 (partial) save set and its corresponding full level save set into a new full level save set. This consolidation process effectively achieves the same outcome as a full level backup at the time partial backup was performed.

Normally, **nsrssc** is invoked within **savegrp(8)** as part of a *Consolidation* level backup. During the *Consolidation* level backup, **savegrp(8)** automatically generates a level 1 backup, then calls **nsrssc** to create a consolidated backup, using the latest full level save set.

Using **nsrssc** allows for greater flexibility in scheduling backups and save set consolidation. Unlike the **savegrp(8)** command, which completes a consolidation backup promptly after the level 1 backup is completed, **nsrssc** enables you to schedule the consolidation at a different time. Scheduling a time between the full backup and consolidation backup frees up Legato Storage Manager to complete other processes.

If **nsrssc** is executed manually, the most recently backed up save set must be a level 1 save set; otherwise, the consolidation will not be successful.

The **nsrssc** command requires at least two active devices. The consolidation process uses simultaneous device reads and writes to create its consolidated save set. This mechanism creates a restriction upon the location of the newly created save set. The new *saveset* *cannot* be created on the same volume on which the partial or full save set from which it was derived reside. Also, volumes containing the previous full and level 1 must reside on the same storage node.

## Options

**-c** *client*

The name of the client whose saveset should be included for the consolidation process.

**-N** *saveset*

The name for the generated consolidated save set.

**-p** *pool*

Specifies the destination media pool to build the consolidated full save set. The pool may be any pool currently registered with **nsrd(8)**; the selected pool *must* be the same pool type as the previous full level save set. Possible pool values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin(8)**. Pool values are also listed in the **NSR pool** resource (see **nsr\_pool(5)**). If this option is omitted, then the consolidated save sets is automatically built on volume(s) whose media pool is the same as that of the previous full level save set.

**-r**

Removes the level 1 save set. If the level 1 save set is on tape, then the save set will be expired. If the level 1 save set is on diskfile type volume, then the save set (both its index entries, its media database entries, and the actual save set data on disk) is removed. Please note that **nsrssc** will *never* attempt to remove the level 1 if consolidation fails.

**-v**

Enable verbose operation. In this mode, additional messages may be generated during the consolidation process.

**-q**

Run quietly. (This is the default mode.)

**See Also:** [nsr\\_schedule\(5\)](#), [mminfo\(8\)](#), [savegrp\(8\)](#)

## Examples

The following examples demonstrate how save set consolidation can be performed. In both examples, a save set defined in a group name *elmanco* is consolidated for client *delepanto*. The save set data for group *elmanco* is */etc* and */users*.

*Example #1:*

To perform a save set consolidation, perform the following commands:

```
savegrp -G elmanco -l 1 -I  
nsrssc -c delepanto -N /etc  
nsrssc -c delepanto -N /users
```

Note that this example is almost the same as doing a **savegrp -G elmanco -l c**. The only differences are: 1) no index and bootstrap is backed after data is consolidated. 2) if there is a failure during the consolidated process, a full backup is not performed.

*Example #2:*

To direct level 1 data to a disk cache (file-type device) and have the level 1 save set removed after a full level save set is built on tape, perform the following operations:

First, set-up a pool which only accepts level 1 data and its devices are only file type devices. Then run the following commands:

```
savegrp -G elmanco -l 1 -I  
nsrssc -c delepanto -N /etc -r  
nsrssc -c delepanto -N /users -r
```

This process removes the level 1 completely. Also, since fast media (the disk-file type) is involved, this process may very well be a much faster way of generating a full level save set then compared to do a regular full level backup.

## Diagnostics

On successful completion, **nsrssc** returns zero; otherwise, a nonzero value is returned.

Some error codes are:

- 98 Failed because the level 1 and previous full are not in the same storage node.
- 99 Failed, most likely due to a renamed/deleted directory condition

You may also see one of the following messages:

**You are not authorized to run this command**

Only root or Legato Storage Manager administrators may run **nsrssc**.

**Cannot contact media database**

Most likely, **nsrmmmd(8)** is unavailable to answer queries, or an additional Legato Storage Manager daemon may be terminated. In this case, the system administrator must determine if the Legato Storage Manager services need to be restarted. Note that there may be a small interval during startup when the services may be unavailable to answer any queries.

## nsrstage(8)

### Name

**nsrstage** – Legato Storage Manager save set staging command

### Synopsis

**nsrstage** [ *-v* ] [ *-d* ] [ *-s server* ] [ *-b pool* ] *-m -S* { *-f file* | *ssid...* }

**nsrstage** [ *-v* ] [ *-s server* ] *-C -V volume*

### Description

The **nsrstage** program is used on a manual basis to migrate existing save sets. Migration is the process of moving one or more save sets between volumes. The process begins with a clone of the specific save sets to the new volume specified, followed by the deletion of cloned save set(s) entries from the media database, and finally the removal of the save sets from the original source volume(s), if possible. The second and the third operations are triggered by the successful completion of the previous operation. The data is moved to new media volumes, making room for new data on the original volumes.

Migration can be made onto any media type (for example, save sets on a file volume can be migrated to an optical disk). The **nsrstage** program does not perform simple *volume migration*; it migrates full save sets.

You can specify exactly which copy (clone) of a save set to use as the source (see the *-S* option, described as follows).

### Options

*-b pool*

Specifies the name of the media pool to which the data should be migrated. The pool may be any pool currently registered with **nsrd**(8). The possible values can be viewed by selecting the Pools menu item from the Administration menu of **nwadmin**(8). If you omit this option, the cloned save sets are automatically assigned to the *Default Clone* pool.

*-m*

Performs the actual migration operation.

**-s** *server*

Specifies a Legato Storage Manager server to migrate save sets from. See **nsr(8)** for a description of server selection. The default is the current system.

**-v**

Enables verbose operation. In this mode, additional messages are displayed about the operation of **nsrstage**, such as save sets that cross volumes, or save set series expansions.

**-d**

Delete the input file used to specify the save set identifiers that need to be staged. This option must always be specified in conjunction with a **-f** option.

**-C**

Instructs **nsrstage** to perform a volume cleaning operation. Scans a volume for save sets with no entries in the media database and recover their space. This operation can only be performed on file type volumes.

**-S**

Causes **nsrstage** to treat subsequent command line parameters as save set identifiers. Save set identifiers are unsigned numbers. You can find out the save set identifier of a save set using the **mminfo -v** command (see **mminfo(8)**). The **-S** option is useful when you want to migrate individual save sets from a volume, or to migrate all save sets matching some **mminfo** query. See the following examples for one possible use. The save set identifiers also specify exactly which copy of a save set with multiple copies to use as the source. To specify exact copies, use the *ssid/cloneid* format for each save set identifier. In this case, the *ssid* and the *cloneid* are unsigned numbers, separated by a single slash (/). You can find out the *cloneid* for a particular copy by using the **mminfo -S** report.

**-f** *file*

Instructs **nsrstage** to read the save set identifiers from the file specified, instead of listing them on the command line. The values must be listed one on each line in the file. The *file* may be "-", in which case the values are read from the command line.

**-V**

The name of the volume to be cleaned. This option cannot be used with **-S** or **-m** options.

## Examples

Migrate save sets **1234** and **4568** to a volume in the **Offsite Clone** pool:

```
nsrstage -b 'Offsite Clone' -m -S 1234 4567
```

Migrate clone instance **12345678** of save set **1234** to a volume in the **Default Clone** pool:

```
nsrstage -m -S 1234/12345678
```

Migrate all save sets created since last Saturday to a volume in the **Default Clone** pool:

```
nsrstage -m -S 'mminfo -r ssid -q 'savetime>last saturday''
```

Recover space from volume **jupiter.013**:

```
nsrstage -C -V jupiter.013
```

Only complete save sets can be migrated by **nsrstage(8)**.

**See Also:** [nsrclone\(8\)](#), [nsr\\_getdate\(3\)](#), [mminfo\(8\)](#), [nsr\(8\)](#), [nsr\\_pool\(5\)](#), [nsrd\(8\)](#), [nsrmmd\(8\)](#), [nwadmin\(8\)](#)

## Diagnostics

The exit status is zero if all of the requested save sets migrated successfully, nonzero otherwise.

Several messages are printed denoting a temporary unavailability of **nsrd(8)** for migrating data. These are self-explanatory. In addition, you may see a message from the following list.

### **Adding save set series which includes ssid**

If running in verbose mode, this message is printed when **nsrstage** notices that a requested save set is continued, requiring the entire series to be migrated (even if only part of the series was specified by the command line parameters).

### **Cannot contact media database on server**

The media database (and most likely, other Legato Storage Manager services as well) on the named server is not answering queries. The server may need to be started, or if it was just started, it must finish its startup checks before answering queries.

### **Cannot open nsrstage session with server**

This message is printed when the server is not accepting migration sessions. A more detailed reason is printed on the previous line.

### **number is not a valid save set**

The given save set identifier is not valid. Two forms are understood: simple save set identifiers and those with a cloneid specified. Simple save set are unsigned

numbers. The save set with the cloneid form is specified as two unsigned numbers separated by a single slash (/).

**save set *number* does not exist**

The given save set (from a **-S** save set list) does not exist. Verify your save set identifiers using **mminfo(8)**.

**save set clone *number/cloneid* does not exist**

You specified a specific clone of a save set, but that save has no clones with that clone identifier. Verify your save set identifiers using **mminfo(8)**.

**volume *name* does not exist**

The given volume, (if you specified the **-V** option) does not exist in the media database.

**waiting 30 seconds then retrying**

A temporary error occurred and **nsrstage** will automatically retry its request until the condition is cleared. For example if all of the devices are busy saving or recovering, **nsrstage** cannot use these devices and must wait for two of them to become free.

**Space can only be recovered from file type devices.**

The given volume (if you specified the **-V** option) is not a file type volume. This message is also printed after a successful migration of data from volumes of type other than file.

## nsrwatch(8)

### Name

**nsrwatch** – Character-based display of Legato Storage Manager status

### Synopsis

**nsrwatch** [ **-s** *server* ] [ **-p** *polltime* ]

### Description

The **nsrwatch** program displays a Legato Storage Manager server's status. The server's name is specified by the optional **-s** *server* argument. If no server is specified, it defaults to the same server that would be used by a command such as **recover(8)** in the current directory. If there is no Legato Storage Manager service on the selected computer, the command issues an error message. The polling interval is specified by the optional **-p** *polltime* argument (in seconds). The default is two seconds.

Users can run **nsrwatch** from any terminal that has enough **termcap(5)** capabilities for cursor positioning; it does not require any particular window system. The **nsrwatch** program gets its information through remote procedure calls to the specified server, so it can be used from any computer that can access the server through the network.

The **nsrwatch** display is divided into a header and several panels: the *Server* panel, the *Device* panel, the *Sessions* panel, the *Messages* panel, and the *Pending* message panel. The panel sizes will be adjusted depending on the size of the terminal or window being used.

The header contains the name of the server and the current time. The *Server* panel provides current status of the server. The first line of the panel is reserved for error messages. This line is usually blank. The next line tells how long the server has been up, and the server's release version (which may not be the same as the client's release version). The following lines display how many saves and recovers the current server has performed.

The *Device* panel displays the devices known to the current server. For each device, the panel displays its name, the device type, the name of the mounted volume, or **(unmounted)** if no volume is mounted, and device status. The name may be followed by **(J)** if the device is configured as part of a jukebox device. The *Sessions* panel provides current save set information for each active session (saving,

recovering, or browsing). The *Messages* panel displays a history of messages of general interest to the operator. Finally, the *Pending* message panel displays messages that require operator intervention.

The **nsrwatch** program will run continuously until quit, stopped, or interrupted (Control-Z or Control-C, for example). Typing the **q** character will quit the program, the Control-L character will force a screen clear and redraw, while any other character will force the status to be updated.

The **nsrwatch** program checks for new devices at a slower rate than the polling rate, so it might take up to a minute after a new device is added before it is noticed. Restarting the program, or typing Control-L, will notice the new device immediately. Deleted devices may cause a “resource does not exist” message temporarily, but otherwise they are noticed immediately.

The **nsrwatch** program will adapt to changes in the screen size, if supported by the underlying environment. For example, if a window terminal emulator is resized, the size of each field may change to match the window. If the window is too small, all the devices, sessions, messages, and so on, might not be displayed. For best results, use a window of at least 30 lines.

## Options

**-s** *server*

Set the current Legato Storage Manager server to *server*.

**-p** *polltime*

Set the polling interval to be *polltime* seconds.

**See Also:** [termcap\(5\)](#), [nsr\\_notification\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_service\(5\)](#), [recover\(8\)](#), [nsradmin\(8\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nwadmin\(8\)](#)

## nwadmin(8)

### Name

**nwadmin** – Graphical administration interface to Legato Storage Manager

### Synopsis

**nwadmin** [ *-s server* ]

### Description

**nwadmin** is an X Window System application. It is used to administer and monitor Legato Storage Manager servers.

The server's name may be specified with the *-s server* argument. When no server is specified, **nwadmin** uses the server selection rules found in **nsr(8)**. When multiple Legato Storage Manager servers are accessible, they may be selected from within the **nwadmin** command.

The **nwadmin** command is used to administer Legato Storage Manager servers. Clients may be added and deleted. Schedules controlling save levels may be created and modified. Groups of clients may be formed and controlled together. Directives controlling how data is saved may be defined and changed. Cloning of save sets and recover by save sets may be specified. Cloning of entire backup volumes may also be specified. The notification messages may be displayed. In general, there is a panel for each component.

The current state of Legato Storage Manager servers may be monitored. The amount of data saved, the number of clients saving, and requests for mounting and unmounting volumes, are among the items displayed. This information is displayed on the console panel.

The graphical interfaces for backup and recover are available through **nwbackup(8)** and **nwrecover(8)**, respectively.

A complete explanation of the **nwadmin** command may be found in the *Legato Storage Manager Administrator's Guide*.

### Options

*-s server*

Set the current Legato Storage Manager server to *server*.

## Files

**/usr/lib/X11/app-defaults/Networker**  
The X11 resources for **nwadmin**.

**See Also:** [nsr\(8\)](#), [nsradmin\(8\)](#), [nwbackup\(8\)](#), [nwrecover\(8\)](#), *Legato Storage Manager Administrator's Guide*

## nwarchive(8)

### Name

**nwarchive** – Legato Storage Manager graphical archive interface

### Synopsis

**nwarchive** [ *-s server* ]

### Description

**nwarchive** is an X Window System application. It is a front end to **nsrarchive(8)** and used to archive files to a Legato Storage Manager server on an as requested basis. Normally, files are archived automatically.

The server's name may be specified with the *-s server* argument. When no server is specified, **nwarchive** uses the server selection rules found in **nsr(8)**. When multiple Legato Storage Manager servers are accessible, they may be selected from within the **nwarchive** command.

Legato Storage Manager supports both scheduled network-wide archives and manual archives of client system files and directories. To request an immediate manual archive, run **nwarchive**.

Check that the correct Legato Storage Manager server is selected. The Server is identified in the Main window. You can change servers using the Change Server command if necessary. This is the Server to which the client files will be backed up. The hostname of the current client is displayed in the Client field. The path name of the current directory is displayed in the Selection field.

Change directories by entering the full path name in the Selection field or by highlighting the icon in the Archive window.

To perform a manual archive, first mark the files and directories that you want to back up by selecting their check boxes. Then select *Start archive...* from the File menu of the Archive window. You must enter an annotation for the archive.

Monitor the progress of the archive in the Archive Status window. Check to see that an archive volume is mounted in the Pending display of the Main window.

## Options

`-s server`

Set the current Legato Storage Manager server to *server*.

## Files

`/usr/lib/X11/app-defaults/Networker`

The X11 resources for **nwarchive**.

**See Also:** [nsr\(8\)](#), [nsradmin\(8\)](#), [nsrarchive\(8\)](#), [nsrretrieve\(8\)](#)

## nwbackup(8)

### Name

**nwbackup** – Legato Storage Manager graphical backup interface

### Synopsis

**nwbackup** [ *-s server* ] [ *path* ]

### Description

**nwbackup** is an X Window System application. It is a front end to **save(8)** and used to save files to a Legato Storage Manager server on an as requested basis. Normally, files are saved automatically.

The server's name may be specified with the *-s server* argument. When no server is specified, **nwbackup** uses the server selection rules found in **nsr(8)**. When multiple Legato Storage Manager servers are accessible, they may be selected from within the **nwbackup** command. If *path* is specified, **nwbackup** will set initialize the current selection to the given path. The default selection if *path* is not specified is the current working directory.

Legato Storage Manager supports both scheduled network-wide backups and manual backups of client system files and directories. To request an immediate manual backup, run **nwbackup**.

Check that the correct Legato Storage Manager server is selected. The Server is identified in the Main window. You can change servers using the Change Server command if necessary. This is the Server to which the client files will be backed up. The hostname of the current client is displayed in the Client field. The path name of the current directory is displayed in the Selection field.

Change directories by entering the full path name in the Selection field or by highlighting the icon in the Backup window.

To perform a manual backup, first mark the files and directories that you want to back up by selecting their check boxes. Then select *Start backup...* from the File menu of the Backup window. You must select whether to compress files or exclude patterns in the Backup Options dialog box to continue the backup.

Monitor the progress of the backup in the Backup Status window. Check to see that a backup volume is mounted in the Pending display of the Main window.

## Options

`-s server`

Set the current Legato Storage Manager server to *server*.

## Files

`/usr/lib/X11/app-defaults/Networker`

The X11 resources for **nwbackup**.

**See Also:** [nsr\(8\)](#), [nsradmin\(8\)](#), [nwbackup\(8\)](#), [nwrecover\(8\)](#), [save\(8\)](#)

## nwrecover(8)

### Name

**nwrecover** – Legato Storage Manager graphical recover interface

### Synopsis

**nwrecover** [ *-s server* ] [ *-c client* ] [ *-T browse time* ] [ *path* ]

### Description

**nwrecover** is an X Window System application. It is used to recover lost files that have been saved with Legato Storage Manager. If you are running in a non-X11 environment, **recover(8)** may be used to recover files.

The server's name may be specified with the *-s server* argument. When no server is specified, **nwrecover** uses the server selection rules found in **nsr(8)**. When multiple Legato Storage Manager servers are accessible, they may be selected from within the **nwrecover** command. If *path* is specified, **nwrecover** will attempt to initialize the current selection to the given path. The default attempted selection if *path* is not specified is the current working directory.

If you are recovering files that were saved with Access Control Lists (ACLs), you need to be root or the file owner to recover the file. Files with an ACL have a trailing '+' (for example, *-rw-r--r--+*) after the mode bits when viewing file details. See **recover(8)** for more information about ACLs.

There are three basic steps to recover a lost file: (1) Browse Legato Storage Manager's index in the Recover window to find the lost file, (2) Mark the file for recovery by selecting its check box, and (3) Start the recovery. In addition, there are recover commands for relocating recovered files (Relocate), finding past versions of a file (Versions), changing the browse time (Change Browse Time), showing the files you have marked for recovery (Show Marked), and overwriting or renaming recovered files that are in conflict with existing files (Conflict Resolution).

Opening the Recover window connects the client to its indexes maintained on the server. The entries in the index represent previously backed-up files and are organized exactly like the file system. To browse the index for another file system, enter the path name in the Selection field.

To browse the index: Use the Recover window View menu to select the browsing level of your directories. Use the mouse to open a directory and display its contents.

To mark files: After you have located your files by browsing the index, mark the files you want to recover by selecting their check boxes. Or, highlight a file and use the Mark command from the File menu to mark files. You can list the files that you have marked for recovery with the Show Marked command from the View menu.

To start the recovery: Select the Start recover command from the File menu. The Conflict Resolution dialog box appears, where you tell Legato Storage Manager what to do when a conflict occurs between a recovered file and an existing file. You select whether to be prompted for each individual conflict or to select one global resolution for all conflicts. Then you tell Legato Storage Manager whether to Rename the recover file with a .R extension to preserve both files, to Discard the recover file and preserve the existing file, or to Overwrite the existing file to preserve the recover file as the only copy of the file.

After you press OK in the Conflict Resolution dialog box, the recover continues. Legato Storage Manager will then automatically determine the media needed to complete the recovery, prompt the operator to mount the media, and start the recovery. You can monitor the status of the recovery in the Recover Status window.

Before starting the recovery, you have the option of relocating the recover files with the Relocate command. Enter the path name of a new or existing directory in which to place your recovered files.

The Recover window also offers two commands for browsing the index in the past. Versions shows you the entire backup history for a file. Change Browse Time enables you to change the time at which you are viewing the on-line index.

## Options

**-s** *server*

Set the current Legato Storage Manager server to *server*.

**-c** *client*

Set the current Legato Storage Manager client index to browse to *client*.

**-T** *browse time*

Set the current index browse time to *browse time* (in `nsr_getdate(3)` format). Using this option is equivalent to using the **change browse time** dialog within `nwrecover`.

## Files

`/usr/lib/X11/app-defaults/Networker`

The X11 resources for `nwrecover`.

**See Also:** [nsr\\_getdate\(3\)](#), [nsr\(8\)](#), [nsradmin\(8\)](#), [nwbackup\(8\)](#),  
[recover\(8\)](#)

---

## nwretrieve(8)

### Name

**nwretrieve** – Legato Storage Manager graphical retrieve interface

### Synopsis

**nwretrieve** [ *-s server* ]

### Description

**nwretrieve** is an X Window System application. It is used to retrieve files that have been archived with Legato Storage Manager. If you are running in a non-X11 environment, **nsrrretrieve(8)** may be used to retrieve files.

The server's name may be specified with the *-s server* argument. When no server is specified, **nwretrieve** uses the server selection rules found in **nsr(8)**. When multiple Legato Storage Manager servers are accessible, they may be selected from within the **nwretrieve** command.

If you are retrieving files that were archived with Access Control Lists (ACLs), you need to be in group operator or the file owner to retrieve the file. See **nsrrretrieve(8)** for more information about ACLs.

There are three basic steps to retrieve a lost file: (1) Browse Legato Storage Manager's list of Archives in the Retrieve window, (2) Select the Archive you want to retrieve, (3) Start the retrieve.

Opening the Retrieve window connects with the Legato Storage Manager server indexes. Selecting the Query button displays a list of archive available on the server. The entries in the list represent previously archived files.

To start the retrieve: Select the *Start retrieve* command from the File menu. The Retrieve Status dialog box appears, and you may enter a path to relocate to and select if you want to overwrite existing files.

After you press OK in the Retrieve Status dialog box, the Retrieve will begin retrieving the selected archives and status will be displayed in the Status field. Legato Storage Manager will then automatically determine the media needed to complete the retrieve, prompt the operator to mount the media, and start the retrieve.

## Options

**-s** *server*

Set the current Legato Storage Manager server to *server*.

## Files

**/usr/lib/X11/app-defaults/Networker**

The X11 resources for **nwretrieve**.

**See Also:** [nsr\(8\)](#), [nsradmin\(8\)](#), [nwarchive\(8\)](#), [nsrarchive\(8\)](#),  
[nsrrtrieve\(8\)](#)

## preclntsave(8)

### Name

**preclntsave** – Child process to run pre-processing commands for Legato Storage Manager **savenpc**.

### Synopsis

```
preclntsave -s server -c client -g group [-D debuglevel ]
```

### Description

The **preclntsave** process checks to see if there is an existing */nsr/res/<grpname>.tmp* file, which indicates that the pre-processing commands had been run. If so, it just simply exits with status 0 to let **savenpc** resume its normal save task. Otherwise, it locks the */nsr/res/<grpname>.lck* file, invokes all the pre-processing commands specified in the */nsr/res/<grpname>.res* file, then creates */nsr/res/<grpname>.tmp* file, finally spawns the **pstclntsave** process and exits with status 0.

**Note:** This is to be invoked by **savenpc** program only. It is not meant for users to use.

### Options

**-s** *server*

Specify the controlling server.

**-c** *client*

The name of the client where the pre-processing commands will be performed on.

**-g** *group*

Specify the group name that is being run.

**-D** *debuglevel*

For debugging purpose, the debuglevel could be 1, 2 or 3.

**See Also:** [pstclntsave\(8\)](#), [save\(8\)](#)

## pstclntsave(8)

### Name

**pstclntsave** – Child process of **preclntsave** to run post-processing commands for Legato Storage Manager **savenpc**.

### Synopsis

```
pstclntsave -s server -c client -g group [ -p pollinterval ] [ -t timeout ] [ -D debuglevel ]
```

### Description

The **pstclntsave** process keeps checking the WORKLIST attribute of the CLIENT resource from the server every number of seconds specified in the poll interval. Whenever the time\_out condition or the WORKLIST is NIL (whichever comes first) **pstclntsave** then performs all the post-processing commands specified in */nsr/res/<grpname>.res* file, unlinks */nsr/res/<grpname>.tmp* and */nsr/res/<grpname>.lck*, then records the results (success or failure) in */nsr/log/savenpc.log* file.

**Note:** This is to be invoked by **preclntsave** program only. It is not meant for users to use.

### Options

**-s** *server*

Specify the controlling server.

**-c** *client*

The name of the client where the pre-processing commands will be performed on.

**-g** *group*

Specify the group name that is being run.

**-p** *pollinterval*

How often (in seconds) to poll the server.

**-t** *timeout*

The timeout condition in **nsr\_getdate**(3) format string to start the post-processing commands. This can also be specified in the */nsr/res/<grpname>.res* file.

**-D** *debuglevel*

For debugging purpose, the debuglevel could be 1, 2 or 3.

**See Also:** [preclntsave\(8\)](#), [save\(8\)](#)

## recover(8)

### Name

**recover** – Browse and recover Legato Storage Manager files

### Synopsis

```
recover [ -f ] [ -n ] [ -q ] [ -u ] [ -i {nNyYrR} ] [ -d destination ] [ -c client ] [ -t date ]  
[ -s server ] [ dir ]
```

```
recover [ -f ] [ -n ] [ -u ] [ -q ] [ -i {nNyYrR} ] [ -I input file ] [ -d destination ]  
[ -c client ] [ -t date ] [ -s server ] -a path...
```

```
recover [ -f ] [ -n ] [ -u ] [ -q ] [ -i {nNyYrR} ] [ -d destination ] -s server  
-S ssid/cloneid [ -S ssid/cloneid ] ... [ path ] ...
```

```
recover [ -f ] [ -n ] [ -q ] [ -i {NYR} ] -R recover-target -c client [ -d destination ]  
[ -t date ] [ -s server ] [ dir ]
```

### Description

**Recover** browses the saved file index and recovers selected files from the Legato Storage Manager system. The file index is created when files are saved with **save(8)**. When in interactive mode (the default), the user is presented with a view of the index similar to a UNIX file system, and may move through the index to select and recover files or entire directories. In automatic mode (**-a** option), the files specified on the command line are recovered immediately and no browsing takes place. While in save set recover mode (**-S** option), the save set(s) specified are retrieved directly without browsing the Legato Storage Manager file index. Use of save set recover mode is restricted to root.

When using **recover** without the **-S** option, users who are root may recover any file. The remaining permission checking rules described in the paragraph apply to users who are not root. For files that don't have an Access Control List (ACL), the normal UNIX mode bits must allow you to read the file in order to recover it. Files with an ACL can only be recovered by their owner or by root.

### Options

**-a**

This option specifies automatic file recovery with no interactive browsing. *Path* specifies one or more files or directories to be recovered.

**-S *ssid*[/*cloneid*]**

This option is used to specify save set recover mode and can only be used by root. This mode can be used to implement fast batch file recovery without requiring the Legato Storage Manager file index entries. *Ssid* specifies the save set id's for the save set(s) to be recovered. When there are multiple clone instances for a save set, the *cloneid* can also be specified to select the particular clone instance to be recovered from. When no *path* arguments are specified, the entire save set contents will be recovered. One more or more *path*'s can be specified to limit which directories and files are actually recovered. If *path*'s are supplied, then the beginning of each path name as it exists in the save set must exactly match one of the *path*'s before it will be recovered. Shell-like filename matching using meta characters like '\*', '?', and '['...' is not performed. You can use a *path* that ends in with a slash ('/') to force a directory only match (for example, use a *path* of */etc/fs/* instead of */etc/fs* to prevent files like */etc/fsck* from being recovered as well).

**-d *destination***

Specifies the destination directory to relocate recovered files to. Using this option is equivalent to using the **relocate** command when in interactive mode (discussed as follows). Relative paths are interpreted relative to the current working directory.

**-s *server***

Selects which Legato Storage Manager server to use. This option is required for save set recover mode (**-S**).

**-c *client***

*Client* is the name of the computer that saved the files. Note that when browsing a directory that was saved by another client, the path names will reflect the file tree of the client that saved the files. By default **save** and **recover** determine the client name from the file system table, but this option might be necessary if the **-L** option was used on the **save** command. This option cannot be used in conjunction with the **-S *ssid*** option (save set recover mode).

**-t *date***

Display/recover files as of the specified *date* (in **nsr\_getdate(3)** format). Using this option is equivalent to using the **changetime** command with the given *date* when in interactive mode (discussed as follows). This option cannot be used in conjunction with the **-S *ssid*** option (save set recover mode).

**-q**

The **recover** command normally runs with verbose output. This flag turns off the verbose output.

**-f**

Force recovered files to overwrite any existing files whenever a name conflict occurs. This is the same as specifying **-iY**.

**-n**

When recovering, do not actually create any directories.

**-i {nNyYrR}**

Specifies the initial default overwrite response to use when recovering files and the file already exists. Only one letter may be specified. This option is the same as the **uasm -i** option when running in recover mode. See the **uasm(8)** man page for a detailed explanation of this option. For directed recovers (see the **-R** flag), only 'N', 'Y', and 'R' "are" "valid" "values."

**-I input file**

In addition to taking the paths to recover from the command line, read paths to recover from the named file. The paths must be listed one on each line. If no paths are specified on the command line, then only those paths specified in the file will be recovered. To be used in conjunction with **-a** option.

**-R recover-target**

This specifies the name of the remote computer to direct the recovery to. This is used in conjunction with the **-c** option to specify browsing of another client's index. When the **-R** option is used, either the **-f** or the **-i** option must also be specified in order to instruct the recover target what to do when it is recovering files and the file already exists. Note that the values 'N', 'Y', and 'R' are the only valid ones to use with the **-i** flag for directed recovers.

**-u**

Stop when an error occurs during recovery. Normally, recover treats errors as warnings and tries to continue to recover the rest of the files requested. However, when this flag is used, recover will stop recovering on the first error it encounters. This option is not valid for directed recovers.

## Usage

When using recover in the interactive mode, an image of the file system at a particular time is presented. Using commands similar to the shell, one can change the view and traverse the file system. Files may be selected for recovering, and the actual recover command issued.

The following commands manipulate the view of the file system and build the list of files to recover. In all of the commands that take a *name* argument pattern

matching characters can be used. The pattern matching characters and regular expression format are the same as for the UNIX shell **sh**(1).

**ls** [ *options* ] [ *name ...* ]

List information about the given files and directories. When no *name* arguments are given, **ls** lists the contents of the current directory. When a *name* is given and *name* is a directory, its contents are displayed. If *name* is a file, then just that file is displayed. The current directory is represented by a '.' (period). The options to this command correspond to those of the UNIX command, **ls**(1). An additional recover specific **-S** option can be used to select the save time instead of the last modified time for sorting (with the **-t** option) and/or printing (with the **-l** option). Files that have been added to the recover list are preceded by a '+'. Files that have an ACL have a trailing '+' (for example, -rw-r--r--+ ) after the mode bits when viewing file details.

**If** [ *name ...* ]

is the same as **ls -F**. Directories are marked with a trailing '/', symbolic links with a trailing '@', sockets with a trailing '=', FIFO special files with a trailing '|', and executable files with a trailing '\*'.

**ll** [ *name ...* ]

is the same as **ls -lgsF**. Generates a long format listing of files and directories. This command can be used to find the value of a symbolic link.

**cd** [ *directory* ]

Change the current working directory to *directory*. The default directory is the directory **recover** was executed in. If *directory* is a simple symbolic link, **cd** will follow the symbolic link. However, if *directory* is a path containing symbolic links anywhere but at the end of the path, the **cd** command will fail; you should **cd** a component of the path at a time instead.

**pwd**

Print the full path name of the current working directory.

**add** [ *name ...* ]

Add the current directory, or the named file(s) or directory(s) to the recover list. If a directory is specified, it and all of its descendent files are added to the recover list.

**delete** [ *name ...* ]

Delete the current directory, or the named file(s) or directory(s) from the recover list. If a directory is specified, that directory and all its descendents are deleted from the list. The most expedient way to recover a majority of files from a directory is to add the directory to the recover list, and then delete the unwanted files.

**list** [ *-l* ] | [ *-c* ]

Display the files on the recover list. With no arguments the recover list is displayed as a list of full path names, one on each line, followed but a total count of the files to be recovered. The *-c* argument prints just the total count of files to be recovered. The *-l* argument prints the files in the same format as the **ll** command with the *-dS* options.

**volumes**

Prints a list of the volumes need to recover the current set of files on the recover list.

**recover**

Recover all of the files on the recover list from the Legato Storage Manager server. Upon completion the recover list is empty.

**verbose**

Toggle the status of the “verbose” option. When verbose mode is on **recover** displays information about each file as it is recovered. When verbose mode is off **recover** only prints information when a problem occurs. The default is verbose mode on.

**force**

If name conflicts exist, overwrite any existing files with recovered files.

**noforce**

Cancel the **force** option. When in ‘noforce’ mode, a prompt is issued each time a naming conflict arises between a file being recovered and an existing file. At each prompt, six choices are presented: ‘y’, ‘Y’, ‘n’, ‘N’, ‘r’ and ‘R’. To overwrite the existing file, select ‘y’. To rename the file to an automatically generated alternative name, select ‘r’. Selecting ‘n’ causes the recovered file to be discarded. The capital letters cause the same action for all subsequent conflicts without further prompting. Hence, selecting ‘Y’ will cause all existing conflicting files to be overwritten, ‘N’ will cause all conflicting recovered files to be discarded, and ‘R’ will automatically rename all conflicting recovered files (except when an external ASM has a conflicting filename that already ends in the rename suffix).

**relocate** [ *directory* ]

Change the target recover location to *directory*, if *directory* is not specified then the user will be prompted for a destination directory. Relative paths are interpreted relative to the current working directory within the **recover** program. The recovered files will be placed into this directory, which will be created if necessary. When files from multiple directories are being recovered, they will be placed below this directory with a path relative to the first common parent of all the files to be recovered. For example, if */usr/include/sys/errno.h* and */usr/include/stdio.h* are being

recovered, and the relocation directory is set to */tmp*, then the first common parent of these two files is *include*, so the recovered files will be named */tmp/sys/errno.h*, and */tmp/stdio.h*.

### **destination**

Print destination location for recovered file.

### **exit**

Immediately exit from **recover**.

### **help**

Display a summary of the available commands.

### **?**

Same as **help**.

### **quit**

Immediately exit from **recover**. Files on the recover list are not recovered.

### **changetime** [ *time* ]

Display the file system as it existed at a different time. If no *time* is specified the 'current' time is displayed, and a prompt is issued for a 'new' time. The new time is given in **nsr\_getdate**(3) format. This format is very flexible. It accepts absolute dates, such as *March 17, 1999*, and relative dates, such as *last Tuesday*. Absolute dates can be given in two formats: *MM/DD/YY*, and *Month DD[, YYYY]*. Times can also be specified as either absolute or relative, with absolute times in the format: *HH[:MM][:SS]] [am | pm] [time zone]*. For example, 12:30 am, 14:21, and 10 pm PST. The current time is used to calculate unspecified parts of a relative date (for example, 2 days ago means 2 days ago at the current time), and the end of the day is assumed for unspecified times on an absolute date (for example, July 2 means July 2 at 11:59:59 PM). By default, the present is used as the current time. The resolution of the file system image at a time in the past depends on how often **save** was run and how far back the Legato Storage Manager file index information goes.

### **versions** [ *name* ]

All instances of the current directory, if *name* is not specified, or the named file or directory, found in the Legato Storage Manager file index are listed. For each instance, three lines of data are displayed. The first line is similar to the **ll** output. The second line lists the instance's save time. The third line specifies which tape(s) this instance may be recovered from. With appropriate use of the **changetime** command, any one of the entries may be added to the recover list. As with **ls**, **If**, and **ll**, files that have been added to the recover list are preceded by a '+'.

**See Also:** **ls**(1), **nsr\_getdate**(3), **nsr\_service**(5), **nsr**(8), **nsrd**(8), **nsrindexd**(8), **nwrecover**(8), **save**(8)

## Diagnostics

**Recover** complains about bad option characters by printing a “usage” message describing the available options.

**Message from server: other clones exist for failed save set**

Recover will automatically re-submit its recover request to the server, if any files remain to be recovered, because the request failed on a save set that had multiple clones. The server automatically picks a different clone on each attempt.

**Path name is within *machine:export-point***

An informative message that lets you know that the given path name is mounted from a network file server and that the recovery will use the index for the named file server. If the *computer* is not a Legato Storage Manager client, then the `-c` option may be necessary.

**Browsing *machine's* on-line file index**

An informative message that explicitly states which Legato Storage Manager client's index is being browsed for interactive recovers which resolve to another computer.

**Using *server* as server for *client***

An informative message that lets you know which Legato Storage Manager server was selected for client's index.

**Cannot open recover session with *server***

This message indicates that some problem was encountered connecting to the Legato Storage Manager server on the named computer.

**error, *name* is not on client list**

This message indicates that the client invoking the **recover** command is not in the server's client list. See **nsr\_service(5)** for details.

***path*: Permission denied**

The filename cannot be recovered because you are not root, and you don't have read permission for the file.

***path*: Permission denied (has acl)**

The filename cannot be recovered because you are not root, the file has an ACL (Access Control List), and you are not the owner of the file.

## save(8)

### Name

**save** – Save files to long term storage with Legato Storage Manager

**savenpc** – Save files to long term storage with Legato Storage Manager and performs pre- and post-processing commands on a Legato Storage Manager client.

### Synopsis

```
save [ -BEiKLnquSVvx ] [ -s server ] [ -c client-name ] [ -N name ] [ -e expiration ]
[ -f dirfile ] [ -b pool ] [ -F file ] [ -I input_file ] [ -g group ] [ -l level ] [ -t date ]
[ -m masquerade ] [ -W width ] [ path ... ]
```

```
savenpc [ -BEiKLnquSVvx ] [ -s server ] [ -c client-name ] [ -N name ]
[ -e expiration ] [ -f dirfile ] [ -b pool ] [ -F file ] [ -I input_file ] [ -g group ] [ -l level ]
[ -t date ] [ -m masquerade ] [ -W width ] [ path ... ]
```

### Description

**save** saves files, including directories or entire file systems, to the Legato Storage Manager server (see **nsr(8)**). The progress of a save can be monitored using the X Window System based **nwadmin(8)** program or the **curses(3X)** based **nsrwatch(8)** program for other terminal types.

If no *path* arguments are specified on the command line or with the **-I** option, the current directory will be saved. **save** will save a directory by saving all the files and subdirectories it contains, but it will not cross mount points, nor will it follow symbolic links. If the paths to be saved are mounted from a network file server, **save** will instruct the user to run the save on the remote computer or use the **-L** option.

The directive files (see **nsr(5)**) encountered in each directory will be read by default, and they contain special instructions directing how particular files are to be saved (compressed, skipped, and so on). These files are named ``.nsr'`."`

Each file in the subdirectory structures specified by the *path* arguments will be encapsulated in a Legato Storage Manager save stream. This stream of data is sent to a receiving process (see **nsrd(8)**) on the Legato Storage Manager server, which will process the data, adding entries to the on-line index (see **nsrindexd(8)**) for each file in the stream, with the data finally ending up on some long term storage media (see **nsrmmmd(8)**).

Details about handling media are discussed in **nsrmm(8)** and **nsr\_device(5)**.

**savenpc** consists of the same command options as **save** and behaves just like **save**. In addition, prior to the actual save on a Legato Storage Manager client, **savenpc** performs pre-processing commands if any exists in the `/nsr/res/<grpname>.res` file, and at the end of the save of the last save set on the client, the post-processing commands (if any) will be invoked. In the condition of failure to run the pre-processing commands, **savenpc** aborts itself. All results are logged in `/nsr/logs/savenpc.log` file on the client. A timeout condition can be set by the user to indicate at which point in time the post-processing commands need to be run without waiting for all the save sets to be backed up. This timeout attribute resides in the `/nsr/res/<grpname>.res` file. The timeout should be specified in such a format that **nsr\_getdate()** can understand (see **nsr\_getdate(3)**).

An example of `/nsr/res/<grpname>.res` can be described as:

```
type:          savenpc;
precmd:        /bin/true;
pstcmd:        /bin/true, "/bin/sleep 5";
timeout:       "12:00pm";
```

The `precmd` field can be manually modified to contain any number of commands that are needed to be run at the beginning of the save of the 1st save set. The `pstcmd` is to hold any commands that are needed to be run at the end of the save of the last save set. The post-processing commands will be run after the save of the last save set or the timeout condition, whichever comes first.

## Options

**-E**

Estimate the amount of data which will be generated by the save, then perform the actual save. Note that the estimate is generated from the inode information, and thus the data is only actually read once.

**-I *input\_file***

In addition to taking the paths to save from the command line, read paths to save from the named file. The paths must be listed one on each line. If no paths are specified on the command line, then only those paths specified in the file will be saved.

**-i**

Ignore any `.nsr` directive files as they are encountered in the subdirectory structures being saved.

**-K**

Do not build connecting directory index entries.

**-L**

Local. Saves will be performed from the local Legato Storage Manager client, even when files are from a network file server. To recover these files, run **recover(8)** with the **-c client** arguments, where *client* is the name of the Legato Storage Manager client that did the **save**.

**-LL**

In addition to treating the backup as a local backup, cause an extra line to be printed at the end of the completion output of the form “complete savetime=number”, where number is the savetime of the save set created by this backup. This option is meant to be used by the **savegrp(8)** command in performing automatic cloning.

**-m masquerade**

Specifies the tag to precede the summary line with. This option is used by **savegrp(8)** and **savefs(8)** to aid in **savegrp** summary notifications.

**-n**

No save. Estimate the amount of data which will be generated by the save, but do not perform the actual save.

**-v**

Verbose. Cause the **save** program to tell you in great detail what it is doing as it proceeds.

**-q**

Quiet. Display only summary information and error messages.

**-S**

Allow only save set recovery. This performs the save without creating any index entries for it. This means that the save set will not be browsable, although save set recovery may be used to recover the data.

**-s server**

Specify which computer to use as the Legato Storage Manager server.

**-c client-name**

Specify the client name for starting the save session. This is useful on clients with multiple network interfaces, and hence multiple host names. It can be used to create multiple index databases for the same physical client. Note that this does not specify the network interface to use. This is specified in the **server network interface** attribute of the client resource (see **nsr\_client(5)**).

**-N name**

The symbolic name of this save set. By default, the most common prefix of the *path* arguments is used as the save set name.

**-e expiration**

Sets the date (in **nsr\_getdate**(3) format) when this save set will expire. When a save set has an explicit expiration date, the save set remains both browsable and non-recyclable until it expires. After it expires and it has passed its browse time, its state will become non-browsable. If it has expired and it has passed its retention time, the save set will become recyclable. By default, no explicit save set expiration date is used.

**-f dirfile**

The file from which to read prototype default directives (see **nsr**(5)). A *dirfile* of - causes the default directives to be read from standard input.

**-b pool**

Specifies a particular destination pool for the save.

**-F file**

Only save files whose change time is newer than the file modification date of *file*.

**-g group**

This option is used by **savegrp**(8) and **savefs**(8) to denote the group of the save (see **nsr\_client**(5) and **nsr\_group**(5)) and is used by the Legato Storage Manager server to select the specific media pool.

**-l level**

The level of the save. This option is used by **savegrp**(8) and **savefs**(8) to specify a particular level for a scheduled save.

**-t date**

The date (in **nsr\_getdate**(3) format) after which files must have been modified before they will be saved. This option is used by **savegrp**(8) and **savefs**(8) to perform scheduled saves by consulting with the media database to determine the appropriate time value based on the previous saves for the save set and the level of the scheduled save.

**-W width**

The width used when formatting summary information output.

**-x**

Cross mount points.

**-B**

Force save of all connecting directory information from root (“/”) down to the point of invocation.

**-u**

Stop the save if an error occurs. The **save** program normally treats errors as warnings and continues to save the rest of the files in the backup. When this option is set, errors will cause **save** to exit and stop the save. This option is not recommended for general use, although it can be useful when a group of files must be backed up as a set.

**-V**

Prevent the OFC mechanism from creating a point-in-time copy of the source volume. (Included for compatibility with NT Legato Storage Manager servers.)

**See Also:** [curses\(3X\)](#), [nsr\\_getdate\(3\)](#), [nwadmin\(8\)](#), [nsr\(5\)](#), [nsr\(8\)](#), [nsr\\_client\(5\)](#), [nsr\\_device\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_service\(5\)](#), [nsrd\(8\)](#), [nsrim\(8\)](#), [nsrindexd\(8\)](#), [nsrmm\(8\)](#), [nsrmmmd\(8\)](#), [nsrwatch\(8\)](#), [recover\(8\)](#), [savefs\(8\)](#), [savegrp\(8\)](#)

## Diagnostics

*Exit codes:*

- 0** Normal exit. This means that a save set was correctly created on the server. Messages about individual file backup failures are *warnings*, and do not cause abnormal exit.
- <>0** Abnormal exit. A save set was not correctly created on the server.

*Messages:*

*host: saveset level=level, size time count files.*

This message (with the appropriate client host name, saveset name, level, total save set size, elapsed time, and file count) is printed whenever **save** is run by **savegrp(8)** and exits normally.

*host: filename: warning*

Messages of this form are warnings about difficulties backing up individual files. Such messages do not normally cause the **save** to fail, and therefore may appear in the **save** output found in the “Savegroup Completion” message’s Successful section.

## savefs(8)

### Name

**savefs** – Save file system to a Legato Storage Manager server

### Synopsis

**savefs** [ *options* ] *file system*

**savefs** **-p** [ *options* ] [ *file system ...* ] [ **-M** *file system ...* ]

*options:*

[ **-BEFnpqRv** ] [ **-s** *server* ] [ **-N** *name* ] [ **-g** *group* ] [ **-c** *client* ] [ **-l** *level* | **-C** *schedule* ] [ **-e** *expiration* ] [ **-f** *filename* ] [ **-W** *width* ] [ **-t** *date* ] [ **-T** *seconds* ]

### Description

The **savefs** command will save a file system (using **save(8)**) to a Legato Storage Manager server. Mount points are not crossed, and symbolic links are not followed. **Note:** Running **savefs** directly is not recommended; use **savegrp(8)** instead.

A *level*-based system (similar to **dump(8)**) is used to save only those files which have been modified since some previous save (a *partial* save).

The **nsr\_schedule(5)** for the local Legato Storage Manager client will be examined to determine the proper level of save for the current date.

The set of files that actually get saved will depend on when, and at what level, previous saves have been performed, in addition to the effects of the default directives (see **nsr\_directive(5)**), and the various directive files (see **nsr(5)**) which are encountered while processing the file system.

### File System Probes

The **savefs** command may also be used to probe a client for its file systems and recent save times. When probing, **savefs** does not actually save data, but instead produces a parsable report describing the layout of the client's file systems. When used with the **-p** probe option, the local Legato Storage Manager client's **nsr\_client(5)** resources will be examined, and the file systems listed in the *save set* attribute will be probed (if no file systems are listed on the command line). If the *save set* list consists of the keyword *All*, then the **/etc/fstab** file (**/etc/vfstab** on Solaris, **/etc/mnttab** on SCO, and a kernel table on AIX) will be examined to

determine which file systems should be saved, making sure to save only local, mounted file systems.

Note that metadevices within the Sun Solaris Online DiskSuite and Logical Volumes within the HP-UX Logical Volume Manager will be treated similar to independent disks. This approach allows each to be saved in its own session, assuming sufficient parallelism.

Care should be taken when the **NSR client** resource explicitly lists the save sets, for two primary reasons. First, this list must be manually updated when new file systems are added which need saving. Second, since **savefs** only stops at the end of a path or a mount point, if you list two save sets in the same file system, and one is a subdirectory of the other, the subdirectory will be saved twice.

*File System* arguments can be specified to limit the file system saves to only those specified, but the specified file systems must appear on some **Save Set** list for this client (see the **-F** option).

## Options

### **-B**

Force save of all connecting directory information from root (“/”) down to the point of invocation. This option is used by **savegrp(8)**, for example, when saving the server’s bootstrap information.

### **-c client**

The name of the client whose file system must be saved. This option is especially needed in a cluster environment where a physical host can represent its own hostname as well as hostnames of any virtual (also known as “logical”) hosts that exist in this physical host. Without this option, the hostname of the physical host is assumed by default. This option is required if a file system that belongs to any of the virtual hosts must be saved.

### **-C schedule**

The name of the schedule (see **nsr\_schedule(5)**) to use when automatically determining the save level. If this option is not specified, **savefs** will use the schedule named by the NSR client resource for the specified file system.

### **-e expiration**

Sets the date (in **nsr\_getdate(3)** format) when the saved data will expire. By default, no explicit expiration date is used.

### **-E**

Estimate. Before saving any data, walk the file system trees to be saved and accurately estimate the amount of data which will be generated. Without this flag,

the estimate size is zero. Note that this flag will consume an amount of time proportional to the number of files in each file system. This is because the entire directory is walked before any saving begins and walked again when actually saving the directory, but the file data is only read from the disk the last time. In many cases, the overhead for using this flag is small and is well-justified.

**-f filename**

The file from which application specific modules (or ASMs) should take their directives (see **nsr(5)**). By default, these are taken from the **NSR directive** resource named by the *directive* attribute in the **NSR client** resource for each client (see **nsr\_directive(5)**).

**-F**

Force. Save every argument like a file system, even if they are not listed in **fstab(5)** nor **nsr\_client(5)**.

**-M**

As part of a probe, signifies that all subsequent file systems should be probed for their ability to be migrated. This option is quietly ignored on systems that do not support file migration.

**-g group**

Restrict the scope of the client to a particular group. If this option is not specified, save sets from all instances of the NSR client resource for this client will be used, regardless of the group. This value is also passed on to **save(8)**, which uses it to select a specific media pool.

**-l level**

The level of save to perform. There are 12 levels: **full**, levels **1** through **9**, **incr**, and **skip**. **Full** specifies that all files are to be saved. It is analogous to a level 0 dump in **dump(8)**. **Incr** specifies incremental saves in which only those files that have been modified since the most recent save, at any level, are saved. This level has no exact analogue in **dump(8)** since the last save at any level, including previous incremental saves, are considered when determining what to save. **Skip** causes no files to be saved. The levels **1** through **9** cause all files to be saved which have been modified since any *lower* level save was performed. As an example, if you did a full on Monday, followed by a level 3 save on Tuesday, a subsequent level 3 save on Wednesday would contain all files modified or added since the Monday full save. By default, the save level is determined automatically from the Legato Storage Manager client's schedule (see **nsr\_schedule(5)**). By using the history of previous saves maintained by **nsrmmmd(8)** on the Legato Storage Manager server, the needed time for the given level can correctly be computed. By using media information on the server, times computed for saves which are based on previous save levels will automatically be adjusted as required when tapes are deleted.

**-n**

No save. Accurately estimate the amount of data which would be generated (as described for **-E**, but don't actually save any data.

**-N name**

The symbolic name this set of saves is to be known by. By default, the first *file system* argument is used as the name.

**-p**

List the name of the file systems, the level of save that would be performed, and the time since which files must have been modified to be saved, but don't actually do the save. This information is gleaned from the */etc/fstab* file (or another operating system specific file, as described previously) and the **nsr\_schedule(5)**.

**-q**

Quiet. Display only summary information and error messages.

**-qq**

Really quiet. Display only error messages.

**-R**

Cause **savefs** to report on its success or failure, by echoing a simple "succeeded" or "failed" message as its last act. This is used by **savegrp(8)** when it is running **savefs**.

**-s server**

Specify which computer to use as the Legato Storage Manager server. See **nsr(8)** for the algorithm Legato Storage Manager uses to choose a server when none is specified.

**-t date**

The date (in **nsr\_getdate(3)** format) from which to base schedule level calculations. If not specified, the current time is used.

**-T seconds**

This specifies the "inactivity timeout" in seconds. If **savefs** detects that the (local) server has made no progress in the specified time, then it concludes that the **save** command is hung. A message is printed to *stderr* and **savefs** exits normally. This option should only be used on Legato Storage Manager server computers.

**-v**

Verbose. Cause lots of debugging style output. This option is also used by **savegrp(8)** when it is probing for the capabilities of the client's **savefs**, for supporting multiple versions.

**-W** *width*

The width used when formatting output or notification messages. By default, this is 80.

## Resource Types

### NSR client

These resources specify the client's save sets, and the default schedule and directives to use when saving them.

### NSR directive

A resource of this type is named by the *directive* attribute in each **NSR client** resource. These are the directives used for the save sets specified in the associated **NSR client** resource.

### NSR schedule

A resource of this type is named by the *schedule* attribute in each **NSR client** resource. This is the schedule used for the save sets specified in the associated **NSR client** resource.

## Files

### **/etc/fstab**

If **All** is specified in the *save set* attribute for a **NSR client** resource, then the list of local file systems is taken from this file.

### **/etc/vfstab**

Solaris only. The same as */etc/fstab* on other operating systems.

### **/etc/mnttab**

SCO only. The same as */etc/fstab* on other operating systems.

**See Also:** [nsr\\_getdate\(3\)](#), [fstab\(5\)](#), [mnttab\(F\)](#) (SCO only), [vfstab\(5\)](#) (Solaris only), [nsr\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_schedule\(5\)](#), [dump\(8\)](#), [nsr\(8\)](#), [nsrd\(8\)](#), [nsrindexd\(8\)](#), [nsrmmd\(8\)](#), [recover\(8\)](#), [save\(8\)](#), [savegrp\(8\)](#)

## Diagnostics

Exit codes:

**0** Normal exit.

**255** Abnormal exit.

## savegrp(8)

### Name

**savegrp** – Start a group of Legato Storage Manager clients saving their file systems

### Synopsis

**savegrp** [ *options* ] [ **-R** | **-G** ] [ *groupname* ]

*options:*

[ **-EIOFXmnpv** ] [ **-l level** | **-C schedule** ] [ **-e expiration** ] [ **-t date** ] [ **-r retries** ]  
 [ **-P printer** ] [ **-W width** ] [ **-c client** [ **-c client ...** ] ]

### Description

The **savegrp** command runs a group of Legato Storage Manager clients through the process of saving their file systems (using **save(8)**). The group of clients is selected by naming a Legato Storage Manager group (see **nsr\_group(5)**), from which individual clients can be selected by using one or more **-c** options. If no group name is specified, the Legato Storage Manager group **Default** is used. If a Legato Storage Manager group is named, clients whose **nsr\_client(5)** resources specify the named group in their *group* attribute will be saved. If an explicit client list is also specified, **savegrp** will only back up those clients, with respect to the named group. The **savegrp** command will automatically make a clone of the newly saved data when the appropriate attributes are set on the **NSR group** resource (see the following description).

The **savegrp** command is normally run automatically by **nsrd(8)**, as specified by each group's **nsr\_group(5)** resource.

The **savegrp** command will set up an RPC connection to **nsrexecd(8)** to run **save(8)** on each client (and will fall back on using the **rcmd(3)** protocol and the client-side **rshd(8)** if **nsrexecd** is unavailable on the client) for each file system listed in the **nsr\_client(5)** resource *save set* attribute. If a save set of *All* is specified for a client, **savegrp** will request from the client a list of file systems to be saved (this is called the *probe* operation). The probe expands *All* into a list by looking for file systems that are both local and automatically mounted on that client computer (for example, NFS mount points and file systems mounted manually are generally ignored). The exact determination of which file systems to save will vary between different operating systems. See **savefs(8)** for additional details on the probe operation. To see which file systems a client will save run a **savegrp** preview, **savegrp -c client -p**

(assuming the client is in the Default group). Each file system saved is called a *save set*.

The **savegrp** command attempts to keep multiple clients busy by individually scheduling the client save sets. As save sets complete, the output is collected and another save set will be started by **savegrp**.

The **parallelism** attribute in the **nsr\_service(5)** resource is the maximum number of save sets to run simultaneously. Modifications to this parameter will take effect as save sets complete – if the value is reduced, no new save set will be started until the number of active save sets running drops below the new value.

When all of the save sets are completed on a client, the client's index on the Legato Storage Manager server will be saved. If the Legato Storage Manager server is one of the computers being saved, its index will be saved after all the other clients are completely finished. When the server's index is saved, the bootstrap save set information will be printed to the default printer (or another specified printer). If **savegrp** detects that the Legato Storage Manager server is not listed in any active group (a group with its *autostart* attribute set), then the server's index and bootstrap will be saved with every group.

The **savegrp** command will detect other active invocations of the same group, and will exit with an error message. If two different Legato Storage Manager groups are running simultaneously, they each will run up to *parallelism* save sessions simultaneously, however, the Legato Storage Manager server will only allow *parallelism* of these sessions to write to the backup devices at a time. Note that running multiple **savegrp** commands simultaneously can use up significant server resources, due to the number of pending saves.

The progress of the actively saving clients can be monitored using the X11 based **nwadmin(8)** program or the **curses(3X)** based **nsrwatch(8)** program. The **nsradmin(8)** browser may also be used to examine the *completion* status and *work list* of each *NSR group* resource, although the *hidden* attribute display option will need to be selected (see **nsradmin(8)**). These two attributes enable you to track the progress of each **savegrp**. See **nsr\_group(5)** for more details.

When **savegrp** starts, it sends an **NSR notification** (see **nsr\_notification(5)**) with an event of **savegrp** and priority of **info** to the **NSR notification** system. This event is normally logged in the *messages* attribute of the **nsr\_service(5)** resource, and also logged in the log file specified in the *Log default NSR notification* resource.

When all the save sets have finished, the save sets are automatically cloned, if the **NSR group** resource has the **clones** attribute enabled. The client save sets and their indexes are cloned before the bootstrap save set is generated so the bootstrap information can track both the original set of save sets and their clones. The

bootstrap save set is also cloned. Clones will be sent to the pool named in the **clone pool** attribute. Changing the values of these attributes while **savegrp** is running has no effect; they must be set before **savegrp** starts. The **nsrclone(8)** command is used to clone the save sets. Since **savegrp** uses a heuristic to determine which save sets were generated as part of the group, it may occasionally clone more save sets than expected, if a client has its file systems separated into multiple groups that run at the same time. Note that at least two enabled devices are required to clone save sets.

When the save sets are all complete and cloned (if cloning is enabled), an **NSR notification** with an event of **savegrp** and priority of **notice** is sent to the **NSR notification** system. This is generally set up to cause e-mail to be sent to the **root** user specifying the list of clients who failed (if any), and all the output collected from all clients. The format and common error messages included in the **savegrp** notification are explained in the **Savegroup Completion Notification Message** section, as follows.

## Options

**-E**

Cause **save(8)** on each client to estimate the amount of data which will be generated by each save set before performing it. This will result in the file system trees being walked twice – once to generate a estimate of how much data would be generated, and again to generate a save stream to the Legato Storage Manager server. Note that the data is only read from the disk on the final file system walk, as the estimate is performed by using inode information.

**-I**

Disable the saving of each client's index.

**-O**

Only save each client's index (for the server, the bootstrap is also saved).

**-m**

Disable monitor status reporting, including all **NSR notification** actions.

**-n**

No save. Cause **save** to perform an estimate as described for **-E**, but not to perform any actual saves. This option also sets **-m**.

**-p**

Run the probe step on each client, so you can see which file system would be saved and at what level, but do not actually save any data. This option also sets **-m**. The output generated by the **-p** option may show several save levels for each save set at different points in the output, as **savegrp** learns the correct level. This is the

expected behavior, and can be useful for debugging. The actual level the **savegrp** will use will be shown the last time each save set is shown in the output. The media pool the save set would be directed to is also listed in the preview output.

**-v**

Verbose. Print extra information about what **savegrp** is doing, and do not pass the **-q** flag along to **save** so it too will be chatty.

**-G**

Just run the group; apply no restart semantics. This is the default mode of operation; the option is provided for compatibility with other versions of **savegrp**.

**-R**

Restart. Use the information stored with the Legato Storage Manager server to restart a group which previously was terminated, generally due to a failure of the Legato Storage Manager server computer.

**-l level**

The level of save (see **nsr\_schedule(5)**) to perform on each client. This overrides the save level which **savegrp** would normally automatically determine. **-l** and **-C** cannot be specified together.

**-C schedule**

The name of the **NSR schedule** (see **nsr\_schedule(5)**) to be used in the automatic save level selection process which **savegrp** normally performs. This overrides the save schedule which **savegrp** would normally use for a given client. **-l** and **-C** cannot be specified together.

**-e expiration**

Set the date (in **nsr\_getdate(3)** format) when the saved data will expire. The special value **forever** is used to indicate that a volume that never expires (an archive or a migration volume) must be used. By default, no explicit expiration date is used.

**-t date**

The time to use instead of the current time for determining which level to use for this **savegrp** (in **nsr\_getdate(3)** format). By default, the current time is used.

**-F**

Automatically perform a full level backup if save set consolidation fails. This option is ignored if the backup level is not "c".

**-X**

Automatically remove the level 1 save set after save set consolidation builds a full level save set. This option is ignored if the backup level is not "c". It is also ignored if the backup level is "c" but the save set consolidation process fails.

**-r** *retries*

The number of times failed clients should be retried before **savegrp** gives up and declares them failed. The default is taken from the group resource. Abandoned saves are not retried, because they may eventually complete. Retries are not attempted if **-p** is specified.

**-P** *printer*

The printer which **savegrp** should use for printing bootstrap information.

**-W** *width*

The width used when formatting output or notification messages. By default, this is 80.

*groupname*

Specifies the Legato Storage Manager group of clients that should be started, rather than the default **NSR group** (which has the **name** attribute of **default**). See **nsr\_group(5)** for more details.

**-c** *client*

The name of a client on which to save file systems. There can be multiple **-c client** specifications. When **-c** options are specified, only the named clients from the specified group (which is "Default" if no group is specified) will be run.

## Resource Types

**NSR**

Use the **parallelism** attribute for the maximum number of saves to start simultaneously.

**NSR group**

The attribute *work list* contains values in groups of 3, specifying the client name, level of save, and path to save, for each save set not yet completed. The attribute *completion* contains values in groups of 4, specifying the client name, path saved, status, and the output, for each save set completed.

**NSR schedule**

Used by the **savegrp** command with each client's **nsr\_client(5)** resource to determine which level of save to perform for each specified save set.

**NSR client**

Each client resource names the groups it should be saved by, the names of the save sets which should be saved, the name of the schedule to use (see **nsr\_schedule(5)**), and the name of the directives to use (see **nsr\_directive(5)**).

**NSR notification**

Three kinds of notices are sent to the NSR notification system, both with the **event** attribute of *savegrp*. While a *savegrp* is in progress, status notices are sent with the **priority** of *info*. At completion of a *savegrp*, a notice is sent containing the collected output of all saves, and the name of clients which had a save which failed (if any). This notice will have an **event** type of *savegrp*, and a **priority** of *notice*. If *savegrp* is interrupted, a notice stating the group was terminated, with an **event** type of *savegrp*, and a **priority** of *alert* will be sent. These last two typically will result in the notice being encapsulated in a mail message to **root**.

## Savegroup Completion Notification Message

The savegroup completion notification message contains 5 parts: the header, the Never Started Save Sets, the Unsuccessful Save Sets, the Successful Save Sets, and the Cloned Save Sets. Each client in the group will be listed in one or more of sections categories (more than one if some save sets are in one category, and other save sets in another category). The clients are listed in alphanumeric sorted order, with the server listed last.

The header shows the name of the group and lists which clients failed. If the group was aborted, the header includes an indicator of this as well. The header also shows the time the group was started (or restarted, if the **-R** option was used), and the time the *savegrp* completed. The failed clients list in the header shows only those clients for which saves were attempted, not those for which saves never started.

The Never Started Save Sets section is optional and will only be included if there are some save sets of some clients in the group that were never started. This should only occur when a *savegrp* is aborted, either by killing the master *savegrp* daemon or by selecting the *Stop* function in the *Group Control* window or the *Stop Now* attribute in the *Group* window of **nwadmin**(8). Each entry listed in this section shows the client and save set that was never started (or *All* if no save sets were saved for that client). No other error messages should appear in this section.

The Unsuccessful Save Sets section shows all of the saves that were attempted but failed. This section will only be present if at least one save set failed. There are many reasons for a save to fail. The most common are listed as follows. More reasons will be listed in the future. It is important to differentiate between the many reasons for a save to fail, so that the administrator can quickly determine the cause and fix it, so the save will succeed the next time.

Each entry in the Unsuccessful Save Sets section lists the client and save set that failed, along with one or more lines of error and information messages. Each client is separated by a blank line, and all the failed save sets for a client a listed together.

Typical error or information messages are listed at the end of this section, (without the `client:saveset` prefix), with the necessary action(s) to take to correct the problem.

Each entry in the Successful Save Sets section lists the client and save set that succeeded, along with level of the save, the amount of data saved, the time to run the save set, and the number of files saved. Each entry may also be preceded by one or more warning or informational messages, the most common of which are listed as follows. These warning or informational messages are usually (but not always) prefixed by “\*”. Note that a save set’s output may include warnings; these do not necessarily mean the save set was unsuccessful. See **save(8)** for the definitions a successful and unsuccessful save sets.

The Cloned Save Sets section is somewhat different, because it refers to the save sets cloned, and not the clients that originated those save sets. The information shown in this section is the result of the **nsrclone** command. See the **nsrclone(8)** man page for information on the output of **nsrclone**.

The following is a list of common informational, warning and error messages found in the completion notification. This list is not complete. Note that the messages you see may vary slightly from those shown here due to differences in the operating system vendor-supplied error messages. Since many messages include client or server names, it is most efficient to look for a keyword in the error message. The messages are listed as follows in alphabetical order, by the first non-variable word in the message (note: initial words like "save", "asm" and "savefs" may or may not vary, and initial path names are always assumed to vary).

### **aborted**

This informational message only occurs when you stop a running `savegrp`, generally by selecting *Stop* from the Group Control Window of the **nwadmin(8)** interface. It means that the specified save set had started saving, but had not completed when the `savegrp` was stopped. The session (in the Sessions display of **nwadmin(8)**) for this save set may not disappear immediately, especially if **savegrp**’s attempt to terminate the save session fails. The save set will be retried if and when you *Restart* the `savegrp` (for example, from the Group Control Window).

### **Access violation from client - insecure port N**

This message, generated by the **save** command on client, means that **save** is not setuid root. Make sure that the **save** command on the client is owned by root and has its setuid bit set. If **save** is on an NFS mounted file system, make sure the file system was not mounted on that client using the "-nosuid" option.

### **Access violation – unknown host: client**

This message is caused when then the client’s hostname and IP address are not correctly listed in one or more of `/etc/hosts`, NIS or DNS on the server. You need to

either change the appropriate host table (depending on which one(s) are in use on your server) to list the client's name as it is known to Legato Storage Manager, as that client's primary name, or you need to add the name listed at the end of the error message to the *aliases* attribute of the client's Client resource(s).

***asm: cannot open path: I/O error***

This message generally means that there are bad blocks on the disk(s) containing the specified file or directory. You should immediately run a file system check on the named client file system and check your client's system error log. If there are bad blocks, repair them if possible, or move the file system to a different disk.

***asm: cannot stat path: Stale NFS file handle***

***asm: cannot stat path: Missing file or file system***

These informational messages (or variants of them for other operating systems) mean that when **save** attempted to test the named directory to determine if it was a different file system from the one currently being saved, the file system was, in fact NFS mounted, but the mount point was bad. While this message does not affect the saved data, it does mean you have a network or NFS problem between the specified client and one or more of its file servers. You may need to remount file systems on the client, or perhaps restart it, to correct the problem.

***/path/nsrexecd: Can't make pipe***

***/path/nsrexecd: Can't fork***

***fork: No more processes***

The specified client-side resource has been exceeded. There are too many other services running on the client while **savegrp** is running. Inspect the client and determine why it has run out of resources. The client may need to be restarted. You should also consider re-scheduling any jobs automatically started on the client (for example, using **cron**(8)) that run while **savegrp** is running.

***asm: chdir failed path: Permission denied***

This message means that while backing up the specified save set, **save** was unable to enter the named directory. This may mean that **save** is not setuid root on the specified client, or that the directory is actually an NFS mount point for which root is not allowed access. Check the permissions on **save** on the specified client (using **ls**(1)) and make sure that **save** is owned by root and that the setuid bit is set.

***connect to address AA.BB.CC.DD: message***

***Trying AA.BB.CC.DD...***

These informational messages are displayed only when the **-v** option is used. They mean that the connection to the client failed on the address specified in the first line of the message. If the client has more than one IP address, **savegrp** has attempted

the address listed in the second line. Looking at subsequent lines of the completion mail show if this second address succeeded. You may want to check and change your network routing tables to avoid getting these messages.

### **Connection refused**

This means the client computer is up, but it is not accepting new network connections for **nsrexecd** (or **rshd**). This could mean the client was in the process of booting when the savegrp attempted to connect, or that the client had exceeded some resource limit, and was not accepting any new connections. You should attempt to log into the client and verify that it is accepting remote connections. If the client is a non-UNIX computer, you may need to start the Legato Storage Manager client on that computer. Refer to your ClientPak installation for more information.

### **Connection timed out**

This usually means the client has crashed or is hung. Make sure the client has restarted, and that **nsrexecd** is running on it (if you are using **nsrexecd**). If the client is a non-UNIX computer, you may need to ensure that the network protocols are loaded, and that the Legato Storage Manager client is running on that computer. Refer to your ClientPak installation for more information.

### **asm: external ASM 'asm2' exited with code 1**

This message generally accompanies another message reporting a specific problem while saving a file or directory on the named save set. The backup will attempt to continue and attempt to save other data, and generally, the backup will not be listed in the failed save sets section of the completion mail if any files on the save set are saved successfully, even if it only saves the top directory of the save set.

### **save: path file size changed!**

This informational message is often generated when Legato Storage Manager backs up log files. It may also occur for other files. For files that you expect to grow while **savegrp** is running, you can use a directive specifying that the **logasm(8)** should be used to back up the file. See also **nsr(5)** and **nsr\_directive(5)**.

### **asm: getwd failed**

This message means that while backing up the specified save set, an attempt to determine the current directory's name failed. This occurs on clients, generally running older versions of the Legato Storage Manager ClientPak, on which the **getwd(3)** library call is broken. You may want to contact Legato Tech Support to find out if there is a patch available for your client platform to work around this vendor-specific bug, or contact your operating system vendor to see if a more recent O.S. version addresses this problem.

**group *groupname* aborted, savegrp is already running**

This message is only delivered by itself. It occurs when the named group has already been started or restarted (for example, after a restart, or when requested through the Group Control Window of **nwadmin(8)**), either automatically by **nsrd(8)** or manually, from the command line. You can use **ps(1)** to find out the process id of a running **savegrp**. The existence of a running group is determined by looking for a file named **/nsr/tmp/sg.*group*** which, if existing and locked, means a **savegrp** is running.

**nsrexec: Attempting a kill on remote save  
*client:saveset* aborted due to inactivity**

The client has not sent any data to the server for the specified inactivity timeout. **savegrp** will attempt to terminate the backup in progress so that the suspended client will not impede other backups or cloning operations.

**has been inactive for *N* minutes since *time*.*****client:saveset* is being abandoned by savegrp.**

A backup of the specified save set started, but after *N* minutes of no activity, **savegrp** gave up on the save set. Generally, this means that the client is suspended waiting for an NFS partition. Unfortunately, Legato Storage Manager (or any other program) has no way of reliably telling if an NFS partition will become suspended until after it tries to access the partition. When the partition comes back on line, the save will complete, although **savegrp** abandoned it. You should check the client, however, since you sometimes need to restart the client to get the NFS partitions back online. Non-UNIX clients also hang for other reasons, most notably, bugs in the operating system implementation of their network protocols.

**Host is unreachable**

The Legato Storage Manager server cannot make TCP/IP connections to the client. This generally means the network itself is not configured correctly; most commonly, one or more gateways or routers are down, or the network routes were not set up correctly. You should verify that the server can connect to the client, and if not, check and, if necessary, reconfigure your routers, gateways or routing tables.

**Login incorrect**

This message is generated when the *remote user* attribute for the client is not set to a valid login on the client. Verify that the *remote user* attribute for the client is set to the correct login name. You may see this message even when running **nsrexecd** if **nsrexecd** has not been started (or was killed) on the client.

***asm: missing hard links not found:***

This message is generated when a backed-up file had one or more hard links that were not found. The message is followed by a list of one or more filenames which were backed up minus some links. The message means that the files were either

created (with multiple hard links) while the backup was occurring, so some of the links were missed due to the order of file system tree walking, or the file (or some links) were removed while the backup was occurring. Only those links that were found can be recovered; additional links will have been lost. One can do an additional incremental backup of the affected file system if a consistent state for the affected file is essential.

**lost connection to server, exiting**

**save: network error, server may be down**

The backup of the named file system was begun, but the connection to the Legato Storage Manager server closed part way through. This typically means that the server computer restarted, one or more Legato Storage Manager server daemon processes were killed by the system administrator or by the system itself (for example, due to overwriting the binary or a disk error in swap space), or there was some transport problem that caused the network connection to be dropped by the operating system. Restart the save at a later time.

**no cycles found in media db; doing full save**

This informational message is added by **savegrp** to any save set that is saved at the level *full* instead of the level found in the client's schedule. Due to timing problems, you can occasionally see this message when the clocks on the client and server are out of sync, or when **savegrp** starts before midnight and ends after midnight. You may also get spurious messages of this type from some versions of Legato Storage Manager client software backing up a NetWare BINDERY, which ignore the schedule and perform a full, no matter what. In both these cases, the client re-checks the level, and overrides the server's requested level.

**No more processes**

See "Can't make pipe", described previously.

**No 'NSR client' resource for client *clienthostname***

**savefs: cannot retrieve client resources**

This pair of messages occurs if the client's hostname changed (in */etc/hosts*, NIS or DNS). You may also have deleted the client's *Client* resource while **savegrp** was running. In the former case, you will need to add the client's new name to the *aliases* attribute of the client (this is a hidden attribute) using **nsradmin(8)** (selecting the *Hidden* display option) or **nwadmin(8)** (selecting the *Details View* option for the *Client* window). In the latter case, no additional action is required if this deletion was intentional (the next run of **savegrp** will not attempt to save the client). If it was accidental, and you did not want to delete the client, you should add the client back again and add the client back into the appropriate group(s). The next time **savegrp** runs, it will back up the client, just as if the client had been down the previous day.

**no output**

The save set completed, but returned no status information. The most common reasons are that the client failed or lost its network connection (a router between the client and server crashed) while the client was being backed up. Another is that the disk on which the client status was being logged filled up (perform a **df /nsr/tmp** to see if this was the case). To determine if the save set was saved, you can use **mminfo(8)**. For example, run **mminfo -v -c clientname -t '1 day ago'** and look at the *flags* column for the completion status. An 'a' flag means it aborted. Use a more distant time (the **-t** option) to look further back in time.

**file system: No such file or directory**

An explicit save set was named in the *Client* resource for the specified client, and that save set does not exist (or is not currently mounted) on the client. Make sure you spelled the save set name correctly (and that it is capitalized correctly), and log into the client and verify that the save set is mounted.

**/path/nsrexecd: Couldn't look up address for your host****/path/nsrexecd: Host address mismatch for server**

The **nsrexecd** daemon on the client managed to look up the server in the client's host table, but the address listed there did not match the address of the server. Every interface of the server must have a unique name listed in the host table (possibly with non-unique aliases or CNAME's), and each unique name must be listed as a valid server to **nsrexecd**.

**/path/nsrexecd: Host server cannot request command execution****/path/nsrexecd: Your host cannot request command execution**

The server is not listed in **nsrexecd**'s list of valid servers on the specified client. The list of valid servers is either on the **nsrexecd** command line (with one or more **-s server** options to **nsrexecd**), or in a file (with the **-f file** option to **nsrexecd**). If neither is specified, some versions of **nsrexecd** will look for a file named *servers* in the same directory that contains the *nsr.res* file. It may also be the case that the server is not listed in one or more of */etc/hosts*, *NIS*, or *DNS*, on the client, in which case **nsrexecd** cannot validate the server until the client's host naming configuration is fixed.

**/path/nsrexecd: Invalid authenticator****/path/nsrexecd: Invalid command**

These two messages should never occur in a savegroup completion message. They mean that **savegrp** did not follow its protocol correctly.

**/path/nsrexecd: Permission denied****Permission denied**

These similar messages are generated by **nsrexecd** and **rshd**, respectively. In either case, the server does not have permission to run commands on the client. In the case

of the first message, make sure that the server is listed as a valid server on the client (see "Host *server* cannot request command execution", described previously, for details). In the case of the second message, which does not mention **nsrexecd**, make sure that "*servername*" is listed in the client's *.rhosts* file (or, if you have set the *remote user* attribute for this client, the *.rhosts* file in the home directory for that user on the client).

***/path/savegrp: printing bootstrap information failed***

See "unknown printer" described as follows.

**reading log file failed**

After the specified save set completed, savegrp was unable to read the log file of the output status from the save set. This generally means that someone, or an automated non-Legato Storage Manager administrative program or script, removed the log file. This message can also occur if the file system on which the client logs are stored has run out of space (use **df /nsr/tmp** to determine if this is the case). Verify that no scripts remove files from */nsr/tmp* (which is where savegrp stores the save set log files).

**request from machine server rejected**

The server is not listed in the PC (NetWare or DOS) client's list of acceptable servers. See your ClientPak installation guide for instructions on adding the server to the client-side list.

***N* retries attempted**

***1* retry attempted**

One of these informational messages is prepended to a save set's output if **savegrp** was unable to backup the data on the first try and if the *client retries* attribute for the group has a value greater than zero. In this case, the specified number of retries was performed before the backup of the save set succeeded or was finally marked as failed.

**RPC error, details...**

**Cannot open save session with '*server*'**

The **save** command generates this message if it is unable to back up data to the Legato Storage Manager server. There are several possible *details*. The most likely causes are: resources are exceeded on the server so nsrd cannot accept new save sessions, nsrd actually died since savegrp started (however, this is unlikely, since you cannot normally receive a savegrp completion message after nsrd dies, but you can see this when using the **-p** option), there are numerous network errors occurring and save cannot open a session to save its data (check this by running **netstat -s** and see how many network errors are occurring; you may need to do this several times a few minutes apart to get the change in errors). Save cannot tell which of these three causes are the real cause. If you see these errors frequently, and

it looks like a server resource problem, you might consider increasing the value of the "*client retries*" attribute of the **group** resource having these problems. This won't decrease the resource utilization, but will make **savegrp** more robust so such problems (the trade-off is that increasing "*client retries*" will increase the load on the server even more).

**RPC exec on *client* is unavailable. Trying RSH.**

This informational message is only displayed when the **-v** flag has been used for verbose information. This message means that **nsrexecd** is not running on the client, and that **savegrp** is attempting to use the **rshd** service instead, for backward compatibility with older versions of **savegrp**.

**save: *clientname2* is not on *client*'s access list**

This error occurs when the named client has more than one name, for example, a short name, *client*, and a fully-qualified domain name, *client.legato.com*. When the client attempts to connect back to the Legato Storage Manager server to start a save, that client is calling itself by the name *client*, which matches the client resource name, but when the server looks up the client's network address, it is getting back the name *clientname2*. If this is, in fact, correct, add the name *clientname2* to the client's **aliases** attribute, and re-run the save.

**save: path length of *xxxx* too long, directory not saved**

This message can occur if you have a directory tree that is very deep, or directory names that are very long. This message can also occur if there are bad blocks in the specified file system, or if the file system is damaged. Legato Storage Manager limits the full path name to 1024 characters which is the system imposed maximum on most systems. To save such directories, you need to rename or move the directories so that the full path name is shorter than 1024 characters. If the file system appears to be damaged (for example, a very long path name that looks like it has a loop in the name), perform a file system check on the specified client.

**/path/save: Command not found**

**/path/savefs: Command not found**

**/path/save: Not found**

**/path/savefs: Not found**

The **save** or **savefs** command could not be found in the specified path. If you are using **nsrexecd**, this probably means that the **save** or **savefs** command is not in the same directory in which **nsrexecd** is installed (or that **save** or **savefs** was removed). If you are using **rshd** for remote execution, then you need to set the *executable path* attribute in the *Client* resource for this client to be the directory in which the Legato Storage Manager executables are installed on the client.

**savefs: error starting save of file system**

This informational message accompanies several other **save** or **asm** messages listed here. This message means that savefs has detected the failed save and has marked the save set as failed.

**save: unknown host name: server****savefs: unknown host name: server**

The host table on the specified client (either `/etc/hosts`, NIS or DNS, depending on that client's configuration) does not include the server's name. You need to add the server's hostname to the specified client's host table. Note that if you use DNS but the server's Client resource name (the client resource for the server itself) is not fully qualified (that is, it looks like "server", not "server.dom.ain", and the server is in a different domain from the client, you will need to add the name *server* to the domain table for the domain containing the client. If you use NIS, this error means that either the NIS hosts map does not contain the server, the `/etc/hosts` file does not list the server, or the NIS master for the specified client is otherwise mis-configured (the server is a secondary server and there is no **yppush(8)** from the primary; run **ypwhich -m** on the client to find out which NIS server is providing master translation).

**savegrp: client rcmd(3) problem for command '*command*'**

This error message normally accompanies another, more specific, error message. It is generated when the attempt to run the specified command (usually **save** or **savefs** with several command line parameters) failed on the specified save set. The previous line of error output should include the more specific error message (look for that message elsewhere in this section). Generally, the problem is a bad hosttable configuration, or various permissions denied errors (server not specified when starting nsrexecd, or missing permissions in `.rhosts` if not using nsrexecd). If not, log into the Legato Storage Manager server as root and run the command **savegrp -p -v -c *clientname groupname*** giving the appropriate client for *clientname* and *groupname*. This verbose output should include the necessary additional information needed for fixing the problem.

**Saving server index because server is not in an active group**

This informational message, generated by **savegrp**, means that savegrp has noticed that the Legato Storage Manager server is not listed in any automatically started, enabled group. Since all of the indexes are stored on the server, **savegrp** is saving the server's index and bootstrap information in case a disaster occurs. You should add the server to a group with autostart enabled, or enable one of the groups of which the server is already a member.

**socket: All ports in use**

The Legato Storage Manager server has run out of socket descriptors. This means that you have exceeded the socket resource limit on your server. To avoid such future messages, you should determine what other network services are running while savegrp is running, and consider re-scheduling either savegrp or the other service(s). You can also reduce the parallelism in the **nsr\_service(5)** resource, to reduce the resource utilization.

**socket: protocol failure in circuit setup.**

The client does not seem to support the TCP/IP protocol stack, or has not used a privileged port for setting up its connection. The latter could occur if you use **nsrexecd** but did not start it as root on the specified client. The **nsrexecd** daemon must run as root on each client.

***path*: This data set is in use and cannot be accessed at this time**

This message is generated by save sets on PC clients running DOS or NetWare. The Legato Storage Manager client software on these systems cannot back up files open for writing, due to the interface provided by the operating system. This message actually comes from Novell's TSA and is not changeable.

**unknown host**

The specified client is not listed in the host table on the server (note: a similar "save" or "savefs" specific message is described previously). Depending on your host configuration, this means the client is not listed in one (or more) of /etc/hosts, NIS, or the Domain Name Service. If you use fully qualified domain names, you may need to make a new client resource for this client, using that fully qualified domain name (name the client resource "mars.legato.com", not "mars").

***printer*: unknown printer*****path/savegrp*: printing bootstrap information failed (reproduced below)**

This message, or similar messages, accompanies the bootstrap information when **savegrp** was unable to print the bootstrap on the printer. You need to either specify a different printer in the *printer* attribute for the group, or configure your print server to recognize the printer (by default, your system's default printer is used). The bootstrap information is listed as part of the savegrp completion mail. You should print out this information immediately, in case your server has a disaster and loses a disk, and fix the printer name used by **savegrp**.

**Warning – file '*path*' changed during save**

This warning message is generated when **save** notices that the file's modification time changed while the file was being backed up. Legato Storage Manager does not attempt to lock files before saving them, since this would make backups run extremely slowly. You may want to backup files which generate this message

manually, to ensure that a consistent copy is saved. Legato Storage Manager does not attempt this automatically, to avoid trying forever on the same file.

**Warning: ‘client’ is not in the hosts table!**

This message is generated by a **save** or **savefs** command run on the specified client to save that client’s file systems. The client’s hostname is not listed in the host table on the client (either /etc/hosts, NIS or DNS, depending on that client’s configuration). This almost always results in a failed save. Fix the client’s host table and re-run the save.

**Warning: unsynchronized client clock detected**

After completing a save set, **savegrp** was able to determine that the client and server clocks were not synchronized (this cannot be detected for older clients). Unsynchronized clocks can cause various anomalous results, such as a client’s save sets always being backed up at the level *full*. Unsynchronized clocks, once detected, will not cause failed or incomplete backups, but can cause too much data to be backed up. Keeping the client and server clocks synchronized to within a minute will avoid such anomalous situations.

**asm: path was not successfully saved**

This message generally accompanies one or more other more-specific messages for the save set. The specified path within the current save set was not saved successfully. The backup will continue trying to back up other files and directories on the save set.

**asm: xdr\_op failed for path**

This error can be caused by several possible conditions (for example, out of memory, buggy networking software in the operating system, an external ASM unexpectedly exiting, a lost network connection). If it was due to a lost network connection, then the Legato Storage Manager server most likely exited (due to **nsr\_shutdown**). After restarting the server, rerun the group. If due to an ASM exiting unexpectedly (in this case, the message should be accompanied by a message describing which ASM exited unexpectedly), you may have found a bad block on the disk, or perhaps a bug. Check if the client ran out of memory (there may be console messages), and verify that there are no bad blocks on the save set’s disk. If there were network errors, there may also have been messages logged by other programs on the system console (client or server), or to system log files.

## Files

**/nsr/tmp/sg.group**

A lock file to keep multiple savegrps of the same group from running simultaneously.

*/nsr/tmp/sg.group.client.\**

Temporary files used to log the output of individual save sets for the named group and client.

*/nsr/tmp/ggroup\**

On file systems with short names (less than 64 characters), the temporary files used to log the output of individual save sets for the named group.

**See Also:** [ls\(1\)](#), [ps\(1\)](#), [nsr\\_getdate\(3\)](#), [rcmd\(3\)](#), [fstab\(5\)](#), [nsr\(5\)](#), [nsr\\_directive\(5\)](#), [nsr\\_notification\(5\)](#), [nsr\\_service\(5\)](#), [nsr\\_group\(5\)](#), [nsr\\_schedule\(5\)](#), [nsr\\_resource\(5\)](#), [mminfo\(8\)](#), [nsrssc\(8\)](#), [netstat\(8\)](#), [nsr\(8\)](#), [nsradmin\(8\)](#), [nsrexec\(8\)](#), [nsrexecd\(8\)](#), [nsrwatch\(8\)](#), [nwadmin\(8\)](#), [rshd\(8\)](#), [save\(8\)](#), [savefs\(8\)](#), [yppush\(8\)](#)

## scanner(8)

### Name

**scanner** – Legato Storage Manager media verifier and index rebuilder

### Synopsis

```

scanner [ options ] -B device
scanner [ options ] -B -S ssid [ -im ] device
scanner [ options ] -i [ -S ssid ] [ -c client ] [ -N name ] device
scanner [ options ] -m [ -S ssid ] device
scanner [ options ] [ -S ssid ] [ -c client ] [ -N name ] device [ command ]

options:
[ -npqv ] [ -f file ] [ -r record ] [ -s server ] [ -t type ] [ -b pool ]

command:
-x command [ arg ... ]

```

### Description

The **scanner** command reads Legato Storage Manager media, such as backup tapes or disks, to confirm the contents of a volume, to extract a save set from a volume, or to rebuild the Legato Storage Manager online indexes. Only the super-user may run this command. The *device* must always be specified, and is usually one of the device names used by the Legato Storage Manager server; for tape drives, it must be the name of a “no-rewind on close” device.

When **scanner** is invoked with either no options or **-v**, the volume on the indicated *device* is opened for reading, scanned, and a table of contents is generated. The table of contents contains information about each save set found on the volume. By default, one line of information is produced for each save set found containing the client name, save set name, save time, level, size, files, ssid and a flag. The **client name** is the name of the system that created this save set. The **name** is the label given to this save set by **save(8)**, usually the path name of a file system. The **save time** is the date and time the save set was created. The **level** values are one-letter abbreviated versions of full, incremental, levels **0** through **9**, or blank for manual saves. The **size** is the number of bytes in the save set. The **files** labeled by column provide the number of client files contained in the save set. The **ssid** (save set

identifier) is an identifier used internally to reference and locate this save set. This same identifier may be specified explicitly with the **-S** option to extract a particular save set.

The table of contents is based on synchronization (sometimes called “note”) *chunks* (see **mm\_data(5)**) interspersed with the actual save set data. There are four types of note chunks: Begin, Continue, Synchronize, and End, symbolized by a *flag* of B, C, S or E respectively. The **B**egin note is used to mark the start of a save set. At the time a beginning chunk is written, the save set size and number of files are not known. The **C**ontinue note is used to indicate that this save set started on a different volume. The **S**ynchronize note marks locations in the save set where one may resume extracting data in the event of previous media damage (a client file boundary). The **E**nd note marks the end of the save set, and causes the table of contents line to be printed. The other notes are displayed only when the **-v** option is selected.

## Options

### **-b** *pool*

Specifies which pool the volume should belong too. This option only applies for versions of Legato Storage Manager that do not store the pool information on the media. For these versions, you might need to specify the media pool the volume should belong to if the user does not want the volume to be a member of the *Default* pool. For volumes where the pool information is stored on the media, the media must be relabeled (destroying all data on the media) to assign the media to a different pool.

### **-B**

When used in absence of the **-S** option, **scanner** quickly scans the tape for the start of bootstrap save sets. The program only reads the first record at each file mark on the media to see if a save set with the name “bootstrap” exists. When the entire media has been exhausted, the save set id and file location of the most recent bootstrap save set is printed. When used in conjunction with the **-S**, option the save set id specified is flagged as that of a bootstrap.

### **-c** *client*

Only processes save sets that come from the specified Legato Storage Manager client computer. This option can be used multiple times and in conjunction with the **-N** option, but only in presence of the **-i** or **-x** option.

### **-f** *file*

Starts the scan at the specific media file number. This can save time by avoiding the scan of potentially unused information if the entire volume is not being scanned and you happen to know from already existing Legato Storage Manager media

information (see **mminfo(8)**) where the area of interest on the volume starts. This option does not make sense on some kinds of media, such as optical disks and file device types, for example.

**-i**

Rebuilds both the media and the online file indexes from the volumes read. If you specify a single save set with the **-S *ssid*** option, only entries for the specified save set are copied to the online file index.

**-m**

Rebuild the media indexes for the volumes read.

**-n**

Go through all the motions, but do nothing with regard to media or index database rebuilding. When used with the **-i** option, this option provides the most complete media checking available, while not modifying the databases at all.

**-N *name***

Only processes save sets specified by *name* (a literal string only). This option can be used multiple times and in conjunction with the **-c** option, but only in presence of the **-i** or **-x** option.

**-p**

Prints out information save set notes as they are processed.

**-q**

Displays only errors or important messages.

**-r *record***

Starts the scan at the specific media record number. This can save time by avoiding the scan of potentially unused information if the entire volume is not being scanned and you happen to know from already existing Legato Storage Manager media information (see **mminfo(8)**) where the area of interest on the volume starts.

**-s *server***

Specifies the controlling server when using **scanner** on a storage node. See **nsr\_storage\_node(5)** for additional detail on storage nodes.

**-S *ssid***

Extracts the specified save set(s). When used with the **-i** or **-x** option, this option can be used multiple times and is in addition to any save sets selected using the **-c** and **-N** options. Otherwise, the volume is scanned for save set *ssid*. Most often this is piped to a **uasm(8)** program running in recover mode to process the save set (potentially with a directory list to limit the files to be recovered and potentially using a **-m** argument to map the file location). When using **-S** without **-i** or **-m**, **scanner** prompts for the volume block size if the volume label is not readable. If the

volume information is still in the media database, the user has the option of running `recover` by save set (see `recover(8)`). When `-B` is also specified, `ssid` is taken to be that of a bootstrap. Only one `ssid` is allowed in this case.

**-t type**

Specifies the type of media, for example, *optical* for an optical disk, or *8mm 5GB* for an 8mm 5GB tape). Normally the media type is obtained from the Legato Storage Manager server, if a known device is being used (see `nsr_device(5)`).

**-v**

Displays more verbose messages, such as a log of each note chunk, and a message after every 100 media records. When the `-i` option is used, a line is printed for each client file (an enormous amount of information can be produced).

**-x command arg ...**

Specifies an arbitrary UNIX command to process each new selected save set. This argument can only occur once at the end of the argument list (after *device*). The save stream for each save set is connected to the stdin of a new instance of the command. Most often this command is `uasm(8)` running in `recover` mode to process each save set (potentially using a `-m` argument to map the file location). If the volume information is still in the media database, the user has the option of running `recover` by save set (see `recover(8)`). Console I/O should not be attempted by a specified UNIX command. The user should instead specify conflict resolution parameters as arguments passed to the command (for example: `scanner -S ssid -x uasm -iR -rv`). If console interaction is required, pipe scanner output to the desired UNIX command instead of invoking the command through the `-x` option.

## Examples

*Verifying a tape:*

**scanner /dev/nrst0**

```
scanner: scanning 8mm tape mars.001 on /dev/nrst0
client name  save set  save time    level  size      files  ssid  S
space       /export  10/07/99 12:38  f      100762460 10035 16983 E
space       /usr     10/07/99 13:14  f      27185116  3185 16984 E
space       /nsr     10/07/99 12:40  f      77292280  8436 16980 S
space       /        10/07/99 13:22  f      1693192   518  6985  S
scanner: reached end of 8mm tape mars.001
```

*Rebuilding the online file index from a tape:*

**scanner -i /dev/nrst8**

```
scanner: scanning 4mm tape monday.fulls on /dev/nrst8
scanner: ssid 17458697: scan complete
scanner: ssid 17458694: scan complete
scanner: ssid 17458698: scan complete
scanner: ssid 17458693: NOT complete
scanner: reached end of 4mm tape monday.fulls
scanner: when next tape is ready, enter device name [/dev/nrst8]?
```

*Extracting a save set for /usr and relocating to /mnt:*

```
scanner -S 637475597 /dev/nrst8 | uasm -rv -m /usr=/mnt
or
scanner -S 637475597 /dev/nrst8 -x uasm -rv -m /usr=/mnt
```

*Extracting all save sets from client mars and relocating to /a:*

```
scanner -c mars /dev/nrst8 -x uasm -rv -m/=/a
```

**See Also:** [mm\\_data\(5\)](#), [mminfo\(8\)](#), [nsrmmdbasm\(8\)](#), [nsr\(8\)](#),  
[nsrindexasm\(8\)](#), [nsrmmmd\(8\)](#), [nsr\\_device\(5\)](#), [nsr\\_storage\\_node\(5\)](#),  
[uasm\(8\)](#)

## Diagnostics

**xdr conversion error, fn %d, rn %d, chunk %d out of %d**

**unexpected file number, wanted %d got %d**

**unexpected record number, wanted %d got %d**

All three preceding messages are indicative of media errors (tape blocks are either lost or damaged). In the case of an **xdr conversion error**, a nonzero “chunk” number means that the block may be partially salvageable. Unexpected file numbers are normal when **scanner** reaches the logical end of the media that has been recycled.

**continuation of data in nsrscan.NNNNN.MMMMMM**

After an **xdr** decode error (an error denoted by one or more of the messages described previously), **scanner** attempts to re-synchronize and send the rest of the stream. However, because programs like **uasm(8)** are unable to handle decoding streams with parts missing in the middle, scanner sends the remainder of the stream to a file. You can decode this stream manually. For example, if your original command was:

**scanner -S ssid | uasm -r**

and a synchronization error occurs, you can decode the rest of the stream with the following command:

**uasm -r < nsrscan.NNNNN.MMMMMM**

where the filename you enter corresponds to the name printed in the diagnostic message.

**unexpected volume id, wanted *valid1* got *valid2***

This message normally appears when running in verbose mode on a tape or disk that has been recycled. It does not indicate an error condition, but details the conditions normally treated as the end of the volume.

**ssid %d: finished, but incomplete**

**Scanner** has detected the end of a save stream, but the stream was aborted, and is of dubious value. If online indexes are being rebuilt, the end of the aborted stream may precipitate the next message.

**(ssid %d): error decoding save stream**

As indexes are being rebuilt, **scanner** detected that the bytes in the save stream are invalid. This is usually caused by processing an aborted save stream. Other causes may include a damaged tape. Once this condition is detected, the process of rebuilding the indexes for the particular save stream exits. This may precipitate the next message.

**write failed, Broken pipe**

Printed by **scanner** when a process rebuilding a save stream's indexes exits before consuming the entire stream.

**You are not authorized to run this command**

A normal (non-root) user invoked this command.

**could not convert 'arg' to a file number**

The **-f** and **-r** options require a numeric argument for the starting file or record number of the media.

**already exists in the media index**

The **-i** or **-m** option was specified and the volume was already listed in the media database. This message is purely informational, and means that the volume is not being added to the media database because it is already listed there.

**fn %d rn 0 read error I/O error**  
**fn %d rn 0 read error I/O error**  
**done with *tape\_type* tape *valid* *volume\_name***

These messages, when occurring together on AIX, are a consequence of **scanner** hitting consecutive filemarks at end of the media. They do not indicate an error condition, and can be ignored.

## Limitations

Although **scanner** was originally intended to be standalone, it now requires the Legato Storage Manager services (for example, **nsrd(8)** and **nsrmmdbd(8)**) to be running in order to query and, if requested, update the online RAP, media and file indexes.

When scanning a relabeled optical volume (a re-writable optical volume that had been written once, then re-labeled and used again), **scanner** may read off the end of the new data, and attempt to read the old data from the previous version of the volume, terminating with an “unexpected volume id” error. This error occurs after all the good data has been read, and can be ignored.

## tapeexercise(8)

### Name

**tapeexercise** – Exercise a tape drive

### Synopsis

**tapeexercise** [ **-vBEFP** ] *devname*

### Description

The **tapeexercise** program writes sample data to a tape, and tests to see if positioning and read operations perform as expected. It needs a write-enabled tape on the indicated (no-rewind) drive. Tapeexercise should be used with extreme caution. Successful completion is indicated by a “*test name*: test begin”, “*test name*: test ok” pair for each test that is run. An example is as follows:

```
BasicTest: test begin
BasicTest: test ok
```

### Options

**-v**  
Operate in verbose mode.

**-B**  
Perform only the basic test.

**-E**  
Perform only the EOT test.

**-F**  
Perform only the File Space Forward test.

**-P**  
Perform only the 'SCO' positioning test.

If none of the options **BEFP** are set, then all of the tests are performed.

*devname*

The device name of the tape device under test. This should be a non-rewinding device, following the local operating system convention.

**See Also:** [nsrmmd\(8\)](#), *Legato Hardware Compatibility Guide*

## Limitations

The **tapeexercise** program will generally fail for QIC drives, because these devices do not support all of the functionality assumed by **tapeexercise**, most importantly, they do not support back-skip-file. Such devices may work with **nsrmmd(8)**; consult the *Legato Hardware Compatibility Guide* on the Web site [www.legato.com](http://www.legato.com) for a complete list of supported devices.

## uasm(8)

### Name

**uasm** – Legato Storage Manager module for saving and recovering UNIX file system data

### Synopsis

**uasm** -s [ -benouv ] [ -ix ] [ -t *time* ] [ -f *proto* ] [ -p *path* ] *path...*

**uasm** -r [ -dnuv ] [ -i{nNyYrR} ] [ -m *src=dst* ] -z *suffix* [ *path* ]...

**uasm** -c [-nv] [ *path* ]...

### Description

The **uasm** command is the default file system ASM (Application Specific Module). It is built into **save(8)** and **recover(8)**. **uasm** may also be called directly in a manner similar to **tar(1)**. This description of **uasm** applies to all ASM's. For clarity, only **uasm** is mentioned in many of the descriptions in this man page.

**uasm** has three basic modes, saving, recovering, and comparing. When saving, **uasm** will browse directory trees and generate a save stream (see **nsr\_data(5)**), to the associated *stdout* file representing the file and directory organization. When recovering, **uasm** reads a save stream from the associated *stdin* file and creates the corresponding directories and files. When comparing, **uasm** reads a save stream from its *stdin* file and compares the save stream with the files already located in the file system.

During backup sessions, the behavior of **uasm** can be controlled by *directives*. Directives control how descendent directories are searched, which files are ignored, how the save stream is generated, and how subsequent directive files are processed. (See **nsr(5)**). When browsing a directory tree, symbolic links are never followed, except in the case of **rawasm**.

ASMs can recover save streams from current or earlier versions of Legato Storage Manager. Older ASMs may not be able to recover files generated by newer ASMs.

The following list provides a brief description of the ASMs supplied with Legato Storage Manager:

**skip**

The **skip** ASM does not back up the specified files and directories, and does not place the filename in the online index of the parent directory.

**null**

The **null** ASM does not back up the specified files and directories, but keeps the filename in the online index of the parent directory.

**nullasm**

**nullasm** is an alternate name for the **null** ASM, named for backward compatibility with earlier releases where **nullasm** was a separate executable program instead of an *internal* ASM.

**holey**

The **holey** ASM handles holes or blocks of zeros when backing up files and preserves these holes during recovery. On some file systems interfaces can be used to find out the location of file hole information. Otherwise, blocks of zeros that are read from the file are skipped. Note that this ASM is normally applied automatically and does not need not be specified.

**always**

The **always** ASM always performs a back up of a file, independent of the change time of the file.

**logasm**

The **logasm** enables file changes during backup sessions. **logasm** can be used for "log" files and other similar files where a file changes during a backup operation is not worth noting.

**mailasm**

The **mailasm** uses mail-style file locking and maintains the access time of a file, preserving "new mail has arrived" flag on most mail handlers.

**atimeasm**

The **atimeasm** is used to backup files without changing the access time of the file. This functionality is a subset of **mailasm**. On most systems, **atimeasm** uses the file *mtime* for selection and then resets the file *atime* after the backup (which changes the file *ctime*). On systems that support interfaces for maintaining the file *atime* without changing the file *ctime*, **atimeasm** has no effect, since the file *atime* is normally preserved.

**posixcrcasm**

The **posixcrcasm** is used to calculate a 32-bit CRC for a file during a backup. This CRC is stored along with the file and is verified when the file is restored; no verification occurs during the backup itself. Using this ASM it is possible to validate a file at restore time, but it does not provide a way to correct any detected errors.

**rawasm**

The **rawasm** is used to back up */dev* entries (for example, block- and character-special files) and their associated raw disk partition data. On some systems, */dev* entries are actually symbolic links to device specific names. Unlike other ASMs, **rawasm** follows symlinks, allowing the shorter */dev* name to be configured. When recovering, **rawasm** requires that the file system node for the raw device exist prior to the recovery. This protects against the recovery of a */dev* entry and the overwriting of data on a reconfigured disk. You can create the */dev* entry, having it refer to a different raw partition, and force an overwrite if desired. If you create the */dev* entry as a symbolic link, the data is recovered to the target of the symbolic link. Precautions should be taken when using **rawasm**, see the following **Caveats** section.

**swapasm**

The **swapasm** does not backup actual file data, but re-creates a zero-filled file of the correct size on recovery. This asm is used on systems where the swapping device is a swap file that must be recovered with the correct size, but the contents of the swap file are not important and do not need to be backed up.

**xlateasm**

The **xlateasm** translates file data so that data backed up is not immediately recognizable.

**compressasm**

The **compressasm** uses a software compression algorithm to compress file data. This asm does not compress directories. The amount of compression achieved is data-dependent. **compressasm** uses considerable amounts of CPU resources, so its benefits may be limited on low-powered systems.

**nsrmmdbasm**

The **nsrmmdbasm** is used to process Legato Storage Manager's media index. Normally, **nsrmmdbasm** is invoked automatically by **savegrp** and **mmrecov**, and should not be used in Legato Storage Manager directives.

**nsrindexasm**

The **nsrindexasm** is used to process Legato Storage Manager's client file indexes. Similar to **nsrmmdbasm**, **nsrindexasm** is invoked automatically by **savegrp**, **mmrecov** and **recover**, and should not be used in Legato Storage Manager directives.

*Internal* ASMs are not separate programs, but are contained within all ASMs. *External* ASMs are separate programs, and are invoked as needed. The only external ASMs provided with Legato Storage Manager are **nsrmmdbasm** and **nsrindexasm**. All other ASMs listed previously are internal.

For security reasons, external ASM names must end in *asm* and be located in the *origin* directory, which is the same directory as the originally invoked program (typically **save** or **recover**). In some system architectures, other directories relative to the *origin* will be searched if an ASM cannot be located in the *origin* directory.

*Walking* ASMs traverse directory trees. The **skip**, **null**, and **nullasm** ASMs do not walk.

The internal ASMs described here are modes, and a number of different internal ASMs may be applied at the same time. When an external ASM is needed to process a file, the new ASM is invoked and generates the save stream. When a *filtering* ASM is traversing a directory tree and invokes another ASM, that ASMs save stream is processed by the *filtering* ASM. Hence, while using **compressasm** to backup a directory, the **mailasm** can still be used to process the mail files correctly. Note that once different modes are set, the only way to turn them off is to explicitly match an ASM directive for **uasm**.

*Auto-applied* ASMs are used under certain conditions, and do not need to be specifically mentioned in a directive file. For example, when a large file only has a small number of disk blocks allocated, the **holey** ASM is automatically invoked to process the file. Auto-applied ASMs are not used when a filename matches an explicit directive.

When used in conjunction with **recover**, all standard ASMs support security at recovery time. If a file is saved with an access control list (ACL), then only the owner of the file, root or Administrator may recover the file. For files that do not contain an ACL, the standard mode bits are used to determine who may recover a file. The file's owner, root and Administrator may always recover the file. Note that when ASMs are invoked by hand, these security checking rules do not apply.

## Options

All ASMs accept the options described as follows. These options are generally referred to as the *standard-asm-arguments*. ASMs may also have additional options, which must be capital letters.

Either **-s** (saving), **-r** (recovering) or **-c** (comparing) mode must be specified and must precede any other options. When saving, at least one *path* argument must be specified. *path* can be either a directory name or filename.

The following options are valid for all modes:

**-n**

Performs a dry run. When backing up, browse the file system, create the save stream, but do not attempt to open any files. When recovering or comparing, consume the input save stream and perform basic sanity checks, but do not create any directories or files when recovering or comparing file data.

**-v**

Turns on verbose mode. The current ASM, its arguments, and the file being processed are displayed. When a filtering ASM operating in filtering mode (processing the save stream of another ASM) modifies the stream, its name, arguments and the current file are displayed within square brackets.

**-u**

This option makes the asm stop when an error that would normally cause a warning occurs. This can be useful if you are recovering to a file system that may not have enough disk space or you are performing a save and you want any warnings to stop the save. If you use this option with **uasm** on recovery, it will stop if it runs out of disk space. Without this option, **uasm** will continue to try to recover each file until it has processed the entire save stream.

When saving, the following options may also be used:

**-b**

Produces a byte count. This option is similar to the **-n** option, but byte count mode will estimate the amount of data that would be produced instead of actually reading file data so it is faster but less accurate than the **-n** option. Byte count mode produces three numbers: the number of records (for example, files and directories), the number of bytes of header information, and the approximate number of bytes of file data. Byte count mode does not produce a save stream, so its output cannot be used as input to another ASM in recover mode.

**-o**

Produces a (see **nsr\_data(5)**) save stream that can be handled by older Legato Storage Manager servers.

**-e**

Do not generate the final "end of save stream" boolean string. This flag should only be used when an ASM invokes an external ASM and as an optimization chooses not to consume the generated save stream itself.

**-i**

Ignore all save directives from *.nsr* directive files found in the directory tree.

**-f proto**

Specifies the location of a *.nsr* directive file to interpret before processing any files (see **nsr(5)**). Within the directive file specified by *proto*, *<<path>>* directives must resolve to files within the directory tree being processed, otherwise their subsequent directives will be ignored.

**-p path**

This string is prepended to the name of each file as it is processed. This argument is used internally when one ASM executes another external ASM. *path* must be a properly formatted path that is either the current working directory or a trailing component of the current working directory.

**-t date**

The date (in **nsr\_getdate(3)** format) after which files were modified will be backed up.

**-x**

Cross file system boundaries. Normally, file system boundaries are not crossed during walking. Symbolic links are never followed, except in the case of **rawasm**.

When recovering, the following options may also be used:

**-i{nNyYrR}**

Specifies the initial default overwrite response. Only one letter can be used. When the name of the file being recovered conflicts with an existing file, the user is prompted for overwrite permission. The default response, selected by pressing [**Return**], is displayed within square brackets. Unless otherwise specified with the **-i** option, **n** is the initial default overwrite response. Each time a response other than the default is selected, the new response becomes the default. When either **N**, **R**, or **Y** is specified, there is no prompting (except when auto-renaming files that already end with the rename suffix) and each subsequent conflict is resolved as if the corresponding lowercase letter had been selected.

The valid overwrite responses and their meanings are:

- n** Do not recover the current file.
- N** Do not recover any files with conflicting names.
- y** Overwrite the existing file with the recovered file.
- Y** Overwrite files with conflicting names.
- r** Rename the conflicting file. A dot, ".", and a suffix are appended to the name of the recovered file. If a conflict still exists, the user will be prompted again.
- R** Automatically renames conflicting files by appending a dot (".") and a suffix. If a conflicting filename already ends in a "." suffix, the user is prompted to avoid potential auto rename looping condition.

**-m** *src=dst*

This option maps the filenames that are created. Any files that start exactly with *src* will be mapped to have the path of *dst*, replacing the leading *src* component of the path name. This option is useful for the relocation of recovered files that were backed up using absolute path names into an alternate directory (for example, **-m /usr/etc=.**).

**-z** *suffix*

Specifies the suffix to append when renaming conflicting files. The default suffix is "R".

*path*

Used to restrict the files being recovered. Only files with prefixes matching *path* will be recovered. This checking is performed before any potential name mapping is performed using the **-m** specification. When *path* is not specified, no checking is done.

## Caveats

Raw partitions are often used to store active DBMS data. If your raw partition contains data managed and updated by an active DBMS product, **rawasm** alone will *not* give a consistent backup. The database must *not* be updating the data in an uncontrolled fashion while **rawasm** saves or recovers data on the partition. The partition must be offline, the database manager shutdown, or the partition placed in an appropriate state for backup. Legato has products to assist with online database backup. Similarly if **rawasm** is used to save a partition containing a UNIX file system, the file system must be unmounted or mounted read-only to obtain a consistent backup.

Ideally, recovery of a raw partition should take place on a system configured with the same disk environment and same size partitions as the system which performed the backup. If the new partition is smaller than the original partition, the recovery will not complete successfully. If the new partition is larger than the original partition, only the amount of data originally saved will be recovered.

If the partition backed up includes the disk label – the label often contains the disk geometry – recovering this partition to a new disk also recovers the label, changing the new disks geometry to match the original disk. Similarly, if a UNIX file system partition is backed up using **rawasm**, recovering the partition resets all information on the partition, including timestamps concerning mount times (if applicable).

Since **rawasm** does not discover the size of the partition completed, the estimated size reported on recovery is not accurate.

## Examples

### *Copying files*

To copy all of the files in the current directory to *target\_dir*, use:

```
uasm -s . | (cd target_dir; uasm -rv)
```

This preserves ownership, time, and the other UNIX attributes. Only the data in holey files is copied; the holes are not copied.

### *Copying a file tree to an archive directory*

To copy the file tree under the directory *here* to *archive* and overwrite any files with conflicting names, use:

```
cd here
```

```
uasm -s . | (cd archive; uasm -r -iY)
```

Note that we **cd** to *here* first and give the first **uasm** determining the save a relative path so that the second **uasm** performing the recover will re-create the file tree under *archive*.

Another way to achieve the same result is to use the **-m** option on the second **uasm** performing the recover to explicitly map the path names.

```
uasm -s here | uasm -r -iY -m here=archive
```

## Files

### **.nsr**

Save directive files located throughout the file system.

**See Also:** [nsr\(5\)](#), [nsr\\_directive\(5\)](#), [nsrindexasm\(8\)](#),  
[nsrmmdbasm\(8\)](#), [nsr\\_data\(5\)](#), [recover\(8\)](#), [save\(8\)](#), [scanner\(8\)](#),  
[XDR\(3N\)](#)

---

---

# Index

## C

---

categories of LSM commands, 1-2

commands

- data management, 1-3
- device management, 1-4
- index management, 1-2
- LSM resources, 1-4
- LSM Server configuration, 1-5
- media management, 1-4
- mm\_data(5), 2-1
- mminfo(8), 2-6
- mmlocate(8), 2-20
- mmpool(8), 2-23
- mmrecov(8), 2-25
- nsr(5), 2-31
- nsr(8), 2-35
- nsr\_archive\_client(5), 2-43
- nsr\_archive\_request(5), 2-46
- nsr\_client(5), 2-50
- nsr\_crash(8), 2-58
- nsr\_data(5), 2-61
- nsr\_device(5), 2-65
- nsr\_directive(5), 2-76
- nsr\_getdate(3), 2-78
- nsr\_group(5), 2-80
- nsr\_label(5), 2-86
- nsr\_layout(5), 2-89
- nsr\_license(5), 2-91
- nsr\_migration(5), 2-94
- nsr\_notification(5), 2-99
- nsr\_policy(5), 2-102
- nsr\_pool(5), 2-104
- nsr\_regexp(5), 2-108

- nsr\_repack\_schedule(5), 2-110
- nsr\_resource(5), 2-113
- nsr\_schedule(5), 2-118
- nsr\_service(5), 2-121
- nsr\_shutdown(8), 2-130
- nsr\_stage(5), 2-132
- nsr\_storage\_node(5), 2-136
- nsradmin(8), 2-140
- nsralist(8), 2-146
- nsrarchive(8), 2-147
- nsrcap(8), 2-150
- nsrcat(8), 2-153
- nsrck(8), 2-154
- nsrclone(8), 2-158
- nsrd(8), 2-164
- nsrexec(8), 2-166
- nsrexecd(8), 2-167
- nsrfile(8), 2-169
- nsrhsmck(8), 2-172
- nsrib(8), 2-174
- nsrim(8), 2-179
- nsrindexasm(8), 2-186
- nsrindexd(8), 2-188
- nsrinfo(8), 2-190
- nsrlic(8), 2-195
- nsrls(8), 2-198
- nsrmig(8), 2-200
- nsrmm(8), 2-202
- nsrmmmd(8), 2-210
- nsrmmdbasm(8), 2-211
- nsrmmdbd(8), 2-213
- nsrmon(8), 2-216
- nsrpmig(8), 2-217
- nsrports(8), 2-221

- nsrrepack(8), 2-223
- nsrretrieve(8), 2-230
- nsrssc(8), 2-232
- nsrstage(8), 2-236
- nsrwatch(8), 2-240
- nwadmin(8), 2-242
- nwarchive(8), 2-244
- nwbackup(8), 2-246
- nwrecover(8), 2-248
- nwretrieve(8), 2-251
- preclntsave(8), 2-253
- pstclntsave(8), 2-254
- recover(8), 2-256
- save(8), 2-263
- savefs(8), 2-268
- savegrp(8), 2-273
- scanner(8), 2-291
- tapeexercise(8), 2-298
- uasm(8), 2-300

## D

---

- data management commands, 1-3
- device management commands, 1-4

## F

---

- features, unsupported, 1-1

## I

---

- index managements commands, 1-2

## L

---

### LSM commands

- categories, 1-2
- data management, 1-3
- device management, 1-4
- index management, 1-2
- LSM resources, 1-4
- LSM Server configuration, 1-5
- media management, 1-4
- mm\_data(5), 2-1
- mminfo(8), 2-6

- mmlocate(8), 2-20
- mmpool(8), 2-23
- mmrecov(8), 2-25
- nsr(5), 2-31
- nsr(8), 2-35
- nsr\_archive\_client(5), 2-43
- nsr\_archive\_request(5), 2-46
- nsr\_client(5), 2-50
- nsr\_crash(8), 2-58
- nsr\_data(5), 2-61
- nsr\_device(5), 2-65
- nsr\_directive(5), 2-76
- nsr\_getdate(3), 2-78
- nsr\_group(5), 2-80
- nsr\_label(5), 2-86
- nsr\_layout(5), 2-89
- nsr\_license(5), 2-91
- nsr\_migration(5), 2-94
- nsr\_notification(5), 2-99
- nsr\_policy(5), 2-102
- nsr\_pool(5), 2-104
- nsr\_regexp(5), 2-108
- nsr\_repack\_schedule(5), 2-110
- nsr\_resource(5), 2-113
- nsr\_schedule(5), 2-118
- nsr\_service(5), 2-121
- nsr\_shutdown(8), 2-130
- nsr\_stage(5), 2-132
- nsr\_storage\_node(5), 2-136
- nsradmin(8), 2-140
- nsralist(8), 2-146
- nsrarchive(8), 2-147
- nsrcap(8), 2-150
- nsrcat(8), 2-153
- nsrck(8), 2-154
- nsrclone(8), 2-158
- nsrd(8), 2-164
- nsrexec(8), 2-166
- nsrexecd(8), 2-167
- nsrfile(8), 2-169
- nsrhmck(8), 2-172
- nsrib(8), 2-174
- nsrim(8), 2-179
- nsrindexasm(8), 2-186
- nsrindexd(8), 2-188

nsrinfo(8), 2-190  
nsrlic(8), 2-195  
nsrls(8), 2-198  
nsrmig(8), 2-200  
nsrmm(8), 2-202  
nsrmmdb(8), 2-210  
nsrmmdbasm(8), 2-211  
nsrmmdbd(8), 2-213  
nsrmon(8), 2-216  
nsrpmig(8), 2-217  
nsrports(8), 2-221  
nsrrepack(8), 2-223  
nsrretrieve(8), 2-230  
nsrssc(8), 2-232  
nsrstage(8), 2-236  
nsrwatch(8), 2-240  
nwadmin(8), 2-242  
nwarchive(8), 2-244  
nwbackup(8), 2-246  
nwrecover(8), 2-248  
nwretrieve(8), 2-251  
preclntsave(8), 2-253  
pstclntsave(8), 2-254  
recover(8), 2-256  
save(8), 2-263  
savefs(8), 2-268  
savegrp(8), 2-273  
scanner(8), 2-291  
tapeexercise(8), 2-298  
uasm(8), 2-300

LSM resource commands, 1-4

LSM Server configuration commands, 1-5

## M

---

media management commands, 1-4

mm\_data(5), 2-1  
mminfo(8), 2-6  
mmlocate(8), 2-20  
mmpool(8), 2-23  
mmrecov(8), 2-25

## N

---

nsr(5), 2-31

nsr(8), 2-35  
nsr\_archive\_client(5), 2-43  
nsr\_archive\_request(5), 2-46  
nsr\_client(5), 2-50  
nsr\_crash(8), 2-58  
nsr\_data(5), 2-61  
nsr\_device(5), 2-65  
nsr\_directive(5), 2-76  
nsr\_getdate(3), 2-78  
nsr\_group(5), 2-80  
nsr\_label(5), 2-86  
nsr\_layout(5), 2-89  
nsr\_license(5), 2-91  
nsr\_migration(5), 2-94  
nsr\_notification(5), 2-99  
nsr\_policy(5), 2-102  
nsr\_pool(5), 2-104  
nsr\_regexp(5), 2-108  
nsr\_repack\_schedule(5), 2-110  
nsr\_resource(5), 2-113  
nsr\_schedule(5), 2-118  
nsr\_service(5), 2-121  
nsr\_shutdown(8), 2-130  
nsr\_stage(5), 2-132  
nsr\_storage\_node(5), 2-136  
nsradmin(8), 2-140  
nsralist(8), 2-146  
nsrarchive(8), 2-147  
nsrcap(8), 2-150  
nsrcat(8), 2-153  
nsrck(8), 2-154  
nsrclone(8), 2-158  
nsrd(8), 2-164  
nsrexec(8), 2-166  
nsrexecd(8), 2-167  
nsrfile(8), 2-169  
nsrhsmck(8), 2-172  
nsrib(8), 2-174  
nsrim(8), 2-179  
nsrindexasm(8), 2-186  
nsrindexd(8), 2-188  
nsrinfo(8), 2-190  
nsrlic(8), 2-195  
nsrls(8), 2-198  
nsrmig(8), 2-200

nsrmm(8), 2-202  
nsrmmmd(8), 2-210  
nsrmmdbasm(8), 2-211  
nsrmmdbd(8), 2-213  
nsrmon(8), 2-216  
nsrpmig(8), 2-217  
nsrports(8), 2-221  
nsrrepack(8), 2-223  
nsrretrieve(8), 2-230  
nsrssc(8), 2-232  
nsrstage(8), 2-236  
nsrwatch(8), 2-240  
nwadmin(8), 2-242  
nwarchive(8), 2-244  
nwbackup(8), 2-246  
nwrecover(8), 2-248  
nwretrieve(8), 2-251

## **P**

---

preclntsave(8), 2-253  
pstclntsave(8), 2-254

## **R**

---

recover(8), 2-256

## **S**

---

save(8), 2-263  
savefs(8), 2-268  
savegrp(8), 2-273  
scanner(8), 2-291

## **T**

---

tapeexercise(8), 2-298

## **U**

---

uasm(8), 2-300  
unsupported features, 1-1