

Veritas NetBackup™ Logging Reference Guide

Release 8.1.1

Veritas NetBackup™ Logging Reference Guide

Legal Notice

Copyright © 2018 Veritas Technologies LLC. All rights reserved.

Veritas, the Veritas Logo, and NetBackup are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third party software for which Veritas is required to provide attribution to the third party ("Third Party Programs"). Some of the Third Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third party legal notices document accompanying this Veritas product or available at:

<https://www.veritas.com/about/legal/license-agreements>

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Veritas Technologies LLC and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

<http://www.veritas.com>

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

<https://www.veritas.com/support>

You can manage your Veritas account information at the following URL:

<https://my.veritas.com>

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

The latest documentation is available on the Veritas website:

<https://sort.veritas.com/documents>

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1	Using logs	8
	About logs	8
	About UNIX system logs	10
	About log retention in NetBackup	10
	About limiting the size of unified and legacy logs	12
	About unified logging	13
	Gathering unified logs for NetBackup	14
	Types of unified logging messages	15
	File name format for unified logging	16
	Originator IDs for the entities that use unified logging	17
	About changing the location of unified log files	23
	About rolling over unified log files	24
	About recycling unified log files	25
	About using the vxlogview command to view unified logs	26
	About query strings used with the vxlogview command	27
	Examples of using vxlogview to view unified logs	30
	Examples of using vxlogmgr to manage unified logs	31
	Examples of using vxlogcfg to configure unified logs	34
	About legacy logging	36
	UNIX client processes that use legacy logging	38
	PC client processes that use legacy logging	40
	File name format for legacy logging	42
	Directory names for legacy debug logs for servers	43
	Directory names for legacy debug logs for media and device management	45
	How to control the amount of information written to legacy logging files	46
	About limiting the size and the retention of legacy logs	47
	Configuring the legacy log rotation	49
	About global logging levels	50
	Changing the logging level	52
	Changing the logging level on Windows clients	53
	Setting Media Manager debug logging to a higher level	53
	Setting retention limits for logs on clients	54
	Logging options with the Windows Event Viewer	54

	Troubleshooting error messages in the NetBackup Administration Console	57
	About extra disk space required for logs and temporary files	59
	Enabling detailed debug logging	59
Chapter 2	Backup process and logging	62
	Backup process	62
	NetBackup process descriptions	65
	Backup and restore startup process	65
	Backup and archive processes	66
	Backups and archives - UNIX clients	67
	Multiplexed backup process	67
	About backup logging	67
	Sending backup logs to Veritas Technical Support	69
Chapter 3	Media and device processes and logging	71
	Media and device management startup process	71
	Media and device management process	73
	Shared Storage Option management process	75
	Barcode operations	77
	Media and device management components	79
Chapter 4	Restore process and logging	88
	Restore process	88
	UNIX client restore	92
	Windows client restore	94
	About restore logging	95
	Sending restore logs to Veritas Technical Support	96
Chapter 5	Advanced Backup and Restore Features	98
	SAN Client Fiber Transport backup	98
	SAN Client Fiber Transport restore	101
	Hot catalog backup	103
	Hot catalog restore	105
	Synthetic backups	107
	Creating legacy log directories to accompany problem reports for synthetic backup	110
	Logs to accompany problem reports for synthetic backups	110

Chapter 6	Storage logging	112
	NDMP backup logging	112
	NDMP restore logging	115
Chapter 7	NetBackup Deduplication logging	118
	Deduplication backup process to the Media Server Deduplication Pool (MSDP)	118
	Client deduplication logging	121
	Deduplication configuration logs	121
	Media server deduplication/pdplugin logging	123
	Disk monitoring logging	124
	Logging keywords	124
Chapter 8	OpenStorage Technology (OST) logging	126
	OpenStorage Technology (OST) backup logging	126
	OpenStorage Technology (OST) configuration and management	128
Chapter 9	Storage lifecycle policy (SLP) and Auto Image Replication (A.I.R.) logging	132
	About storage lifecycle policies (SLPs) and Auto Image Replication (A.I.R.)	132
	Storage lifecycle policy (SLP) duplication process flow	133
	Automatic Image Replication (A.I.R.) process flow logging	134
	Import process flow	135
	SLP and A.I.R. logging	135
	SLP configuration and management	136
Chapter 10	Snapshot technologies	138
	Snapshot Client backup	138
	VMware backup	141
	Snapshot backup and Windows open file backups	144
Chapter 11	Locating logs	148
	acsssi logging	149
	bpbackup logging	149
	bpbkar logging	150
	bpbrm logging	150
	bpcd logging	151
	bpcompatd logging	151

bpdbm logging	152
bpjobd logging	152
bprd logging	153
bprestore logging	153
bptm logging	153
daemon logging	154
ltid logging	154
nbemm logging	155
nbjm logging	155
nbpem logging	156
nbproxy logging	156
nbrb logging	157
NetBackup web services logging	157
NetBackup web server certificate logging	158
PBX logging	159
reqlib logging	160
robots logging	160
tar logging	161
txxd and txxcd logging	161
vnetd logging	162

Chapter 12	Java-based administration console logging	163
	About the Java-based administration console logging	163
	Java-based administration console logging process flow	164
	Setting up a secure channel between the Java-based administration console and bpjava-*	165
	Setting up a secure channel between the Java-based administration console and either nbsl or nbvault	166
	Java-based administration console logging configuration on NetBackup servers and clients	168
	Java-based remote administration console logging on a Windows computer where NetBackup is not installed	168
	Configuring and gathering logs when troubleshooting Java GUI issues	169
	Undo logging	171

Index	172
--------------------	------------

Using logs

This chapter includes the following topics:

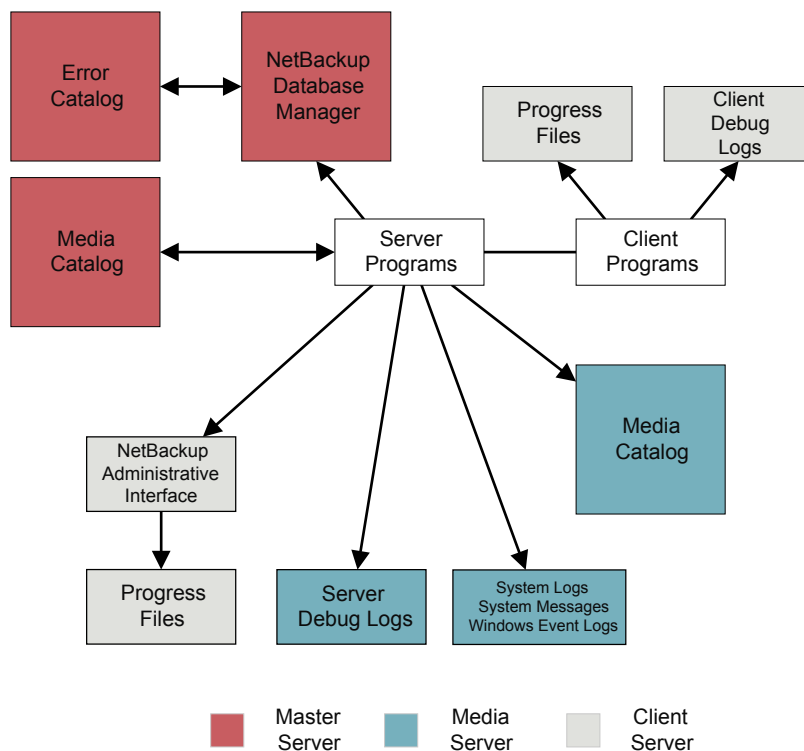
- [About logs](#)
- [About UNIX system logs](#)
- [About log retention in NetBackup](#)
- [About limiting the size of unified and legacy logs](#)
- [About unified logging](#)
- [About legacy logging](#)
- [About global logging levels](#)
- [Setting retention limits for logs on clients](#)
- [Logging options with the Windows Event Viewer](#)
- [Troubleshooting error messages in the NetBackup Administration Console](#)

About logs

NetBackup uses several different logs and reports to help you troubleshoot any problems that you encounter.

Users need to know where the log and report information is on their systems.

[Figure 1-1](#) shows the location of the log and report information on the client and the server and the processes that make the information available.

Figure 1-1 Logs in the NetBackup Enterprise system

You can review a functional overview that describes the programs and daemons that are mentioned in this figure.

You can also use NetBackup reports to help troubleshoot problems. NetBackup reports give information about status and errors. To run reports, use the **NetBackup Administration Console**.

See the Reports information in the [NetBackup Administrator's Guide, Volume I](#).

Note: The log-entry format in the NetBackup logs is subject to change without notice.

About UNIX system logs

The NetBackup server daemons and programs occasionally log information through `syslogd` and it then shows a message or writes the information in an appropriate system log or the console log.

On UNIX, NetBackup automatically records robotic and network errors in the system logs by using `syslogd`. On Windows, NetBackup records robotic and drive errors in the **Event Viewer** Application log. On both operating systems, log entries are also made when robotically controlled drives change between UP and DOWN states.

Note: On HP-UX, the `sysdiag` tool may provide additional information on hardware errors.

To enable additional logging by NetBackup to the system logs, use one of the following:

- Use the `ltid` command that started the device management processes. If the `-v` option is included on the `ltid` command, all daemons that were started as a result also have the `-v` option in effect.
- Use a command to start a specific daemon (for example, `acsd -v`).

On UNIX, enable debug logging to the system logs by including the verbose option (`-v`) on the command that you use to start a daemon.

To troubleshoot `ltid` or robotic software, you must enable system logging. See the `syslogd(8)` man page for information on setting up system logs. Errors are logged with `LOG_ERR`, warnings with `LOG_WARNING`, and debug information with `LOG_NOTICE`. The facility type is `daemon`.

See the `syslogd` man page for the locations of system log messages on your system.

About log retention in NetBackup

This section talks about various log retention options in NetBackup that help you recycle or delete logs as per your logging requirements.

Note: You can verify the log pruning behavior in NetBackup by using the logs at the following location:

On Windows: `install_path\NetBackup\logs\nbutils`

On UNIX: `/usr/openv/netbackup/logs/nbutils`

Table 1-1 Log retention options in NetBackup

Log retention option	Use this option...	Reference link
Keep logs up to GB	<p>To limit the size of unified and legacy logs.</p> <p>When the log size across NetBackup processes grows up to this configuration value, the older logs are deleted.</p> <p>This option is available on the NetBackup Administration Console > NetBackup Management > Host Properties > Logging dialog box.</p>	See “About limiting the size of unified and legacy logs” on page 12.
NumberOfLogFiles	<p>To limit the number of unified log files that you want to retain for a NetBackup process.</p> <p>When the number of log files exceeds this configuration value, the oldest log files become eligible for deletion during log cleanup.</p> <p>This option can be set using a command-line interface.</p>	See “About recycling unified log files” on page 25.
MaxLogFileSizeKB and other <code>vxlogcfg</code> options	<p>To prevent unified log files from becoming too large.</p> <p>When a file size or time setting is reached, the current log file is closed. New log messages for the logging process are written or “rolled over” to a new log file.</p> <p>These options can be set using a command-line interface.</p>	See “About rolling over unified log files” on page 24.

Table 1-1 Log retention options in NetBackup (continued)

Log retention option	Use this option...	Reference link
Keep logs for days	To limit the days for which NetBackup retains legacy logs. Logs are deleted after this configuration value is reached. NetBackup Administration Console > NetBackup Management > Host Properties > Logging dialog box.	See “About limiting the size and the retention of legacy logs” on page 47.
MAX_LOGFILE_SIZE and MAX_NUM_LOGFILES	To limit the legacy log size and the number of legacy log files to be retained. These options can be set using a command-line interface.	See “Configuring the legacy log rotation” on page 49.

Note: Before you enable logging for critical NetBackup processes, review the log retention options and select them appropriately.

About limiting the size of unified and legacy logs

To limit the size of the NetBackup logs, specify the log size in the **Keep logs up to GB** option in the NetBackup Administration Console. When the NetBackup log size grows up to this configuration value, the older logs are deleted. To set the log size in GB, select the check box, which lets you select the value in GB from the drop-down list.

See [“About log retention in NetBackup”](#) on page 10.

You can specify the **Keep logs up to GB** setting under **Host Properties** in the **Logging** dialog box in the NetBackup Administration Console.

Note: You can verify the log pruning behavior in NetBackup by creating the following directories:

On Windows: `install_path\NetBackup\logs\nbutils`

On UNIX: `/usr/opensv/netbackup/logs/nbutils`

About unified logging

Unified logging and legacy logging are the two forms of debug logging used in NetBackup. All NetBackup processes use one of these forms of logging. Server processes and client processes use unified logging.

Unified logging creates log file names and messages in a standardized format. These logging files cannot be easily viewed with a text editor. They are in binary format and some of the information is contained in an associated resource file. Only the `vxlogview` command can assemble and display the log information correctly.

See [“Originator IDs for the entities that use unified logging”](#) on page 17.

Unlike legacy logging, unified logging does not require that you create logging subdirectories. Log files for originator IDs are written to a subdirectory with the name specified in the log configuration file. All unified logs are written to subdirectories in the following directory:

Windows `install_path\NetBackup\logs`

UNIX `/usr/opensv/logs`

You can access logging controls in the **NetBackup Administration Console**. In the left pane, expand **NetBackup Management > Host Properties > Master Servers** or **Media Servers**. Double-click the server you want to change. In the left pane of the dialog box, click **Logging**.

You can also manage unified logging by using the following commands:

<code>vxlogcfg</code>	Modifies the unified logging configuration settings. See “Examples of using vxlogcfg to configure unified logs” on page 34.
<code>vxlogmgr</code>	Manages the log files that the products that support unified logging generate. See “Examples of using vxlogmgr to manage unified logs” on page 31.
<code>vxlogview</code>	Displays the logs that unified logging generates. See “Examples of using vxlogview to view unified logs” on page 30.

These commands are located in the following directory:

Windows `install_path\NetBackup\bin`

UNIX `/usr/opensv/netbackup/bin`

See the [NetBackup Commands Reference Guide](#) for a complete description about these commands.

More information about legacy logging is available.

See [“About legacy logging”](#) on page 36.

Gathering unified logs for NetBackup

This topic uses an example to describe how to gather unified logs for NetBackup.

To gather unified logs for NetBackup

- 1 Create a directory named `/upload` by using the following command.

```
# mkdir /upload
```

- 2 Copy unified logs (for NetBackup only) to the `/upload` directory by using the following command:

```
# vxlogmgr -p NB -c --dir /upload
```

Example output:

Following are the files that were found:

```
/usr/opensv/logs/bmrsetup/51216-157-2202872032-050125-0000000.log  
/usr/opensv/logs/nbemmm/51216-111-2202872032-050125-0000000.log  
/usr/opensv/logs/nbrb/51216-118-2202872032-050125-0000000.log  
/usr/opensv/logs/nbjm/51216-117-2202872032-050125-0000000.log  
/usr/opensv/logs/nbpem/51216-116-2202872032-050125-0000000.log  
/usr/opensv/logs/nbsl/51216-132-2202872032-050125-0000000.log
```

Total 6 file(s)

Copying

```
/usr/opensv/logs/bmrsetup/51216-157-2202872032-050125-0000000.log ...
```

Copying

```
/usr/opensv/logs/nbemmm/51216-111-2202872032-050125-0000000.log ...
```

Copying

```
/usr/opensv/logs/nbrb/51216-118-2202872032-050125-0000000.log ...
```

Copying

```
/usr/opensv/logs/nbjm/51216-117-2202872032-050125-0000000.log ...
```

Copying

```
/usr/opensv/logs/nbpem/51216-116-2202872032-050125-0000000.log ...
```

Copying

```
/usr/opensv/logs/nbsl/51216-132-2202872032-050125-0000000.log ...
```

3 Change to the `/upload` directory and list its contents.

```
# cd /upload
ls
```

Example output:

```
51216-111-2202872032-050125-0000000.log
51216-116-2202872032-050125-0000000.log
51216-117-2202872032-050125-0000000.log
51216-118-2202872032-050125-0000000.log
51216-132-2202872032-050125-0000000.log
51216-157-2202872032-050125-0000000.log
```

4 Tar the log files.

```
# tar -cvf file_name.logs ./*
```

Types of unified logging messages

The following message types can appear in unified logging files:

Application log messages

Application log messages include informational, warning, and error messages. They are always logged and cannot be disabled. These messages are localized.

An example of an application message follows:

```
12/04/2015 15:48:54.101 [Application] NB
51216 nbjm 117 PID:5483 TID:14 File
ID:117 [reqid=-1446587750] [Info]
V-117-40 BPBRM pid = 17446
```

Diagnostic log messages

Diagnostic log messages are the unified logging equivalent of the legacy debug log messages. They can be issued at various levels of detail (similar to verbose levels in legacy logging). These messages are localized.

Diagnostic messages can be disabled with the `vxlogcfg` command.

An example of a diagnostic message follows:

```
12/04/2015 15:48:54.608 [Diagnostic] NB
51216 nbjm 117 PID:5483 TID:14 File
ID:117 [No context] 3 V-117-298
[JobInst_i:requestResourcesWithTimeout]
callback object timeout=600
```

Debug log messages

Debug log messages are intended primarily for Veritas engineering. Like diagnostic messages, they can be issued at various levels of detail. These messages are not localized.

Debug messages can be disabled with the `vxlogcfg` command.

An example of a debug message follows:

```
12/04/2015 15:48:56.982 [Debug] NB
51216 nbjm 117 PID:5483 TID:14 File
ID:117 [jobid=2 parentid=1] 1
[BackupJob::start()] no pending proxy
requests, start the job
```

File name format for unified logging

Unified logging uses a standardized naming format for log files. The following is an example of a log file name.

```
/usr/opensv/logs/nbpem/51216-116-2201360136-041029-0000000000.log
```

[Table 1-2](#) describes each part of the log file name.

Table 1-2 Description of the file name format for unified logging

Example	Description	Details
51216	Product ID	Identifies the product. The NetBackup product ID is 51216. The product ID is also known as the entity ID.
116	Originator ID	Identifies the log writing entity, such as a process, service, script, or other software. The number 116 is the originator ID of the <code>nbpem</code> process (the NetBackup policy execution manager).
2201360136	Host ID	Identifies the host that created the log file. Unless the file was moved, this ID is the host where the log resides.
041029	Date	Shows the date when the log was written in YYMMDD format.
0000000000	Rotation	Identifies the numbered instance of a log file for a given originator. The rollover number (rotation) indicates the instance of this log file. By default, log files roll over (rotate) based on file size. If the file reaches maximum size and a new log file is created for this originator, the new file is designated 0000000001. See “About rolling over unified log files” on page 24.

The log configuration file specifies the name of the directories where the log files for originator IDs are written. These directories and the log files that they hold are written to the following directory, except as noted in the following:

See [“Originator IDs for the entities that use unified logging”](#) on page 17.

Windows `install_path\NetBackup\logs`

UNIX `/usr/opensv/logs`

Originator IDs for the entities that use unified logging

Many server processes, services, and libraries use unified logging. Also, UNIX and Windows clients use unified logging. An originator identifier (OID) corresponds to a NetBackup process, service, or library.

An OID identifies a process, a service, or a library. A process creates entries in its own log file. The process can call a library that also creates entries in the same file but with an OID unique to the library. Hence, a log file can contain entries with different OIDs. Multiple processes can use the same library, so a library OID can appear in several different log files.

[Table 1-3](#) lists the NetBackup server and NetBackup client processes, services, and libraries that use unified logging.

Table 1-3 Originator IDs for the server entities that use unified logging

Originator ID	Entity	Description
18	nbatd	The authentication service (nbatd) is a service (daemon) that verifies the user identity and issues credentials. These credentials are used for Secure Sockets Layer (SSL) communication. The (nbatd) directory is created under the <code>usr/netbackup/sec/at/bin</code> directory (UNIX) or the <code>install_path\NetBackup\sec\at\bin</code> directory (Windows).
103	pbx_exchange	The Private Branch Exchange (PBX) service provides single-port access to clients outside the firewall that connect to Veritas product services. Service name: <code>VRTSspb</code> . It writes logs to <code>/opt/VRTSspb/log</code> (UNIX) or <code>install_path\VxPBX\log</code> (Windows). The PBX product ID is 50936.
111	nbemm	The Enterprise Media Manager (EMM) is a NetBackup service that manages the device and the media information for NetBackup. It runs only on the master server.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
116	nbpem	The NetBackup Policy Execution Manager (<i>nbpem</i>) creates policy and client tasks and determines when jobs are due to run. It runs only on the master server.
117	nbjm	The NetBackup Job Manager (<i>nbjm</i>) accepts the jobs that the Policy Execution Manager submits and acquires the necessary resources. It runs only on the master server.
118	nbrb	The NetBackup Resource Broker (<i>nbrb</i>) maintains a cache list of available resources. It uses that list to locate the physical and the logical resources that are required for a backup or a tape restore. It initiates a SQL call to <i>nbemm</i> to update the database, and then passes the allocation information to <i>nbjm</i> . It runs only on the master server.
119	bmrdb	The NetBackup Bare Metal Restore (BMR) master server daemon.
121	bmrsavecfg	The BMR Save Configuration is a data collection utility that runs on the NetBackup client, not the server.
122	bmrcl	The BMR Client Utility originates on the BMR boot server and runs on the restoring client. UNIX clients use it to communicate to the BMR master server during a restore.
123	bmrsv	The BMR Server Utility.
124	bmrcreatefloppy	The BMR commands that create floppy disks use the BMR Create Floppy utility. The utility runs on the BMR boot server and is Windows only.
125	bmrstrt	The BMR Create SRT utility creates a shared resource tree. It runs on the BMR boot server.
126	bmrprep	The BMR Prepare to Restore utility prepares the BMR servers for a client restoration.
127	bmrsetup	The BMR Setup Commands utility sets up BMR installation, configuration, and upgrade processes.
128	bmrcommon	The BMR Libraries and Common Code catalog provides log messages to the BMR libraries.
129	bmrconfig	The BMR Edit Configuration utility modifies the client configuration.
130	bmrcreatepkg	The BMR Create Package utility adds Windows drivers, service packs, and hotfixes to the BMR master server for restore operations.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
131	<code>bmrrst</code>	The BMR Restore utility restores Windows BMR clients. It runs on the restoring client for Windows systems only.
132	<code>nbsl</code>	The NetBackup Service Layer facilitates the communication between the NetBackup graphical user interface and NetBackup logic. <code>nbsl</code> is required to run NetBackup OpsCenter, an application that manages and monitors multiple NetBackup environments. This process runs only on the master server.
134	<code>ndmpagent</code>	The NDMP agent daemon manages NDMP backups and restores. It runs on the media server.
137	<code>libraries</code>	The libraries control the logging level in the NetBackup libraries. The application and the diagnostic messages are for customer use; the debug messages are intended for Veritas engineering.
140	<code>mmui</code>	The media server user interface is used for the Enterprise Media Manager (EMM).
142	<code>bmrepadm</code>	The BMR External Procedure process manages the BMR external procedures that are used during a restore operation.
143	<code>mds</code>	The EMM Media and Device Selection process manages the media selection component and device selection component of the Enterprise Media Manager (EMM).
144	<code>da</code>	The EMM Device Allocator is used for shared drives.
146	<code>NOMTRS</code>	The NetBackup OpsCenter reporting service is part of NetBackup OpsCenter.
147	<code>NOMClient</code>	The NetBackup OpsCenter Client is part of NetBackup OpsCenter.
148	<code>NOMServer</code>	The NetBackup OpsCenter Server is part of NetBackup OpsCenter
151	<code>ndmp</code>	The NDMP message log (<code>ndmp</code>) handles NDMP protocol messages, <code>avrd</code> , and robotic processes.
154	<code>bmrovradm</code>	The BMR Override Table Admin Utility manages the custom override functions for Bare Metal Restore.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
156	ace	<p>The NBACE process controls the logging level in the (ACE/TAO) CORBA components for any process that uses a CORBA interface. The default level is 0 (only important messages are logged). This logging is intended for Veritas engineering.</p> <p>If Veritas Technical Support instructs you to increase the logging level, increase the level for originator ID 137 to 4 or higher.</p> <p>Warning: A debug logging level greater than 0 generates large amounts of data.</p>
158	ncfrai	Remote access interface for NetBackup clients.
159	ncftfi	Transporter for NetBackup clients.
163	nbsvcmon	The NetBackup Service Monitor monitors the NetBackup services that run on the local computer and tries to restart a service that unexpectedly terminates.
166	nbvault	The NetBackup Vault Manager manages NetBackup Vault. <code>nbvault</code> must be running on the NetBackup Vault server during all NetBackup Vault operations.
178	dsm	The Disk Service Manager (DSM) performs set and get operations on disk storage and disk storage units.
199	nbftsrvr	The Fibre Transport (FT) server process runs on the media servers that are configured for the NetBackup Fibre Transport. On the server side of the FT connection, <code>nbftsrvr</code> controls data flow, processes SCSI commands, manages data buffers, and manages the target mode driver for the host bus adapters. <code>nbftsrvr</code> is part of SAN client.
200	nbftclnt	The Fibre Transport (FT) client process runs on the client and is part of SAN Client.
201	fsm	The FT Service Manager (FSM) is a component of the Enterprise Media Manager (EMM) and is part of SAN Client.
202	stssvc	The Storage service manages the storage server and runs on the media server.
210	ncfive	Exchange Firedrill Wizard for NetBackup clients.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
219	rsrcevtmgr	The Resource Event Manager (REM) is a CORBA loadable service that runs inside nbemm. REM works with the Disk Polling Service to monitor free space and volume status, and to watch for disk-full conditions.
220	dps	Disk polling service for NetBackup clients.
221	mpms	The Media Performance Monitor Service (MPMS) runs on every media server within RMMS and gathers CPU load and free memory information for the host.
222	nbrmms	Remote monitoring and Management Service (RMMS) is the conduit through which EMM discovers and configures disk storage on media servers.
226	nbstserv	The Storage services controls the lifecycle image duplication operations.
230	rdsm	The Remote Disk Service Manager interface (RDMS) runs within the Remote Manager and Monitor Service. RDMS runs on media servers.
231	nbevtmgr	The Event Manager Service provides asynchronous event management services for cooperating participants.
248	bmrlauncher	The BMR Launcher Utility in the Windows BMR Fast Restore image configures the BMR environment.
254	SPSV2RecoveryAsst	Recovery Assistant for SharePoint Portal Server for NetBackup clients.
261	aggs	Artifact Generator Generated Source.
263	wingui	The NetBackup Administration Console for Windows
271	nbecmsg	Legacy error codes.
272	expmgr	The Expiration Manager handles the capacity management and the image expiration for storage lifecycle operations.
286	nbkms	The Encryption Key Management Service is a master server-based symmetric service that provides encryption keys to the media server NetBackup Tape Manager processes.
293	nbaudit	NetBackup Audit Manager.
294	nbauditmsgs	NetBackup Audit Messages.
309	ncf	NetBackup Client Framework.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
311	ncfnbservercom	NetBackup Client/Server Communications.
317	ncfbedspi	NetBackup Client Beds Plug-in.
318	ncfwinpi	NetBackup Client Windows Plug-in.
321	dbaccess	NetBackup Relational Database access library.
348	ncforaclepi	NetBackup Client Oracle Plug-in.
351	ncflbc	Live Browse Client.
352	ncfgre	Granular restore.
355	ncftarpi	NetBackup TAR Plug-in.
356	ncfvxmspi	NetBackup Client VxMS Plug-in.
357	ncfnbrestore	NetBackup Restore.
359	ncfnbbrowse	NetBackup Browser.
360	ncforautil	NetBackup Client Oracle utility.
361	ncfdb2pi	NetBackup Client DB2 Plug-in.
362	nbars	NetBackup Agent Request Services.
363	dars	Database Agent Request Server process call
366	ncfnbcs	NetBackup Client Service.
369	impmgr	NetBackup Import Manager.
371	nbim	Indexing manager.
372	nbhsm	Hold service.
375	ncfnbsearchserverpi	NetBackup Client Search Server Plug-in.
377	ncfnbdiscover	NetBackup Client Component Discovery.
380	ncfnbquiescence	NetBackup Client Component Quiescence/Unquiescence.
381	ncfnbdboffline	NetBackup Client Component Offline/Online.

Table 1-3 Originator IDs for the server entities that use unified logging
(continued)

Originator ID	Entity	Description
386	ncfvmwarepi	NetBackup NCF VMware Plug-in.
387	nbrntd	NetBackup Remote Network Transport. If multiple backup streams run concurrently, the Remote Network Transport Service writes a large amount of information to the log files. In such a scenario, set the logging level for OID 387 to 2 or less. See “Changing the logging level” on page 52.
395	stsem	STS Event Manager.
396	nbutils	NetBackup Utilities.
400	nbdisco	NetBackup Discovery.
401	ncfmssqlpi	NetBackup Client MSSQL plug-in.
402	ncfexchangeapi	NetBackup Client Exchange plug-in.
403	ncfsharepointpi	NetBackup Client SharePoint plug-in.
412	ncffilesyspi	NetBackup Client File System plug-in.
480	libvcloudsuite	NetBackup vCloudSuite Library.
486	nbpxyhelper	The <code>vnetd</code> proxy helper process.
490	nbpxytnl	The HTTP tunnel of the <code>vnetd</code> proxy.

About changing the location of unified log files

The unified logging files can consume a lot of disk space. If necessary, enter the following to direct them to a different location.

UNIX `/usr/opensv/netbackup/bin/vxlogcfg -a -p NB -o Default -s LogDirectory=new_log_path`

Where `new_log_path` is a full path, such as `/bigdisk/logs`.

Windows `install_path\NetBackup\bin\vxlogcfg -a -p NB -o Default -s LogDirectory=new_log_path`

Where `new_log_path` is a full path, such as `D:\logs`.

About rolling over unified log files

To prevent log files from becoming too large, or to control when or how often logs are created, you can set a log rollover option. When a file size or time setting is reached, the current log file is closed. New log messages for the logging process are written or “rolled over” to a new log file.

See [“About log retention in NetBackup”](#) on page 10.

You can set log file rollover to occur based on file size, time of day, or elapsed time. Set the conditions by using the `vxlogcfg` command with the options described in [Table 1-4](#).

Table 1-4 vxlogcfg options that control the rollover of the unified log files

Option	Description
MaxLogFileSizeKB	Specifies the maximum size that is allowed for the log file (in kilobytes) before rollover occurs, if the <code>RolloverMode</code> is set to <code>FileSize</code> .
RolloverAtLocalTime	Specifies the time of day at which the log file is rolled over, if the <code>RolloverMode</code> is set to <code>LocalTime</code> .
RolloverPeriodInSeconds	Specifies a period of time in seconds after which the log file is rolled over, if the <code>RolloverMode</code> is set to <code>Periodic</code> .
MaxLogFileSizeKB or RolloverAtLocalTime	<p>Specifies that the log file rollover occurs whenever the file size limit or the local time limit is reached, whichever is first.</p> <p>An example of the command:</p> <pre>vxlogcfg -a -p 51216 -g Default MaxLogFileSizeKB=256 RolloverAtLocalTime=22:00</pre>
MaxLogFileSizeKB or RolloverPeriodInSeconds	Specifies that the log file rollover occurs whenever the file size limit or the periodic time limit is reached, whichever is first.

A complete description of `vxlogcfg` is in the [NetBackup Commands Reference Guide](#).

By default, log file rollover is based on a file size of 51200 KB. When a log file reaches 51200 KB in size, the file closes and a new log file opens.

The following example sets the NetBackup (`prodid 51216`) rollover mode to `Periodic`.


```
# vxlogcfg -a --prodid 51216 --orgid 116 -s RolloverMode=Periodic  
RolloverPeriodInSeconds=86400
```

The previous example uses the `vxlogcfg` command with the `RolloverMode` option. It sets rollover mode for `nbpem` (originator ID 116) to `Periodic`. It also sets the interval until the next `nbpem` log file rollover to 24 hours (86400 seconds).

In the following example, the file names show the log file rollover with the rotation ID incremented:

```
/usr/opensv/logs/nbpem/51216-116-2201360136-041029-0000000000.log  
/usr/opensv/logs/nbpem/51216-116-2201360136-041029-0000000001.log  
/usr/opensv/logs/nbpem/51216-116-2201360136-041029-0000000002.log
```

In addition, you can use log file rotation with the following:

- Logs for the server processes that use unified logging
See [“Originator IDs for the entities that use unified logging”](#) on page 17.
- Certain legacy logs
- The unified logging files that the Bare Metal Restore process `bmrsavecfg` creates

About recycling unified log files

Deleting the oldest log files is referred to as recycling. You can recycle unified logging files in the following ways.

See [“About log retention in NetBackup”](#) on page 10.

Limit the number of log files	Specify the maximum number of log files that NetBackup retains. When the number of log files exceeds the maximum, the oldest log files become eligible for deletion during log cleanup. The <code>NumberOfLogFiles</code> option for the <code>vxlogcfg</code> command defines that number.
-------------------------------	---

In the following example, the maximum number of log files that are allowed for each of the unified logging originators in the NetBackup (product ID 51216) is 8000. When the number of log files exceeds 8000 for a particular originator, the oldest log files become eligible for deletion during log cleanup.

```
# vxlogcfg -a -p 51216 -o ALL -s  
NumberOfLogFiles=8000
```

See [“Examples of using vxlogcfg to configure unified logs”](#) on page 34.

Specify the number of days the log files are kept Use the **Keep logs for days** property to specify the maximum number of days logs are kept. When the maximum number of days is reached, the unified logs and legacy logs are automatically deleted.

In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Host Properties > Master Servers**. Double-click the server you want to change. A new dialog box appears. In the left pane, click **Logging > Keep logs for days**.

Explicitly delete the log files To initiate recycling and delete the log files, run the following command:

```
# vxlogmgr -a -d
```

If you cannot manually delete or move files with `vxlogmgr`, the **Keep logs for days** property removes the old logs for both unified logging and legacy logging.

See [“Examples of using vxlogmgr to manage unified logs”](#) on page 31.

If the `vxlogcfg LogRecycle` option is ON (true), the **Keep logs for days** setting is disabled for unified logs. In this case, unified logging files are deleted when their number (for a particular originator) exceeds the number that the `NumberOfLogFiles` option specifies on the `vxlogcfg` command.

About using the vxlogview command to view unified logs

Use the `vxlogview` command to view the logs that unified logging creates. These logs are stored in the following directory.

UNIX `/usr/opensv/logs`

Windows `install_path\NetBackup\logs`

Unlike the files that are written in legacy logging, unified logging files cannot be easily viewed with a text editor. The unified logging files are in binary format, and some of the information is contained in an associated resource file. Only the `vxlogview` command can assemble and display the log information correctly.

You can use `vxlogview` to view NetBackup log files as well as PBX log files.

To view PBX logs using the `vxlogview` command, do the following:

- Ensure that you are an authorized user. For UNIX and Linux, you must have root privileges. For Windows, you must have administrator privileges.

- To specify the PBX product ID, enter `-p 50936` as a parameter on the `vxlogview` command line.

`vxlogview` searches all the files, which can be a slow process. Refer to the following topic for an example of how to display results faster by restricting the search to the files of a specific process.

About query strings used with the `vxlogview` command

Use the `vxlogview` command to display the logs that unified logging generates.

The `vxlogview` command includes the following option: `-w (-where)`

QueryString.

QueryString represents a text expression similar to a database WHERE clause. The query string expression is used to retrieve log entries from the unified logging system. The expression is a combination of relational operators, constant integers, constant strings, and names of log fields that evaluate to a single value. Expressions are grouped by logical operators such as AND and OR.

The supported relational operators are as follows:

<	less than
>	greater than
<=	less than and equal to
>=	greater than and equal to
=	equal to
!=	not equal to

The supported logical operators are as follows:

&&	logical AND
	logical OR

Table 1-5 shows data types for specific fields as well as description and an example. When more than one example is listed, both examples produce the same results.

Table 1-5 Data types for fields

Field name	Type	Description	Example
PRODID	Integer or string	Provide the product ID or the abbreviated name of product.	PRODID = 51216 PRODID = 'NBU'
ORGID	Integer or string	Provide the originator ID or the abbreviated name of the component.	ORGID = 116 ORGID = 'nbpem'
PID	Long Integer	Provide the process ID	PID = 1234567
TID	Long Integer	Provide the thread ID	TID = 2874950
STDATE	Long Integer or string	Provide the start date in seconds or in the locale-specific short date and time format. For example, a locale can have the format 'mm/dd/yy hh:mm:ss AM/PM'	STDATE = 98736352 STDATE = '4/26/11 11:01:00 AM'
ENDATE	Long Integer or string	Provide the end date in seconds or in the locale-specific short date and time format. For example, a locale can have the format 'mm/dd/yy hh:mm:ss AM/PM'	ENDATE = 99736352 ENDATE = '04/27/11 10:01:00 AM'
PREVTIME	String	Provide the hours in 'hh:mm:ss' format. This field should be used only with operators =, <, >, >=, and <=	PREVTIME = '2:34:00'
SEV	Integer	Provide one of the following possible severity types: 0 = INFO 1 = WARNING 2 = ERR 3 = CRIT 4 = EMERG	SEV = 0 SEV = INFO

Table 1-5 Data types for fields (*continued*)

Field name	Type	Description	Example
MSGTYPE	Integer	Provide one of the following possible message types: 0 = DEBUG (debug messages) 1 = DIAG (diagnostic messages) 2 = APP (application messages) 3 = CTX (context messages) 4 = AUDIT (audit messages)	MSGTYPE = 1 MSGTYPE = DIAG
CTX	Integer or string	Provide the context token as string identifier or 'ALL' to get all the context instances to be displayed. This field should be used only with the operators = and !=.	CTX = 78 CTX = 'ALL'

Consider the following when writing a query string.

- | | |
|------------------|---|
| Case sensitivity | Field names, severity types, and message types are not case-sensitive. For example, the following are valid entries: <ul style="list-style-type: none"> ■ sev = info ■ msgtype = diag |
| String constants | String constants should be given in single quotes. For example, PRODID = 'NBU' |
| Dates | Start and end dates can be provided in the following formats: <ul style="list-style-type: none"> ■ A string constant that corresponds to the regional display short date format ■ A UNIX long value of number of seconds that elapsed since midnight January 1, 1970. |

Table 1-6 provides examples of query strings.

Table 1-6 Examples of query strings

Example	Description
(PRODID == 51216) && ((PID == 178964) ((STDATE == '2/5/15 09:00:00 AM') && (ENDDATE == '2/5/15 12:00:00 PM'))	Retrieves the log file message for the NetBackup product ID 51216 between 9AM and 12PM on 2015-05-02.

Table 1-6 Examples of query strings (*continued*)

Example	Description
<pre>((prodid = 'NBU') && ((stdate >= '11/18/14 00:00:00 AM') && (enddate <= '12/13/14 12:00:00 PM')) ((prodid = 'BENT') && ((stdate >= '12/12/14 00:00:00 AM') && (enddate <= '12/25/14 12:00:00 PM')))</pre>	Retrieves the log messages for the NetBackup product NBU between 2014-18-11 and 2014-13-12 and the log messages for the NetBackup product BENT between 2014-12-12 and 2014-25-12.
<pre>(STDATE <= '04/05/15 0:0:0 AM')</pre>	Retrieves the log messages that were logged on or before 2015-05-04 for all of the installed Veritas products.

Examples of using vxlogview to view unified logs

The following examples demonstrate how to use the `vxlogview` command to view unified logs.

Table 1-7 Example uses of the vxlogview command

Item	Example
Display all the attributes of the log messages	<code>vxlogview -p 51216 -d all</code>
Display specific attributes of the log messages	<p>Display the log messages for NetBackup (51216) that show only the date, time, message type, and message text:</p> <pre>vxlogview --prodid 51216 --display D,T,m,x</pre>
Display the latest log messages	<p>Display the log messages for originator 116 (nbpem) that were issued during the last 20 minutes. Note that you can specify <code>-o nbpem</code> instead of <code>-o 116</code>:</p> <pre># vxlogview -o 116 -t 00:20:00</pre>
Display the log messages from a specific time period	<p>Display the log messages for <code>nbpem</code> that were issued during the specified time period:</p> <pre># vxlogview -o nbpem -b "05/03/15 06:51:48 AM" -e "05/03/15 06:52:48 AM"</pre>

Table 1-7 Example uses of the vxlogview command (*continued*)

Item	Example
Display results faster	<p>You can use the <code>-i</code> option to specify an originator for a process:</p> <pre># vxlogview -i nbpem</pre> <p>The <code>vxlogview -i</code> option searches only the log files that the specified process (<code>nbpem</code>) creates. By limiting the log files that it has to search, <code>vxlogview</code> returns a result faster. By comparison, the <code>vxlogview -o</code> option searches all unified log files for the messages that the specified process has logged.</p> <p>Note: If you use the <code>-i</code> option with a process that is not a service, <code>vxlogview</code> returns the message "No log files found." A process that is not a service has no originator ID in the file name. In this case, use the <code>-o</code> option instead of the <code>-i</code> option.</p> <p>The <code>-i</code> option displays entries for all OIDs that are part of that process including libraries (137, 156, 309, etc.).</p>
Search for a job ID	<p>You can search the logs for a particular job ID:</p> <pre># vxlogview -i nbpem grep "jobid=job_ID"</pre> <p>The <code>jobid=</code> search key should contain no spaces and must be lowercase.</p> <p>When searching for a job ID, you can use any <code>vxlogview</code> command option. This example uses the <code>-i</code> option with the name of the process (<code>nbpem</code>). The command returns only the log entries that contain the job ID. It misses related entries for the job that do not explicitly contain the <code>jobid=job_ID</code>.</p>

See the *NetBackup Commands Reference Guide* for a complete description of the `vxlogview` command. The guide is available through the following URL:

<http://www.veritas.com/docs/DOC5332>

Examples of using vxlogmgr to manage unified logs

The following examples show how to use the `vxlogmgr` command to manage unified logging files. Log file management includes actions such as deleting or moving the log files.

Table 1-8 Example uses of the vxlogmgr command

Item	Example
List the log files	<p>List all unified log files for the <code>nbrb</code> service:</p> <pre># vxlogmgr -s -o nbrb /usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log /usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log /usr/openv/logs/nbrb/51216-118-1342895976-050505-00.log Total 3 file(s)</pre>
Delete the oldest log files	<p>If the <code>vxlogcfg NumberOfLogFiles</code> option is set to 1, the following example deletes the two oldest log files for the <code>nbrb</code> service:</p> <pre># vxlogcfg -a -p 51216 -o nbrb -s NumberOfLogFiles=1 # vxlogmgr -d -o nbrb -a Following are the files that were found: /usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log /usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log Total 2 file(s) Are you sure you want to delete the file(s)? (Y/N): Y Deleting /usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log ... Deleting /usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log ...</pre>
Delete the newest log files	<p>Delete all the unified log files that NetBackup created in the last 15 days:</p> <pre># vxlogmgr -d --prodid 51216 -n 15</pre> <p>Make sure that you roll over (rotate) the log files before you recycle them.</p>
Delete the log files for a specific originator	<p>Delete all unified log files for originator <code>nbrb</code>:</p> <pre># vxlogmgr -d -o nbrb</pre> <p>Make sure that you roll over (rotate) the log files before you recycle them.</p>
Delete all the log files	<p>Delete all unified log files for NetBackup:</p> <pre># vxlogmgr -d -p NB</pre> <p>Make sure that you roll over (rotate) the log files before you recycle them.</p>

Table 1-8 Example uses of the `vxlogmgr` command (*continued*)

Item	Example
Control the number of log files	<p>You can use the <code>vxlogmgr</code> command with the <code>vxlogcfg</code> command's <code>NumberOfLogFiles</code> option to manually delete log files.</p> <p>For example, the <code>NumberOfLogFiles</code> option is set to 2, you have 10 unified logging files, and cleanup has not occurred. Enter the following to keep the two most recent log files and delete the rest for all originators:</p> <pre># vxlogmgr -a -d</pre> <p>The following command keeps the two most recent log files of all PBX originators:</p> <pre># vxlogmgr -a -d -p ics</pre> <p>The following deletes the older log files for the <code>nbrb</code> service only:</p> <pre># vxlogmgr -a -d -o nbrb</pre>
Control disk space usage	<p>Periodically run the <code>vxlogmgr -a -d</code> command (such as through a <code>cron</code> job) to delete logs and monitor the disk space that unified logging uses.</p> <p>The disk space that a given originator uses can be calculated as follows:</p> $\text{NumberOfFiles for originator} * \text{MaxLogFileSizeKB for originator}$ <p>The total disk space that unified logs consume is the sum of the disk space that each originator consumes. If none of the originators override the <code>NumberOfFiles</code> and <code>MaxLogFileSizeKB</code> settings, then the total disk space that unified logging consumes is as follows:</p> $\text{Number of originators} * \text{default MaxLogFileSizeKB} * \text{default NumberOfFiles}$ <p>Use the <code>vxlogcfg</code> command to list the current unified logging settings.</p> <p>For example, assume the following:</p> <ul style="list-style-type: none"> ■ <code>vxlogmgr -a -d -p NB</code> is configured as a <code>cron</code> job with a frequency of one hour. ■ No originators override default settings for <code>MaxLogFileSizeKB</code> or <code>NumberOfFiles</code>. ■ The number of active NetBackup originators on the host is 10. (Typical of a NetBackup master server that is not running BMR or NDMP.) ■ The default <code>MaxLogFileSizeKB</code> is equal to 51200. ■ The default <code>NumberOfFiles</code> is equal to 3. <p>To calculate the total disk space that unified logging consumes, insert the values from the example into the previous formula. The results are as follows:</p> $10 * 51200 * 3 \text{ KB} = 1,536,000 \text{ KB of additional disk space used each hour.}$

A complete description of `vxlogmgr` is in the [NetBackup Commands Reference Guide](#).

Examples of using vxlogcfg to configure unified logs

Use the `vxlogcfg` command to change logging levels and rollover settings.

The `vxlogcfg` command has the following characteristics:

- The `vxlogcfg` command is the only way to turn off diagnostic and debug messages in unified logging. In legacy logging, the writing of messages cannot be turned off, only minimized.
- Absolute paths must be specified. Do not use relative paths.

The following examples show how to use the `vxlogcfg` command to configure unified logging settings.

Table 1-9 Example uses of the `vxlogcfg` command

Item	Example
Set the maximum log file size	<p>By default, the maximum log file size in unified logging is 51200 KB. When a log file reaches 51200 KB, the file closes and a new log file opens.</p> <p>You can change the maximum file size with the <code>MaxLogFileSizeKB</code> option. The following command changes the default maximum log size to 100000 KB for the NetBackup product:</p> <pre># vxlogcfg -a -p 51216 -o Default -s MaxLogFileSizeKB=100000</pre> <p>For <code>MaxLogFileSizeKB</code> to be effective, the <code>RolloverMode</code> option must be set to <code>FileSize</code>:</p> <pre># vxlogcfg -a --prodid 51216 --orgid Default -s RolloverMode=FileSize</pre> <p><code>MaxLogFileSizeKB</code> can be set per originator. An originator that is not configured uses the default value. The following example overrides the default value for service <code>nbrb</code> (originator ID 118).</p> <pre># vxlogcfg -a -p 51216 -o nbrb -s MaxLogFileSizeKB=1024000</pre>

Table 1-9 Example uses of the vxlogcfg command (*continued*)

Item	Example
Set log recycling	<p>The following example sets automatic log file deletion for nbemm logs (originator ID 111):</p> <pre># vxlogcfg -a --prodid 51216 --orgid 111 -s RolloverMode=FileSize MaxLogFileSizeKB=512000 NumberOfLogFiles=999 LogRecycle=TRUE</pre> <p>This example sets nbemm rollover mode to file size, and turns on log recycling. When the number of log files exceeds 999, the oldest log file is deleted. EXAMPLE 5 shows how to control the number of log files.</p>
Set debug level and diagnostic level	<p>The following example sets the default debug level and diagnostic level of product ID NetBackup (51216):</p> <pre># vxlogcfg -a --prodid 51216 --orgid Default -s DebugLevel=1 DiagnosticLevel=6</pre>

Table 1-9 Example uses of the vxlogcfg command (*continued*)

Item	Example
List the unified logging settings	<p>The following vxlogcfg example shows how to list the active unified logging settings for a given originator (the nbrb service). Note that MaxLogFileSizeKB, NumberOfLogFiles, and RolloverMode are included in the output.</p> <pre># vxlogcfg -l -o nbrb -p NB</pre> <p>Configuration settings for originator 118, of product 51,216...</p> <pre>LogDirectory = /usr/openv/logs/nbrb/ DebugLevel = 1 DiagnosticLevel = 6 DynaReloadInSec = 0 LogToStdout = False LogToStderr = False LogToOslog = False RolloverMode = FileSize LocalTime LogRecycle = False MaxLogFileSizeKB = 51200 RolloverPeriodInSeconds = 43200 RolloverAtLocalTime = 0:00 NumberOfLogFiles = 3 OIDNames = nbrb AppMsgLogging = ON L10nLib = /usr/openv/lib/libvxexticu L10nResource = nbrb L10nResourceDir = /usr/openv/resources SyslogIdent = VRTS-NB SyslogOpt = 0 SyslogFacility = LOG_LOCAL5 LogFilePermissions = 664</pre>

A complete description of vxlogcfg is in the [NetBackup Commands Reference Guide](#).

About legacy logging

Legacy logging and unified logging are the two forms of debug logging used in NetBackup. All NetBackup processes use either unified logging or legacy logging.

See “[About unified logging](#)” on page 13.

In legacy debug logging, each process creates log files of debug activity in its own logging directory. The NetBackup legacy debug log directories are located in the following directories:

Windows	<code>install_path\NetBackup\logs</code> <code>install_path\Volmgr\debug</code>
UNIX	<code>/usr/opensv/netbackup/logs</code> <code>/usr/opensv/volmgr/debug</code>

These top-level directories can contain a directory for each NetBackup process that uses legacy logging. By default, NetBackup creates only a subset of all of the possible log directories. For example, the following directories are created by default on UNIX servers:

- `nbfsp`
- `nbliveup`
- `nblogadm`
- `user_ops`

To enable logging for all of the NetBackup processes that use legacy logging, you must create the log file directories that do not already exist, unless you use the Logging Assistant. For more information about the Logging Assistant, see the *NetBackup Administrator's Guide, Volume I*. The guide is available at the following location:

<http://www.veritas.com/docs/DOC5332>

See “Directory names for legacy debug logs for servers ” on page 43.

See “Directory names for legacy debug logs for media and device management” on page 45.

You can use the following batch files to create all of the debug log directories at once:

- Windows: `install_path\NetBackup\Logs\mklogdir.bat`
- UNIX: `usr/opensv/netbackup/logs/mklogdir`

See the *NetBackup Commands Reference Guide* for a complete description about the `mklogdir` command. The guide is available at the following location:

<http://www.veritas.com/docs/DOC5332>

After the directories are created, NetBackup creates log files in the directory that is associated with each process. A debug log file is created when the process

begins. Each log file grows to a certain size before the NetBackup process closes it and creates a new log file.

See “[File name format for legacy logging](#)” on page 42.

To enable debug logging for the NetBackup Status Collection Daemon (`vmgcd`), create the following directory before you start `nbemm`.

Windows `install_path\Volmgr\debug\vmgcd\`

UNIX `/usr/opensv/volmgr/debug/vmgcd`

As an alternative, you can restart `vmgcd` after creating the directory.

UNIX client processes that use legacy logging

Many UNIX client processes use legacy logging. To enable legacy debug logging on UNIX clients, create the appropriate subdirectories in the following directory.

You can use the following batch file to create all of the debug log directories at once:

Windows `Install_path\NetBackup\Logs\mklogdir.bat`

UNIX `usr/opensv/netbackup/logs/mklogdir`

[Table 1-10](#) describes the directories for the legacy debug logs that apply to UNIX clients.

Table 1-10 UNIX client processes that use legacy logging

Directory	Associated process
<code>bp</code>	Menu driven client-user interface program.
<code>bparchive</code>	Archive program. Also useful for debugging <code>bp</code> .
<code>bpbackup</code>	Backup program. Also useful for debugging <code>bp</code> .
<code>bpbkcar</code>	Program that is used to generate backup images.
<code>bpcd</code>	NetBackup client daemon or manager.
<code>bpclimagelist</code>	Command-line utility that produces a status report on client NetBackup images or removable media.
<code>bpclntcmd</code>	Command-line utility on the clients that test NetBackup system functionality and enables Fibre Transport services.

Table 1-10 UNIX client processes that use legacy logging (*continued*)

Directory	Associated process
bphdb	Program that starts a script to back up a database on a NetBackup database agent client. See the system administrator's guide for the appropriate NetBackup database agent for more information.
bpjava-msvc	The NetBackup Java application server authentication service that <code>inetd</code> starts during the startup of the NetBackup Java interface applications. This program authenticates the user that started the application.
bpjava-usvc	The NetBackup program that <code>bpjava-msvc</code> starts upon successful logon through the logon dialog box that is presented when a NetBackup Java Backup, Archive, and Restore (BAR) interface is started. This program services all requests from the Java user interfaces on the host where <code>bpjava-msvc</code> is running.
bplist	Program that lists backed up and archived files. Also useful to debug <code>bp</code> . On pre-7.6 versions of NetBackup, the <code>bpcIntcmd</code> command and the <code>bpclimagelist</code> command send their debug log messages to the <code>bplist</code> directory. On NetBackup 7.6, <code>bpcIntcmd</code> and <code>bpclimagelist</code> send their debug log messages to the <code>bpcIntcmd</code> and the <code>bpclimagelist</code> directory, respectively.
bpmount	Program that determines the local mount points and wildcard expansion for multiple data streams.
bporaexp	Command-line program on clients to export Oracle data in XML format. Communicates with <code>bprd</code> on the server.
bporaexp64	64-bit command-line program on clients to export Oracle data in XML format. Communicates with <code>bprd</code> on the server.
bporaimp	Command-line program on clients to import Oracle data in XML format. Communicates with <code>bprd</code> on the server.
bporaimp64	64-bit command-line program on clients to import Oracle data in XML format. Communicates with <code>bprd</code> on the server.
bprestore	Restore program. Also useful for debugging <code>bp</code> .
db_log	For more information on these logs, see the NetBackup guide for the database-extension product that you use.
mtfrd	These logs have information about the <code>mtfrd</code> process that is used for phase 2 imports and restores of the Backup Exec media.

Table 1-10 UNIX client processes that use legacy logging (*continued*)

Directory	Associated process
tar	nbtar processing during restores.
user_ops	<p>The <code>user_ops</code> directory is created during the install of NetBackup on all servers and clients. The NetBackup Java interface programs use this directory for temporary files and for job and progress log files that the Backup, Archive, and Restore program (<code>jbpsa</code>) generates. This directory must exist for successful operation of any of the Java programs and must have public read, write, and run permissions. This directory contains a directory for every user that uses the Java programs.</p> <p>In addition, on NetBackup Java capable platforms, the NetBackup Java interface log files are written in a subdirectory that is called <code>nbglogs</code>. All of the files that are in the <code>user_ops</code> directory hierarchy are removed according to the setting of the <code>KEEP_LOGS_DAYS</code> configuration option.</p>

PC client processes that use legacy logging

Most PC client processes use legacy logging. To enable the detailed legacy debug logging on Windows clients, create the directories in the following location. The directory names that you create correspond to the processes to which you want to create logs.

```
C:\Program Files\VERITAS\NetBackup\Logs\
```

Note: These are the default locations in which to place these directories. You can specify another location during client installation.

[Table 1-11](#) lists the legacy debug log directories that apply to these clients.

Table 1-11 PC client processes that use legacy logging

Directory	NetBackup client	Description
bpinetd	All Windows clients	Client service logs. These logs have information on the <code>bpinetd32</code> process.
bparchive	All Windows clients	Archive program that is run from the command line.

Table 1-11 PC client processes that use legacy logging (*continued*)

Directory	NetBackup client	Description
bpbackup	All Windows clients	The backup program that is run from the command line.
bpbkarr	All Windows clients	Backup and archive manager. These logs have information on the <code>bpbkarr32</code> process.
bpcd	All Windows clients	NetBackup client daemon or manager. These logs have information on communications between the server and client.
bpjava-msvc		NetBackup Java application server authentication service that the <code>Client Services</code> service starts during startup of the NetBackup Java interface applications. This program authenticates the user who started the application. (On all Windows platforms.)
bpjava-usvc		NetBackup program that <code>bpjava-msvc</code> starts upon successful logon through the logon dialog box that is presented when a NetBackup Java Backup, Archive, and Restore (BAR) interface is started. This program services all requests from the Java user interfaces on the NetBackup host where <code>bpjava-msvc</code> is running. (On all Windows platforms.)
bplist	All Windows clients	List program that is run from the command line.
bpmount	All Windows clients	The program that is used to collect drive names on the client for multistreaming clients.
bprestore	All Windows clients	The restore program that is run from the command line.
tar	All Windows clients	<code>tar</code> processing. These logs have information about the <code>tar32</code> process.

Table 1-11 PC client processes that use legacy logging (*continued*)

Directory	NetBackup client	Description
<code>user_ops</code>	All Windows clients	<p>The <code>user_ops</code> directory is created during the install of NetBackup on all servers and clients. The NetBackup Java interface programs use it for the following: temporary files and for job and progress log files that the Backup, Archive, and Restore program (<code>jbpsa</code>) generates. This directory must exist for successful operation of any of the Java programs and must have public read, write, and run permissions. <code>user_ops</code> contains a directory for every user that uses the Java programs.</p> <p>In addition, on NetBackup Java-capable platforms, the NetBackup Java interface log files are written in a subdirectory that is called <code>nbglogs</code>. All files in the <code>user_ops</code> directory hierarchy are removed according to the setting of the <code>KEEP_LOGS_DAYS</code> configuration option.</p>

File name format for legacy logging

NetBackup legacy logging creates debug log files in the following format:

`user_name.mmddyy_nnnnn.log`

The following items describe the log file name elements:

- `user_name` The name of the user in whose context the process runs, as follows:
- For UNIX root user, the `user_name` is **root**.
 - For UNIX user other than the root user, the `user_name` is the user's login ID.
 - For all users who are part of the Administrator group in Windows, the `user_name` is `ALL_ADMINS`.
 - For Windows user, the `user_name` is either `username@domain_name` or `username@machine_name`.

<i>mmddyy</i>	The month, day, and year on which NetBackup created the log file.
<i>nnnnn</i>	The counter or the rotation number for the log file. When the counter exceeds the setting for number of log files, the oldest log file is deleted. The <code>MAX_NUM_LOGFILES</code> configuration parameter sets the maximum number of a legacy log file per process.

The retention of all logs files in the legacy debug log directories is managed by using the following options:

- **Keep logs for days** setting of the NetBackup **Host Properties Logging** dialog box. The default is 28 days.
- **Keep logs up to size** setting of the NetBackup **Host Properties Logging** dialog box.
- The legacy logging settings.
See [“About limiting the size and the retention of legacy logs”](#) on page 47.

Any mixture of new and old log file names in a legacy debug log directory is managed according to the **Keep logs** setting and the robust logging settings.

Directory names for legacy debug logs for servers

[Table 1-12](#) describes the directories you need to create to support legacy debug logs for servers. Each directory corresponds to a process. Unless it is noted, each directory should be created under the following directory.

Windows	<code>install_path\NetBackup\logs</code>
UNIX	<code>/usr/opensv/netbackup/logs</code>

Table 1-12 Directory names for legacy debug logs

Directory	Associated process
<code>admin</code>	Administrative commands
<code>bpbrm</code>	NetBackup backup and restore manager
<code>bpcd</code>	NetBackup client daemon or manager. The NetBackup Client service starts this process.
<code>bpjobd</code>	NetBackup jobs database manager program
<code>bpdm</code>	NetBackup disk manager

Table 1-12 Directory names for legacy debug logs (*continued*)

Directory	Associated process
bpdbm	NetBackup Database Manager. This process runs only on master servers. On Windows systems, it is the NetBackup Database Manager service.
bpjava-msvc	<p>The NetBackup Java application server authentication service that is started when the NetBackup interface applications start. On UNIX servers, <code>inetd</code> starts it. On Windows servers, the Client Services service starts it.</p> <p>This program authenticates the user that started the application.</p>
bpjava-susvc	The NetBackup program that <code>bpjava-msvc</code> starts upon successful logon through the logon dialog box that is presented when a NetBackup interface starts. This program services all requests from the Java user interfaces on the NetBackup master or media server host where the <code>bpjava-msvc</code> program runs (all Windows platforms).
bprd	NetBackup request daemon or manager. On Windows systems, this process is called the NetBackup Request Manager service.
bpsynth	The NetBackup process for synthetic backup. <code>nbjm</code> starts <code>bpsynth</code> . <code>bpsynth</code> runs on the master server.
bptm	NetBackup tape management process
nbatd	Authentication daemon (UNIX and Linux) or service (Windows). <code>nbatd</code> authenticates access to interfaces of NetBackup services or daemons.
nbazd	Authorization daemon (UNIX and Linux) or service (Windows). <code>nbazd</code> authorizes access to interfaces of NetBackup services or daemons.
syslogs	<p>System log</p> <p>You must enable system logging to troubleshoot <code>ltid</code> or robotic software. See the <code>syslogd</code> man page.</p>
user_ops	<p>The <code>user_ops</code> directory is created during the install of NetBackup on all servers and clients. NetBackup interface programs use it for the following: temporary files and for job and progress log files that the Backup, Archive, and Restore program (<code>jbpSA</code>) generates. This directory must exist for successful operation of any of the Java programs and must have public read, write, and execute permissions. <code>user_ops</code> contains a directory for every user that uses the Java programs.</p> <p>In addition, on Java-capable platforms, the NetBackup Java interface log files are written in the <code>nbjlogs</code> subdirectory. All files in the <code>user_ops</code> directory hierarchy are removed according to the setting of the <code>KEEP_LOGS_DAYS</code> configuration option.</p>

Table 1-12 Directory names for legacy debug logs (*continued*)

Directory	Associated process
<code>vnetd</code>	<p>The Veritas network daemon, used to create firewall-friendly socket connections. Started by the <code>inetd(1M)</code> process.</p> <p>Note: Logging occurs in either the <code>/usr/opensv/logs</code> directory or the <code>/usr/opensv/netbackup/logs</code> if the <code>vnetd</code> directory exists there. If the <code>vnetd</code> directory exists in both locations, logging occurs only in <code>/usr/opensv/netbackup/logs/vnetd</code>.</p>

More information is available on the programs and daemons that write the logs.

See [“Multiplexed backup process”](#) on page 67.

On UNIX systems, also refer to the `README` file in the `/usr/opensv/netbackup/logs` directory.

Directory names for legacy debug logs for media and device management

The debug log directories enable logging for the media management processes and device management processes. [Table 1-13](#) describes the directories you need to create to support legacy debug logs for media and device management. Each directory corresponds to a process.

Table 1-13 Media and device management legacy debug logs

Directory	Associated process
<code>acsssi</code>	UNIX only. Debug information on transactions between NetBackup and the StorageTek ACSLS server.
<code>daemon</code>	Debug information for <code>vmd</code> (NetBackup Volume Manager service, Windows) and its associated processes (<code>opr</code> and <code>rdevmi</code>). Stop and restart <code>vmd</code> after creating the directory.
<code>ltid</code>	Debug information on <code>ltid</code> , the Media Manager device daemon (UNIX), or on the NetBackup Device Manager service (Windows), and on <code>avrd</code> . Stop and restart <code>ltid</code> after creating the directory.
<code>reqlib</code>	Debug information on the processes that request media management services from <code>vmd</code> or EMM. Stop and restart <code>vmd</code> after creating the directory.
<code>robots</code>	Debug information on all robotic daemons, which includes <code>tl8cd</code> , <code>tl8cd</code> , and <code>tl4d</code> daemons. Stop and restart robotic daemons.

Table 1-13 Media and device management legacy debug logs (*continued*)

Directory	Associated process
tpcommand	Debug information for device configuration, including the <code>tpconfig</code> and the <code>tpautoconf</code> commands and the NetBackup Administration Console .
vmcsd	Debug information for the NetBackup Status Collection daemon. Stop and restart <code>vmcsd</code> after creating the directory.

Unless it is noted, each directory should be created under the following directory.

Windows `install_path\Volmgr\debug`

UNIX `/usr/opensv/volmgr/debug`

NetBackup creates 1 log per day in each of the debug directories.

You can disable debug logging by deleting or renaming the following directory:

Windows: NetBackup Volume Manager service `install_path\Volmgr\debug\daemon`

UNIX: vmd command `/usr/opensv/volmgr/debug/daemon`

See [“File name format for legacy logging”](#) on page 42.

See [“About limiting the size and the retention of legacy logs”](#) on page 47.

See [“Directory names for legacy debug logs for media and device management”](#) on page 45.

How to control the amount of information written to legacy logging files

You can set legacy logging levels to increase the amount of information that NetBackup processes write in the logs.

The following settings affect legacy logging, except media and device management.

- Increase the **Global logging level**.
See [“Changing the logging level”](#) on page 52.

Note: This setting also affects unified logging.

- On UNIX, add a `VERBOSE` entry in the `/usr/opensv/netbackup/bp.conf` file. If you enter `VERBOSE` without a value, the verbose value defaults to 1. For more log detail, enter `VERBOSE = 2` or a higher value. This setting affects legacy logging only.

Warning: High verbose values can cause debug logs to become very large.

- Set the logging level for individual processes.
In **Host Properties**, change logging levels for individual processes in the **Logging** dialog box. Or, specify the verbose flag (if available) when you start the program or daemon.
Also, you can set the logging level of an individual process to a negative value in the `bp.conf` file as follows:
`<processname>_VERBOSE = -2` completely disables logs for the corresponding process.
See more about logging properties in the [NetBackup Administrator's Guide, Volume I](#).

Media and device management legacy logging has two levels: not verbose (the default) and verbose. To set the verbose (higher) level, add the word `VERBOSE` to the `vm.conf` file. Create the file if necessary. Restart `ltid` and `vmd` after you add the `VERBOSE` entry. This entry affects logging levels in the **Event Viewer** Application and System log. The `vm.conf` file is located in the following directory:

Windows	<code>install_path\Volmgr\</code>
UNIX	<code>/usr/opensv/volmgr/</code>

About limiting the size and the retention of legacy logs

Certain NetBackup processes write legacy debug logs. Because legacy debug logs can grow very large, enable them only if unexplained problems exist. Delete the logs and the associated directories when they are no longer needed.

See [“About log retention in NetBackup”](#) on page 10.

To limit the time for which NetBackup retains logs, specify the number of days in the **Keep logs for days** field. The default is 28 days. You can specify the number under **Host Properties** in the **Logging** dialog box.

Note: The following properties have been moved from the **Clean-up** host properties to the **Logging** host properties: **Keep logs** and **Keep Vault logs**. On the **Logging** properties screen, these properties are referred to as **Keep logs for days** and **Keep Vault logs for** respectively.

See the [NetBackup Administrator's Guide, Volume I](#) for more information about logging properties.

To limit the amount of disk space that the logs consume, use robust logging. Robust logging involves file rotation, like that which is used in unified logging. Robust logging does not apply to media and device management logging.

See [“About rolling over unified log files”](#) on page 24.

Specify the maximum size for a log file and the maximum number of log files to keep in a logging directory. When a log file grows to its maximum size, it closes and a new file opens. If the number of log files exceeds the number that is allowed for the directory, the oldest file is deleted.

Logs created by the following NetBackup processes can use log rotation (robust logging):

- `bpbkar` (UNIX/Linux client only)
- `bpbarm`
- `bpcd`
- `bpdbrm`
- `bpdm`
- `bprd`
- `bptm`
- `nbproxy`

For the logs that are created by the other NetBackup processes (except media and device management logs), use the **Keep logs for days** property. The **Keep logs for days** property may override the robust file logging settings. If **Keep logs for days** is set to 10 days and robust file logging settings allow more than 10 days, the logs are deleted on day 11.

For media and device management legacy logs, use the `DAYS_TO_KEEP_LOGS` setting in the `vm.conf` file to control log file rotation. The default is 30 days. The `vm.conf` file is located in the following directory:

Windows `install_path\Volmgr\`


```
UNIX /usr/opensv/volmgr/
```

To retain logs for 3 days, enter the following in the `vm.conf` file:

```
DAYS_TO_KEEP_LOGS = 3
```

See the [NetBackup Administrator's Guide, Volume II](#) for instructions about how to use this entry.

Configuring the legacy log rotation

You can specify the maximum file size for a legacy log and the maximum number of log files to retain.

See [“About log retention in NetBackup”](#) on page 10.

In the case of legacy logging, NetBackup uses the `bp.conf` configuration file to set the maximum size of a log file. Use the `bpsetconfig` command to configure the following `bp.conf` parameters to do the log settings: `MAX_LOGFILE_SIZE` and

`MAX_NUM_LOGFILES`

Initially, the `bp.conf` file does not contain the `MAX_LOGFILE_SIZE` and `MAX_NUM_LOGFILES` entries. In this case, the parameters are set to their default values, which are 256 MB and infinite, respectively.

Note: Beginning in NetBackup 7.7, the robust logging option is enabled by default.

To configure the legacy log rotation

- ◆ To change the maximum file size or the maximum number of log files per directory, use the `MAX_LOGFILE_SIZE` and the `MAX_NUM_LOGFILES` options. These options are part of the `bpsetconfig` command that is located in the following directory:

Windows `install_pathNetBackup\bin\admincmd\`

UNIX `/usr/opensv/netbackup/bin/admincmd/`

Use the following UNIX example to set the maximum file size to 512 MB and the maximum number of log files per log directory to 4:

```
#bpsetconfig

bpsetconfig> MAX_LOGFILE_SIZE = 512

bpsetconfig> MAX_NUM_LOGFILES = 4

bpsetconfig>

CTRL-D
```

A complete description of `bpsetconfig` is in the [NetBackup Commands Reference Guide](#).

About global logging levels

Global logging levels refer to unified logging and legacy logging. The logging level determines how much information is included in the log message. The higher the level number, the greater the amount of detail is in the log messages.

[Table 1-14](#) describes all of the logging levels and the details that are included in each level.

Table 1-14 Global logging levels

Logging level	Description
Minimum logging	<p>Includes very important, low-volume diagnostic messages and debug messages.</p> <p>The Host Properties Logging page or the Logging Assistant can set minimum logging.</p> <p>Legacy logs use the following values to represent minimum logging:</p> <ul style="list-style-type: none"> ■ Windows: Registry displays the following hexadecimal value: 0xffffffff ■ UNIX: The <code>bp.conf</code> file displays <code>VERBOSE = 0</code> (global). <code>processname_VERBOSE = 0</code> represents using the global default for an individual process. <p>If the global <code>VERBOSE</code> value is set to a value other than 0, an individual process can be decreased by using the value -1. For example, <code>processname_VERBOSE = -1</code>.</p> <p>Unified logging uses the value 1 to represent minimum logging.</p>
Disable logging	<p>The Host Properties Logging page or the Logging Assistant can set disable logging.</p> <p>Legacy logs use the following values to represent disabled logging:</p> <ul style="list-style-type: none"> ■ UNIX: The <code>bp.conf</code> file displays <code>VERBOSE=-2</code> (global) or <code>processname_VERBOSE = -2</code> for an individual process. ■ Windows: Registry displays the following hexadecimal value: 0xfffffffffe <p>Unified logging uses the value 0 to represent disabled logging.</p>
1	Adds verbose diagnostic messages and debug messages to the low-volume diagnostic messages that are associated with minimum logging.
2	Adds the progress messages.
3	Adds the informational dumps.
4	Adds the function entry and exits.
5	Includes everything. The finest detail of messages.

Unified logging is enabled by default to log debug messages at level 0 and application messages at level 5.

The following actions affect logging levels:

- In the **Global logging level** list, a zero (0) level specifies the minimum level of logging for both legacy and unified logging. However, for diagnostic and debug messages in unified logging, the logging level can be turned off completely. No

diagnostic messages or debug messages are logged. This level cannot be set with the **Global logging level** list in the **NetBackup Administration Console**. You can set it with the `vxlogcfg` command or the Logging Assistant.

See [“Changing the logging level”](#) on page 52.

See [“Examples of using vxlogcfg to configure unified logs”](#) on page 34.

- A change to the **Global logging level** list affects the logging level of all NetBackup and Enterprise Media Manager (EMM) processes on the server or client. (The exceptions are PBX and media and device management logging.) This setting overrides any previous settings.
- If you make a change to the VERBOSE entry (or entries) in the `bp.conf` file or entry in the `vm.conf` file, it only affects the legacy logging.
See [“How to control the amount of information written to legacy logging files”](#) on page 46.
- If you make a change with the `vxlogcfg` command, it only affects the unified logging level.

A change to the **Global logging level** list does not affect the level of the following logging processes:

- PBX logging
See the [NetBackup Troubleshooting Guide](#) for more information on how to access the PBX logs.
- Media and device management logging (`vmd`, `ltid`, `avrd`, robotic daemons, media manager commands)
See [“Directory names for legacy debug logs for media and device management”](#) on page 45.
- Any unified logging process whose debug level has been changed from the default setting

Changing the logging level

The logging level determines how much information is included in the log message. The log range is 0-5. The higher the level number, the greater the amount of detail is in the log message.

To change the logging level

- 1 In the **NetBackup Administration Console**, in the left pane, expand **NetBackup Management > Host Properties**.
- 2 Select **Master Servers**, **Media Servers**, or **Clients**.
- 3 In the right pane, click the server or client to view the version and platform. Then, double-click to view the properties.

- 4 In the properties dialog box, in the left pane, click **Logging**.
- 5 In the **Global logging level** list, select a value from 0 to 5.
Changes affect the logging level of both unified logging and legacy logging.
See [“About global logging levels”](#) on page 50.
- 6 Click **OK**.

Changing the logging level on Windows clients

You can increase the amount of information that client processes write in the logs.

To change the logging level on Windows clients

- 1 On the client, open the **Backup, Archive, and Restore** interface.
- 2 Click on the **File** menu and select **NetBackup Client Properties**.
- 3 In the **NetBackup Client Properties** dialog box, select the **Troubleshooting** tab.
- 4 In the **Verbose** property field, enter a debug level from 0 to 5.
Use the default level of 0 unless advised otherwise by Technical Support.
Higher levels can cause the logs to accumulate large amounts of information.
- 5 Click **OK**.

For the unified logging files that the Bare Metal Restore process `bmrsavecfg` creates, you also can control the logging level with the `vxlogcfg` command.

See [“Examples of using vxlogcfg to configure unified logs”](#) on page 34.

An increase in the log level can cause the logs to grow very large; increase the logging level only if unexplained problems exist.

Setting Media Manager debug logging to a higher level

To solve many error conditions, set the debug logging to a higher level. Then retry the operation and examine the debug logs.

To set debug logging to a higher level

- 1 Enable legacy debug logging by creating the necessary directories and folders.
- 2 Increase the level of verbosity for media and device management processes by adding the **VERBOSE** option in the `vm.conf` file. This file is located in `/usr/openv/volmgr/` (UNIX and Linux) or `install_path\Volmgr\` (Windows).
- 3 Restart the daemons and services or run the command `verbose` option, if available.

Setting retention limits for logs on clients

You can specify the numbers of days that NetBackup retains client logs on UNIX and Windows.

To set retention limits for logs on UNIX clients

- 1 In the **NetBackup Administration Console**, in the left pane, expand **Host Properties > Clients**.
- 2 In the right pane, double-click the client you want to modify.
- 3 In the properties dialog box, click **UNIX Client**.
- 4 In the **Client Settings** dialog box, find the **Keep status of user-directed backups, archives, and restores for** field.
- 5 Enter the number of days you want to retain the log files, and click **OK**.

To set the retention limits for logs on Windows clients

- 1 In the **NetBackup Administration Console**, on the **File** menu, click **Backup, Archive, and Restore**.
- 2 In the **Backup, Archive, and Restore** interface, on the **File** menu, click **NetBackup Client Properties**.
- 3 In the **NetBackup Client Properties** dialog box, select the **General** tab.
- 4 In the **Keep status of user-directed backups, archives, and restores for** field, enter the number of days you want to retain the log files.
- 5 Click **OK**.

Logging options with the Windows Event Viewer

The NetBackup Windows master servers can be configured to write messages from NetBackup processes to the Application Event log as well as their normal location. These messages can be reviewed in the Windows **Event Viewer** and also use third-party tools to monitor the Application Event log for these messages.

Two logging options can be used to write messages to the Application Event log. These can be used separately or together and are specific to the type of process that you want to log, as follows:

- To monitor unified processes (process names that start with nb; for example, `nbrb`), use the `vxlogview` command.
- To monitor legacy processes (process names that start with bp; for example, `bpdbm`), configure the `eventlog` file.

Note: For the settings in the `vxlogcfg` command or the `eventlog` file to take effect, you must restart the NetBackup services.

To route unified logging application and diagnostic messages for an originator to the Windows **Event Viewer** Application log, use the `vxlogcfg` command and set the `LogToOslog` value to true for that originator.

The following example routes the application and diagnostic messages for `nbrb` to the Windows **Event Viewer** Application log:

```
# vxlogcfg -a -o nbrb -p NB -s "LogToOslog=true"
```

and the following example message is written in the Windows **Event Viewer** Application log when the operating system logging is enabled for `nbrb`:

```
from nbrb - request ID {1C7FF863-4BCB-46EA-8B35-629A43A4FF1F} failed with status 0
(Not Enough Valid Resources); releasing 2 allocated resources
```

Note: For this setting to take effect, you must restart the NetBackup services.

When you change this option, the ignorable error messages are also written to the Windows **Event Viewer** Application log. For example, if you specify the following command:

```
# vxlogcfg -a -o nbpem -p NB -s "LogToOslog=true"
```

the following example of an ignorable message is written in the Windows **Event Viewer** Application log when a storage lifecycle policy does not exist:

```
call NBProxy::getClientList failed to nbproxy with status 227
```

A complete description of `vxlogcfg` is in the [NetBackup Commands Reference Guide](#).

To use the `eventlog` file, do the following:

- Create the following file on the NetBackup master server.

```
install_path\NetBackup\db\config\eventlog
```

- Optionally, add an entry to the `eventlog` file. The following is an example:

```
56 255
```

Note: For this setting to take effect, you must restart the NetBackup services.

The parameters in the `eventlog` represent severity and type. The parameters have the following characteristics:

- | | |
|----------|---|
| Severity | <ul style="list-style-type: none"> ■ Listed as the first parameter. ■ Controls the messages that NetBackup writes to the Application log. ■ If the file is empty, the default severity is Error (16). ■ If the file has only one parameter, it is used for the severity level. |
| Type | <ul style="list-style-type: none"> ■ Listed as the second parameter. ■ Controls the type of messages that NetBackup writes to the Application log. ■ If the file is empty, the default type is Backup Status (64). |

Both parameters are specified as decimal numbers and equate to a bitmap that expresses the following values:

- | | |
|----------|--|
| Severity | <p>1 = Unknown</p> <p>2 = Debug</p> <p>4 = Info</p> <p>8 = Warning</p> <p>16 = Error</p> <p>32 = Critical</p> |
| Type | <p>1 = Unknown</p> <p>2 = General</p> <p>4 = Backup</p> <p>8 = Archive</p> <p>16 = Retrieve</p> <p>32 = Security</p> <p>64 = Backup Status</p> <p>128 = Media Device</p> |

You can configure the `eventlog` file to log the messages that include several different severities and types. If you specify an entry of **56 255** in the `eventlog` file, the results are as follows:

- | | |
|----------|---|
| Entry 56 | Produces a log with the messages that have a severity of warning, error, and critical. ($56 = 8 + 16 + 32$) |
|----------|---|

Entry 255 Produces a log with messages for all types. ($255 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128$)

The following example message is written in the Windows **Event Viewer** Application log:

```
16 4 10797 1 cacao bush nbpem backup of client bush exited with status 71
```

The definition of each value is as follows (left to right):

- Severity = 16 (Error)
- Type = 4 (Backup)
- Job ID = 10797
- Job group ID = 1
- Server = cacao
- Client = bush
- Process = nbpem
- Text = backup of client bush exited with status 71

Troubleshooting error messages in the NetBackup Administration Console

Most error messages in the **NetBackup Administration Console** appear in the following locations:

- An attention dialog box
- An error message pane in the lower right area of the console

If the errors appear elsewhere, they are Java exception errors. They can appear in the status line (bottom) of the **NetBackup Administration Console** window. They also can appear in the log file that contains the `stdout` or the `stderr` messages written by the Java APIs or the **NetBackup Administration Console**. Veritas does not document Java exception errors.

Four types of error messages appear in the **NetBackup Administration Console**.

Table 1-15 Error message types

Error type	Description
NetBackup status codes and messages	<p>The operations that are performed in the NetBackup Administration Console can result in the errors that are recognized in other parts of NetBackup. These errors usually appear exactly as documented in the NetBackup status codes and messages.</p> <p>Note: A status code does not always accompany the error message.</p> <p>To find the status code, look up the NetBackup message in the alphabetical listing and click the link to see a full description.</p> <p>See the Status Codes Reference Guide.</p>
NetBackup Administration Console: application server status codes and messages	<p>These messages have status codes in the 500 range. Messages with status codes 500, 501, 502, 503 and 504 begin with "Unable to login, status:". Messages with status codes 511 and 512 may or may not begin with "Unable to login, status:".</p> <p>Note: A status code does not always accompany the error message.</p> <p>See the Status Codes Reference Guide.</p>
Java exceptions	<p>Either the Java APIs or NetBackup Administration APIs generate these exceptions. These messages begin with the name of the exception. For example:</p> <pre>java.lang.ClassCastException</pre> <p>or</p> <pre>vrts.nbu.NBUCommandExecutionException</pre> <p>Java exceptions usually appear in one of the following places:</p> <ul style="list-style-type: none"> ■ The status line (bottom) of the NetBackup Administration window ■ The log file that the jnbSA or jbpSA commands generate ■ The output file of the Windows Display Console .bat file if it is set up <p>See "Troubleshooting error messages in the NetBackup Administration Console" on page 57.</p>
Operating system errors	Any messages that do not match those in the NetBackup documentation are most likely messages from the operating system.

About extra disk space required for logs and temporary files

For successful operation, the **NetBackup Administration Console** requires extra disk space to store logs and temporary files. The disk space should be available in the following locations.

- On the host that is specified in the logon dialog box
- In `/usr/opensv/netbackup/logs/user_ops`
- On the host where the console was started
- In `/usr/opensv/netbackup/logs/user_ops/nbjlogs`

If space is not available in the respective file systems, you can experience the following:

- Long waits for application response
- Incomplete data
- No response during logon
- Reduced functionality in the NetBackup interface, for example, only the Backup, Archive, and Restore and Files System Analyzer nodes appear in the tree
- Unexpected error messages:
 - "Cannot connect" socket errors during logon to the NBJava application server
 - "Unable to log in, status: 35 cannot make required directory"
 - `"/bin/sh: null: not found (1) "`
 - "An exception occurred: vrts.nbu.admin.bpmgmt.CommandOutputException: Invalid or unexpected class configuration data: *<the rest of the message will vary>*"
- Empty warning dialog boxes

Enabling detailed debug logging

The **NetBackup Administration Console** is a distributed application that allows the administration of remote NetBackup servers. All administration is accomplished through the application server of the **NetBackup Administration Console**. This application server is made up of an authentication service and a user service.

The logon request from the logon dialog box is sent to the authentication service for validation. The user name and password have to be valid in the Windows/UNIX authentication files and process.

After validation, the authentication service starts a user service under the user's account. Thereafter, all NetBackup administrative tasks are performed through an

instance of the user service. Additional user service processes are initiated to process requests from the console.

On both UNIX and Windows, the authentication service is the `bpjava-msvc` application. The user service is the `bpjava-susvc` or `bpjava-usvc` application. To enable detailed debug logging, you must first create logging directories for these applications.

Table 1-16 Enabling detailed debug logging

Step	Action	Description
Step 1	Create logging directories	<p>On the NetBackup client or server that is specified in the logon dialog box, create the following directories:</p> <ul style="list-style-type: none"> ■ <code>bpjava-msvc</code> ■ <code>bpjava-susvc</code> (if a NetBackup server) ■ <code>bpjava-usvc</code> (if a NetBackup client) <p>Create the directories in the following locations:</p> <ul style="list-style-type: none"> ■ <code>install_path\NetBackup\logs</code> (Windows) ■ <code>/usr/openv/netbackup/logs</code> (UNIX) <p>See “About unified logging” on page 13.</p> <p>See “About legacy logging” on page 36.</p>
Step 2	Edit the <code>Debug.properties</code> file	<p>Add the following line to the <code>Debug.properties</code> file:</p> <pre>debugMask=0x00040000</pre> <p>The <code>Debug.properties</code> file can be found in the following locations:</p> <ul style="list-style-type: none"> ■ <code>/usr/openv/java</code> Change the file on the UNIX machine where you run the <code>jnbSA</code> or <code>jbpSA</code> commands. The log file name is displayed in the xterm window where you ran the <code>jnbSA</code> or <code>jbpSA</code> commands. ■ <code>install_path\VERITAS\java</code> Change the file at this location if you use the NetBackup Windows Display Console.
Step 3	Edit the <code>nbjava.bat</code> file	<p>Perform this step if you use the Windows Display Console on a host where NetBackup is not installed.</p> <p>Edit the <code>nbjava.bat</code> file to redirect output to a file.</p> <p>The <code>nbjava.bat</code> file is located in <code>install_path\VERITAS\java</code> See the <code>nbjava.bat</code> file for details.</p>

This detailed debug logging provides more information than the **NetBackup Administration Console** logging that you can configure within the **Administration Console**. See the *NetBackup Administrator's Guide, Volume I* at the following URL:

<http://www.veritas.com/docs/DOC5332>

For information on how to create logs when you start the Java-based **NetBackup Administration Console** from a Windows computer in NetBackup:

See “[About the Java-based administration console logging](#)” on page 163.

Backup process and logging

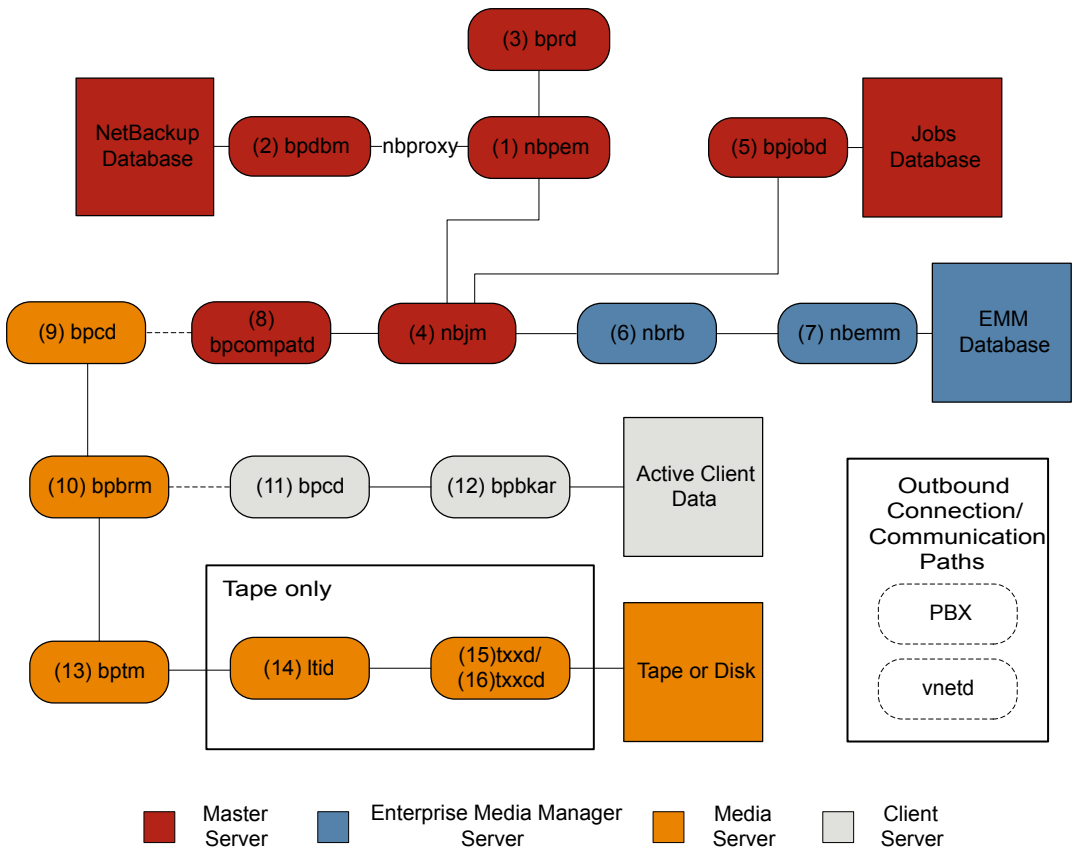
This chapter includes the following topics:

- [Backup process](#)
- [NetBackup process descriptions](#)
- [About backup logging](#)
- [Sending backup logs to Veritas Technical Support](#)

Backup process

Understanding how the backup process works is a helpful first step in deciding which processes to review for troubleshooting purposes.

[Figure 2-1](#) illustrates the backup procedure and the process flow during a scheduled backup.

Figure 2-1 Basic backup process flow**Basic backup procedure**

- 1 The (1) NetBackup Policy Execution Manager (*nbpem*) initiates a backup when the job becomes due. To determine when the job is due, *nbpem* uses the proxy service *nbproxy* to get the backup policy information from the (2) NetBackup Database Manager (*bpdbm*).

In the case of a user-initiated backup, the backup is started when *nbpem* receives a request from the (3) NetBackup request daemon (*bprd*).
- 2 When the job is due, *nbpem* issues a request to the (4) NetBackup Job Manager (*nbjm*) to submit the backup and get a *jobid*.

- 3 The `nbjm` service communicates with (5) `bpjobd`, and the job is added to the job list in the jobs database. The job is now visible in the Activity Monitor, in a queued state.
- 4 Once the job has been added to the jobs database, `nbjm` checks for resources through the (6) NetBackup Resource Broker (`nbrb`).
- 5 The `nbrb` process secures the required resources from the (7) Enterprise Media Manager (`nbemm`) and notifies `nbjm` that resources have been allocated.
- 6 After resource allocation, `nbjm` makes a call to the images database to create the image files in a temporary location. The required entries in the backup header tables are also created at this time. The job is now seen as “Active” in the Activity Monitor.
- 7 Once the job is active, `nbjm` uses (8) `bpcompatd` to open a connection to the (9) client service (`bpcd`) on the media server. The `bpcompatd` service creates the connection through Private Branch Exchange (PBX) and the NetBackup Legacy Network Service (`vnetd`).
- 8 The `bpcd` service starts the (10) NetBackup backup and restore manager (`bpbrm`).
- 9 The `bpbrm` service communicates with (11) `bpcd` on the client server (through PBX and `vnetd`) to start the (12) backup and archive manager (`bpbkar`). The `bpbrm` service also starts the (13) tape management process (`bptm`).
- 10 In the case of a tape backup, `bptm` reserves the drives and issues a mount request to the (14) logical tape interface daemon (`ltid`). The `ltid` service calls on the (15) robotic drive daemon (`txxd`, where `xx` varies based on the type of robot being used). The `txxd` daemon communicates the mount request to the (16) robotic control daemon (`txxcd`), which mounts the media.

In the case of a disk backup, `bptm` communicates directly with the disk.
- 11 The `bpbkar` service sends the backup data through `bptm` to be written to the media storage or the disk storage.
- 12 When the backup is completed, `nbjm` is notified and sends a message to `bpjobd`. The job now appears as “Done” in the Activity Monitor. The `nbjm` service also reports the job exit status to `nbpem`, which recalculates the next due time of the job.

Each of the processes that is involved in a backup has an accompanying log file. These logs can be consulted to diagnose any issues that you encounter with your backups.

Some additional logs that are not included in the backup process flow but that can be of use in resolving backup problems include: `bpbbackup`, `reqlib`, `daemon`, `robots`, and `acsssi`.

NetBackup process descriptions

The following topics provide a functional overview of NetBackup backup and restore operations for both UNIX and Windows. The discussions include descriptions of important services or daemons and programs, and the sequence in which they execute during backup and restore operations. The databases and the directory structure of the installed software are also described.

See [“Backup and restore startup process”](#) on page 65.

See [“Backup and archive processes”](#) on page 66.

See [“Backups and archives - UNIX clients”](#) on page 67.

See [“Multiplexed backup process”](#) on page 67.

Backup and restore startup process

When the NetBackup master server starts up, a script automatically starts all of the services, daemons, and programs that NetBackup requires. (The startup commands that the script uses vary according to the platform.)

The same is true on a media server. NetBackup automatically starts additional programs as required, including robotic daemons.

For more information about SAN client and Fibre Transport startup processes, see the [NetBackup SAN Client and Fibre Transport Guide](#).

Note: No daemons or programs need to be explicitly started. The necessary programs are started automatically during the backup or restore operation.

A daemon that executes on all servers and clients is the NetBackup client daemon, `bpcd`. On UNIX clients, `inetd` starts `bpcd` automatically so no special actions are required. On Windows clients, `bpinetd` performs the same functions as `inetd`.

Note: All NetBackup processes on UNIX can be started manually by running the following: `/usr/openv/netbackup/bin/bp.start_all`

Backup and archive processes

The backup processes and archive processes vary depending on the type of client. The following explains the various NetBackup processes involved in backups and restores including snapshot, SAN client, synthetic backup, and NetBackup catalog backup.

The job scheduler processes consist of the following:

- The `nbpem` service (Policy Execution Manager) creates policy-client tasks and determines when jobs are due to run. It starts the job and upon job completion, determines when the next job should run for the policy-client combination.
- The `nbjm` service (Job Manager) does the following:
 - Accepts requests from `nbpem` to run backup jobs or media jobs from commands such as `bplabel` and `tpreq`
 - Requests the resources for each job, such as storage units, drives, media, and client and policy resources.
 - Executes the job and starts the media server processes.
 - Fields updates from the media server `bpbrm` process and routes them to the jobs database and the images database.
 - Receives the preprocessing requests from `nbpem` and initiates `bpmount` on the client.
- The `nbrb` service (Resource Broker) does the following:
 - Allocates the resources in response to requests from `nbjm`.
 - Acquires the physical resources from the Enterprise Media Manager service (`nbemm`).
 - Manages the logical resources such as multiplex groups, maximum jobs per client, and maximum jobs per policy.
 - Initiates the drive unloads and manages pending request queues.
 - Queries the media servers periodically for current drive state.

The NetBackup master server and the Enterprise media manager (EMM) server must reside on the same physical host.

The master server is responsible for running jobs as configured in NetBackup policies by using the services `nbpem` and `nbjm`.

The EMM services allocate resources for the master server. The EMM services are the repository for all device configuration information. The EMM services include `nbemm` and its subcomponents along with the `nbrb` service for device and resource allocation.

Backups and archives - UNIX clients

For UNIX clients, NetBackup supports scheduled, immediate manual, and user-directed backups of both files and raw partitions. User-directed archives of files are also supported; raw partition archives are not supported. When the operations start, they are all similar to the extent that the same daemons and programs execute on the server.

Each type of backup is started differently as follows:

- Scheduled backups begin when the `nbpem` service detects that a job is due. It checks the policy configurations for the scheduled client backups that are due.
- Immediate manual backups begin if the administrator chooses this option in the NetBackup Administration Console or runs the `bpbackup -i` command. This action causes `bprd` to contact `nbpem`, which then processes the policy, client, and schedule that the administrator selects.
- User-directed backups or archives begin when a user on a client starts a backup or archive through the user interface on the client. The user can also enter the `bpbackup` or `bparchive` command on the command line. This action invokes the client's `bpbackup` or `bparchive` program, which sends a request to the request daemon `bprd` on the master server. When `bprd` receives the user request it contacts `nbpem`, which checks the policy configurations for schedules. By default `nbpem` chooses the first user-directed schedule that it finds in a policy that includes the requesting client.

For user-directed backups or archives, it is also possible to specify a policy and schedule. A description is available of the UNIX `BPBACKUP_POLICY` and `BPBACKUP_SCHED` options in `bp.conf` and the Windows equivalents.

For more information, see the [NetBackup Administrator's Guide, Volume I](#).

Multiplexed backup process

The process for a multiplexed backup is essentially the same as a non-multiplexed backup. An exception is that a separate `bpbrm` process and `bptm` process is created for each backup image being multiplexed onto the media. NetBackup also allocates a separate set of shared memory blocks for each image. The other client and server processes for multiplexed backups are the same.

About backup logging

A variety of logs exist to help diagnose any issues that occur with backups.

The following common log files are used to review the media and master server failures:

See [“nbpem logging”](#) on page 156.

See [“nbproxy logging”](#) on page 156.

See [“bpdbm logging”](#) on page 152.

See [“bprd logging”](#) on page 153.

See [“nbjm logging”](#) on page 155.

See [“bpjobd logging”](#) on page 152.

See [“nrb logging”](#) on page 157.

See [“nbemm logging”](#) on page 155.

See [“bpcompatd logging”](#) on page 151.

See [“PBX logging”](#) on page 159.

See [“vnetd logging”](#) on page 162.

See [“bpcd logging”](#) on page 151.

See [“bpbrm logging”](#) on page 150.

See [“bpbkar logging”](#) on page 150.

See [“bptm logging”](#) on page 153.

See [“ltid logging”](#) on page 154.

See [“txxd and txxcd logging”](#) on page 161.

The following logs are not included in the backup process flow, but they can be helpful to resolve backup problems:

- `acsssi`
- `bpbbackup`
- `daemon`
- `reqlib`
- `robots`

See [“acsssi logging”](#) on page 149.

See [“bpbbackup logging”](#) on page 149.

See [“daemon logging”](#) on page 154.

See [“reqlib logging”](#) on page 160.

See [“robots logging”](#) on page 160.

If you need assistance, send the logs to Veritas Technical Support.

See [“Sending backup logs to Veritas Technical Support”](#) on page 69.

Sending backup logs to Veritas Technical Support

If you encounter a problem with a backup, you can send a problem report and the relevant logs to Veritas Technical Support for assistance.

See [“Logs to accompany problem reports for synthetic backups”](#) on page 110.

[Table 2-1](#) provides a list of logs and the recommended logging levels that Veritas Technical Support may need to diagnose certain backup issues.

Note: Veritas recommends that the diagnostic level for unified logging be set at the default level of 6.

See [“About global logging levels”](#) on page 50.

Table 2-1 Logs to gather for specific backup issues

Type of problem	Logs to gather
Problems with backup scheduling	<ul style="list-style-type: none"> ■ The <code>nbpem</code> log at debug level 5 ■ The <code>nbjm</code> log at debug level 5 ■ The <code>nbproxy</code> log at verbose 4 ■ The <code>bpdbm</code> log at verbose 2 ■ The <code>bprd</code> log at verbose 5 <p>Note: The <code>bprd</code> log is only needed for problems with manual or user-initiated backups.</p>
Problems with the queued backup jobs that do not go active	<ul style="list-style-type: none"> ■ The <code>nbpem</code> log at debug level 3 ■ The <code>nbjm</code> log at debug level 5 ■ The <code>nbrb</code> log at debug level 4 ■ The <code>nbproxy</code> log at verbose 4 ■ The <code>bpdbm</code> log at verbose 2 ■ The <code>nbemm</code> logs at the default levels ■ The <code>mds</code> log at debug level 2 <p>Note: The <code>mds</code> log writes to the <code>nbemm</code> log.</p>

Table 2-1 Logs to gather for specific backup issues (*continued*)

Type of problem	Logs to gather
Problems with the active backup jobs that do not write	<ul style="list-style-type: none"> ■ The <code>nbgm</code> log at debug level 5 ■ The <code>nbrb</code> log at debug level 4 ■ The <code>bpdgm</code> log at verbose 2 ■ The <code>bpbrm</code> log at verbose 5 ■ The <code>bptm</code> log at verbose 5 ■ The <code>bpcd</code> log at verbose 5 <p>If the problem is a tape load or unload issue, Support may also need the following logs:</p> <ul style="list-style-type: none"> ■ The <code>ltid</code> log ■ The <code>reqlib</code> log ■ The <code>daemon</code> log ■ The <code>robots</code> log ■ The <code>acsssi</code> log (UNIX only)

See [“Setting Media Manager debug logging to a higher level”](#) on page 53.

See [“About backup logging”](#) on page 67.

Media and device processes and logging

This chapter includes the following topics:

- [Media and device management startup process](#)
- [Media and device management process](#)
- [Shared Storage Option management process](#)
- [Barcode operations](#)
- [Media and device management components](#)

Media and device management startup process

Media and device management processes are automatically initiated during NetBackup startup. To start these processes manually, run `bp.start_all` (UNIX) or `bpup` (Windows). The `ltid` command automatically starts other daemons and programs as necessary. The daemons should be running after initial startup.

See [Figure 3-1](#) on page 73.

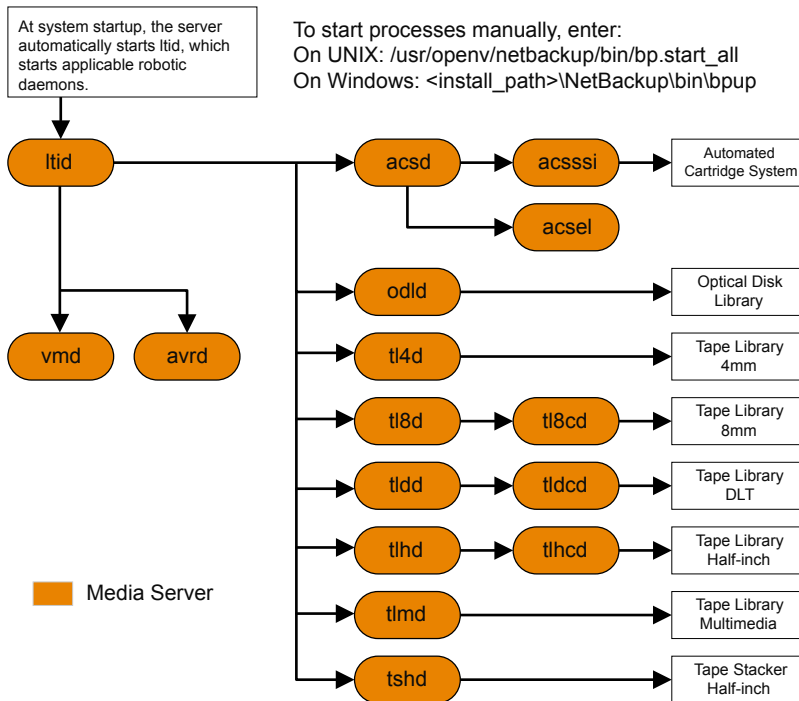
In the case of robotic daemons, such as `tl8d` and `tlhd`, the associated robot must also be configured for the daemon to run. Additional ways to start and stop daemons are available.

See [Table 3-1](#) on page 80.

TL8, TLH, and TLD require following types of daemons:

robotic	Each host with a robotic drive attached must have a robotic daemon. These daemons provide the interface between <code>ltid</code> and the robot. If different drives within a robot can attach to different hosts, the robotic daemon communicates with a robotic-control daemon (see Figure 3-1).
robotic control	Robotic-control daemons centralize the control of robots when drives within a robot can connect to different hosts. A robotic-control daemon receives mount and unmount requests from the robotic daemon on the host to which the drive is attached. It then communicates these requests to the robot.

You must know the hosts that are involved to start all the daemons for a robot.

Figure 3-1 Starting media and device management

Media and device management process

When the media management and device management daemons are running, NetBackup or users can request data storage or retrieval. The scheduling services initially handle the request.

See [“Backup and archive processes”](#) on page 66.

The resulting request to mount a device is passed from `nbgm` to `nbrb`, which acquires the physical resources from `nbeem` (the Enterprise Media Manager service).

If the backup requires media in a robot, `ltid` sends a mount request to the robotic daemon that manages the drives in the robot that are configured on the local host. The robotic daemon then mounts the media, and sets a drive busy status in memory that is shared by itself and `ltid`. Drive busy status also appears in the Device Monitor.

See [Figure 3-2](#) on page 75.

Assuming that the media is physically in the robot, the media is mounted and the operation proceeds. If the media is not in the robot, `nbrb` creates a pending request, which appears as a pending request in the Device Monitor. An operator must insert the media in the robot and use the appropriate Device Monitor command to resubmit the request so the mount request occurs.

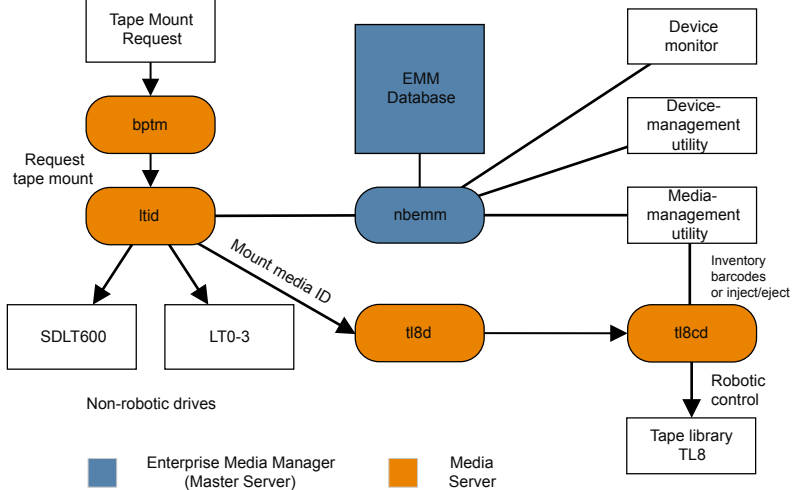
A mount request is issued if the media is for a nonrobotic (standalone) drive that does not contain the media that meets the criteria in the request. If the request is from NetBackup and the drive does contain appropriate media, then that media is automatically assigned and the operation proceeds.

For more information about NetBackup media selection for nonrobotic drives, see the [NetBackup Administrator's Guide, Volume II](#).

Note: When you mount a tape on UNIX, the `drive_mount_notify` script is called. This script is in the `/usr/openv/volmgr/bin` directory. Information on the script can be found within the script itself. A similar script is called for the unmount process (`drive_unmount_notify`, in the same directory).

When a robotic volume is added or removed through the media access port, the media management utility communicates with the appropriate robotic daemon to verify the volume location or barcode. The media management utility (through a library or command-line interface) also calls the robotic daemon for robot inventory operations.

[Figure 3-2](#) shows an example of the media and device management process.

Figure 3-2 Media and device management example process

Shared Storage Option management process

Shared Storage Option (SSO) is an extension to tape drive allocation and configuration for media and device management. SSO allows individual tape drives (standalone or in a robotic library) to be dynamically shared between multiple NetBackup media servers or SAN media servers.

For more information about the Shared Storage Option, see the [NetBackup Administrator's Guide, Volume II](#).

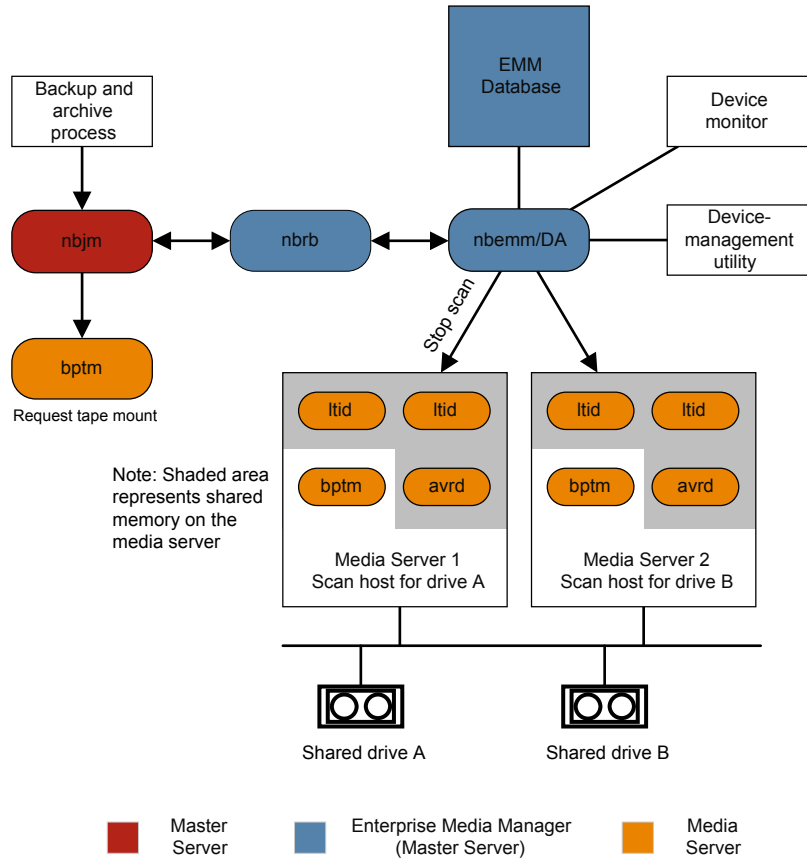
The following shows the Shared Storage Option management process in the order presented:

- NetBackup or users can initiate backups. The `nbjm` process makes a mount request for the backup.
- `nbrb` tells the EMM server to obtain a drive for the backup.
- `nbrb` tells the device allocator (DA) in the EMM server to stop scanning the selected drive.
- `nbemm` tells the appropriate media server (the scan host for the selected drive) to stop scanning the drive. The stop scan request is carried out by means of `oprdr`, `ltid`, and `avrd` in the media server's shared memory.

- `nbemm` informs `nbrb` when the scanning on the selected drive has stopped.
- `nbrb` informs `nbjm` that the selected drive (A) is available for the backup.
- `nbjm` conveys the mount request and drive selection to `bptm`, which proceeds with the backup. To protect the integrity of the write operation, `bptm` uses SCSI reservations.
For more information about how NetBackup reserves drives, see the [NetBackup Administrator's Guide, Volume II](#).
- The mount-media operation is initiated.
- `bptm` makes position checks on the drive to ensure that another application has not rewound the drive. `bptm` also does the actual write to the tape.
- When the backup is complete, `nbjm` tells `nbrb` to release resources.
- `nbrb` de-allocates the drive in EMM.
- EMM tells the scan host to resume scanning the drive. The scan request is carried out by means of `oprdr`, `ltid`, and `avrd` in the media server's shared memory.

[Figure 3-3](#) illustrates the Shared Storage Option management process.

Figure 3-3 Media and device management process flow showing SSO components



Barcode operations

Barcode reading is mainly a function of the robot hardware instead of media and device management. When a robot has a barcode reader, it scans any barcode that is on a tape and stores the code in its internal memory. This associates the slot number and the barcode of the tape in that slot. NetBackup determines that association for its own use by interrogating the robot.

If a robot supports barcodes, NetBackup automatically compares a tape's barcode to what is in the EMM database as an extra measure of verification before you

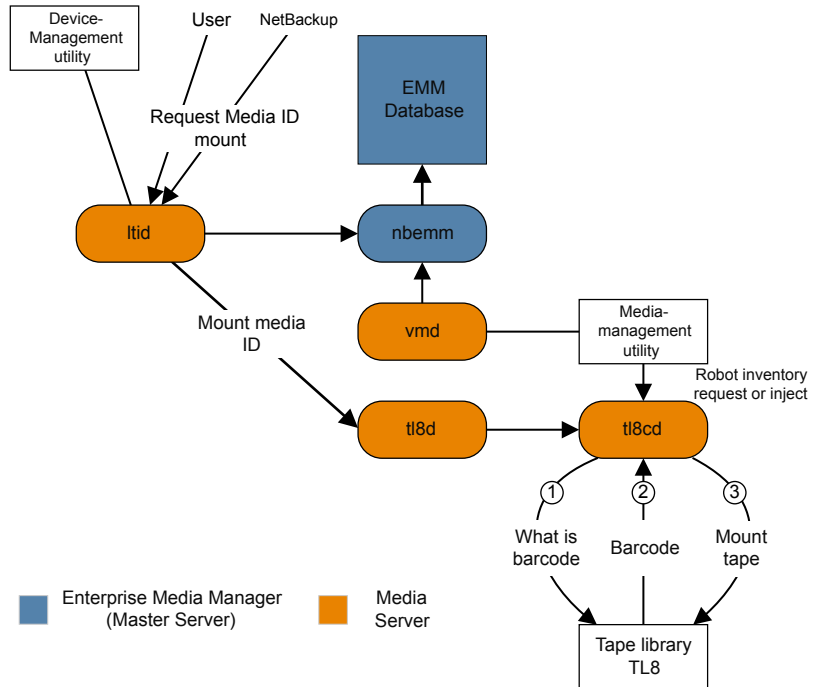
mount the tape. A request for the media that is in a robot that can read barcodes begins in the same manner as other requests.

See [Figure 3-4](#) on page 79.

The `ltid` command includes the media ID and location information in a mount request to the robotic daemon for the robot that has the media ID. This request causes the robotic daemon to query the robotic-control daemon or the robot for the barcode of the tape in the designated slot. (This is a preliminary check to see if the correct media is in the slot.) The robot returns the barcode value it has in memory.

The robotic daemon compares this barcode with the value it received from `ltid` and takes one of the following actions:

- If the barcodes don't match, and the mount request is not for a NetBackup backup job, the robotic daemon informs `ltid` and a pending action request (Misplaced Tape) appears in the Device Monitor. An operator must then insert the correct tape in the slot.
- If the barcodes don't match and the mount request is for a NetBackup backup job, the robotic daemon informs `ltid` and the mount request is canceled. NetBackup (bptm) then requests a new volume from nbjm and from EMM.
- If the barcodes match, the robotic daemon requests the robot to move the tape to a drive. The robot then mounts the tape. At the start of the operation, the application (for example, NetBackup) checks the media ID and if it also matches what should be in this slot, the operation proceeds. For NetBackup, a wrong media ID results in a "media manager found wrong tape in drive" error (NetBackup status code 93).

Figure 3-4 Barcode request

Media and device management components

This topic shows the file and the directory structure and the programs and the daemons that are associated with media management and device management.

Figure 3-5 shows the file structure and directory structure for media management and device management on a UNIX server. A Windows NetBackup server has the equivalent files and the directories that are located in the directory where NetBackup is installed (by default, the `C:\Program Files\VERITAS` directory).

Figure 3-5 Media and device management directories and files

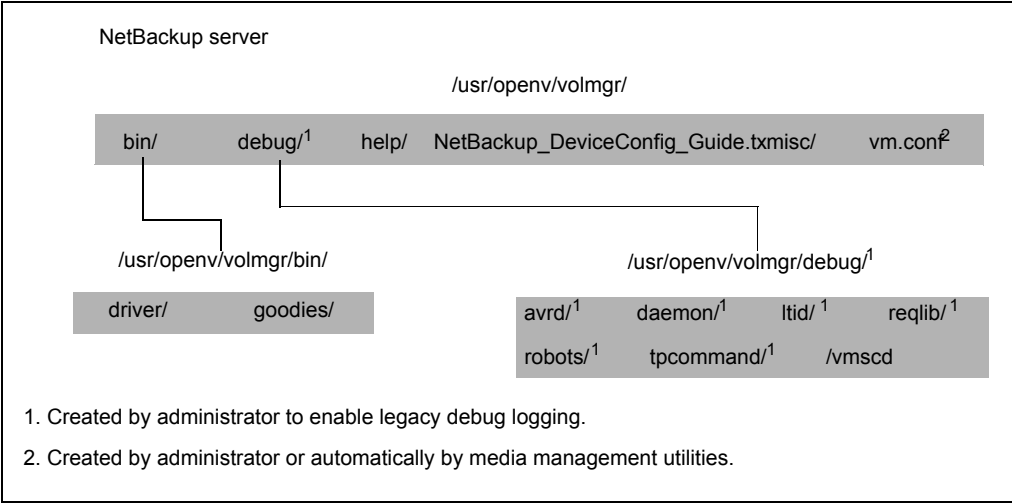


Table 3-1 describes the directories and files that are of special interest.

Table 3-1 Media and device management directories and files

File or directory	Contents
bin	Commands, scripts, programs, daemons, and files that are required for media and device management. The following subdirectories under bin are available: driver: Contains the SCSI drivers that are used on various platforms to control robotics. goodies: Contains the <code>vmconf</code> script and scan utility.
debug	Legacy debug logs for the Volume Manager daemon, <code>vmd</code> , and all requesters of <code>vmd</code> , <code>ltid</code> , and device configuration. The administrator must create these directories for debug logging to occur.
help	Help files that the media and device management programs use. These files are in ASCII format.
misc	Lock files and temporary files that are required by the various components of media and device management.

Table 3-1 Media and device management directories and files (*continued*)

File or directory	Contents
vm.conf	Media and device management configuration options.

[Table 3-2](#) describes the media management and device management programs and daemons. The explanations include what starts and stops the program or daemon, and the log (if any) where it records its activities. On UNIX, all of the components that are discussed in this table are in `/usr/opensv/volmgr/bin`. On Windows, they are in `install_path\volmgr\bin`.

Note: The following table contains references to the system log. On UNIX, `syslog` manages this log (the facility is `daemon`). On Windows, the Event Viewer manages the system log (the log type is `Application`).

Table 3-2 Media and device management daemons and programs

Program or daemon	Description
acsd	<p>The Automated Cartridge System daemon interfaces with the Automated Cartridge System. It communicates with the server that controls the ACS robotics through the <code>acsssi</code> process (UNIX) or the STK Libattach Service (Windows).</p> <p>For UNIX, see the <code>acsssi</code> and the <code>acssel</code> programs.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/acsd</code> command).</p> <p>Stopped By: Stopping <code>ltid</code> (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option: this option can also be used through <code>ltid</code>, or by putting <code>VERBOSE</code> in the <code>vm.conf</code> file.</p>
acssel	<p>Available only on UNIX.</p> <p>See the NetBackup Device Configuration Guide.</p>
acsssi	<p>Available only on UNIX.</p> <p>See the NetBackup Device Configuration Guide.</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
avrd	<p>The automatic-volume-recognition daemon controls the automatic volume assignment and label scanning. This daemon lets NetBackup read labeled tape volumes and automatically assigns the associated removable media to the requesting processes.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/openv/volmgr/bin/avrd</code> command).</p> <p>Stopped By: Stopping <code>ltid</code>, (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: All errors are logged in the system log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by aborting <code>avrd</code> and starting the daemon with the <code>-v</code> option.</p>
ltid	<p>The device daemon (UNIX) or NetBackup Device Manager service (Windows) controls the reservation and assignment of tapes.</p> <p>Started By: <code>/usr/openv/volmgr/bin/ltid</code> command on UNIX or the <code>Stop/Restart Device Manager Service</code> command in the Media and Device Management window on Windows.</p> <p>Stopped By: <code>/usr/openv/volmgr/bin/stopltd</code> command on UNIX or the <code>Stop/Restart Device Manager Service</code> command in the Media and Device Management window on Windows.</p> <p>Debug Log: Errors are logged in the system log and the <code>ltid</code> debug log. Debug information is included if the daemon is started with the <code>-v</code> option (available only on UNIX) or adding <code>VERBOSE</code> to the <code>vm.conf</code> file.</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
tl4d	<p>The Tape Library 4MM daemon is the interface between <code>ltid</code> and the Tape Library 4MM and communicates with the robotics through a SCSI interface.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tl4d</code> command).</p> <p>Stopped By: Stopping <code>ltid</code> (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: All errors are logged in the system log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through <code>ltid</code>).</p>
tl8d	<p>The Tape Library 8MM daemon provides the robotic control for a TL8 robot (Tape Library 8mm or Tape Stacker 8mm). The Tape Library 8MM daemon drives in the same TL8 robot can be attached to different hosts than the robotic control. <code>tl8d</code> is the interface between the local <code>ltid</code> and the robotic control. If a host has a device path for a drive in a TL8 robot, then mount or unmount requests for that drive go first to the local <code>ltid</code> and then to the local <code>tl8d</code> (all on the same host). <code>tl8d</code> then forwards the request to <code>tl8cd</code> on the host that is controls the robot (it can be on another host).</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tl8d</code> command).</p> <p>Stopped By: Stopping <code>ltid</code> (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: Errors are logged in the system log and the robots debug log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through <code>ltid</code>).</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
tl8cd	<p>The Tape Library 8MM control daemon provides the robotic control for a TL8 robot and communicates with the robotics through a SCSI interface. tl8cd receives mount and unmount requests from tl8d on the host to which the drive is attached and then communicates these requests to the robot.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tl8cd</code> command).</p> <p>Stopped By: Stopping <code>ltid</code> or by using the <code>tl8cd -t</code> command.</p> <p>Debug Log: Errors are logged in the system log and the robots debug log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through <code>ltid</code>).</p>
tldd	<p>The Tape Library DLT daemon works with <code>tl8cd</code> to handle requests to TLD robots (Tape Library DLT and Tape Stacker DLT). <code>tldd</code> provides the interface between the local <code>ltid</code> and the robotic control (<code>tl8cd</code>) in the same way as explained previously for <code>tl8d</code>.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tl8cd</code> command).</p> <p>Stopped By: Stopping <code>ltid</code> (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included by adding <code>VERBOSE</code> to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through <code>ltid</code>).</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
tlcdc	<p>The tape library DLT control daemon provides robotic control for a TLD robot in the same way as explained previously for tl8cd.</p> <p>Started By: Starting ltid (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tlcdc</code> command).</p> <p>Stopped By: Stopping ltid or by using the <code>tlcdc -t</code> command.</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through ltid).</p>
tlhd	<p>The Tape Library Half-inch daemon works with tlhcd to handle requests to the TLH robots that are in an IBM Automated Tape Library (ATL). tlhd provides the interface between the local ltid and the robotic control (tlhcd) in the same way as explained previously for tl8d.</p> <p>Started By: Starting ltid (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tlhd</code> command).</p> <p>Stopped By: Stopping ltid (or on UNIX, independently by finding the PID (process ID) and then using the <code>kill</code> command).</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the <code>vm.conf</code> file. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through ltid).</p>
tlhcd	<p>The Tape Library half-inch control daemon provides robotic control for a TLH robot that is in an IBM Automated Tape Library (ATL) in the same way as explained previously for tl8cd.</p> <p>Started By: Starting ltid (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tlhcd</code> command).</p> <p>Stopped By: Stopping ltid or by using the <code>tlhcd -t</code> command.</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included if the daemon is started with the <code>-v</code> option (either by itself or through ltid). The <code>-v</code> option is available only on UNIX. Also, add the VERBOSE option to the <code>vm.conf</code> file.</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
tlmd	<p>The Tape Library Multimedia daemon is the interface between <code>ltid</code> and a TLM robot that is in an ADIC Distributed AML Server (DAS). This daemon communicates with the TLM robotics through a network API interface.</p> <p>Started By: Starting <code>ltid</code> or starting independently by using the <code>/usr/opensv/volmgr/bin/tlmd</code> command.</p> <p>Stopped By: Stopping <code>ltid</code> or stopping independently by finding the PID (process ID) and then using the <code>kill</code> command.</p> <p>Debug Log: Errors are logged in the system log and robots debug log. Debug information is included if the daemon is started with the <code>-v</code> option (either by itself or through <code>ltid</code>). The <code>-v</code> option is available only on UNIX. Also, add the <code>VERBOSE</code> option to the <code>vm.conf</code> file.</p>
tshd	<p>The Tape Stacker Half-inch daemon is the interface between <code>ltid</code> and the half-inch-cartridge stacker and communicates with the robotics through a SCSI interface. This robot is not supported on Windows.</p> <p>Started By: Starting <code>ltid</code> (or on UNIX, independently by using the <code>/usr/opensv/volmgr/bin/tshd</code> command).</p> <p>Started By: <code>tpconfig</code> command.</p> <p>Stopped By: Quit option from within the utility on UNIX. On Windows, <code>tpconfig</code> is only a command-line interface that runs to completion (no quit option).</p> <p>Debug Log: <code>tpcommand</code> debug logs.</p>
vmd	<p>The Volume Manager daemon (NetBackup Volume Manager service on Windows) allows the remote administration and control of Media and Device Management.</p> <p>Started By: Starting <code>ltid</code>.</p> <p>Stopped By: Terminating the Media Manager Volume Daemon option.</p> <p>Debug Log: System log and also a debug log if the daemon or <code>reqlib</code> debug directories exist.</p>

Table 3-2 Media and device management daemons and programs
(continued)

Program or daemon	Description
vmscd	<p>The Media Manager Status Collector Daemon keeps the EMM server database up-to-date with the actual status of the drives that are attached to the 5.x servers.</p> <p>Started By: the EMM server.</p> <p>Stopped By: the EMM server.</p> <p>Debug Log: /usr/openv/volmgr/debug/vmscd (UNIX), install_path\Volmgr\debug\vmscd (Windows)</p>

Restore process and logging

This chapter includes the following topics:

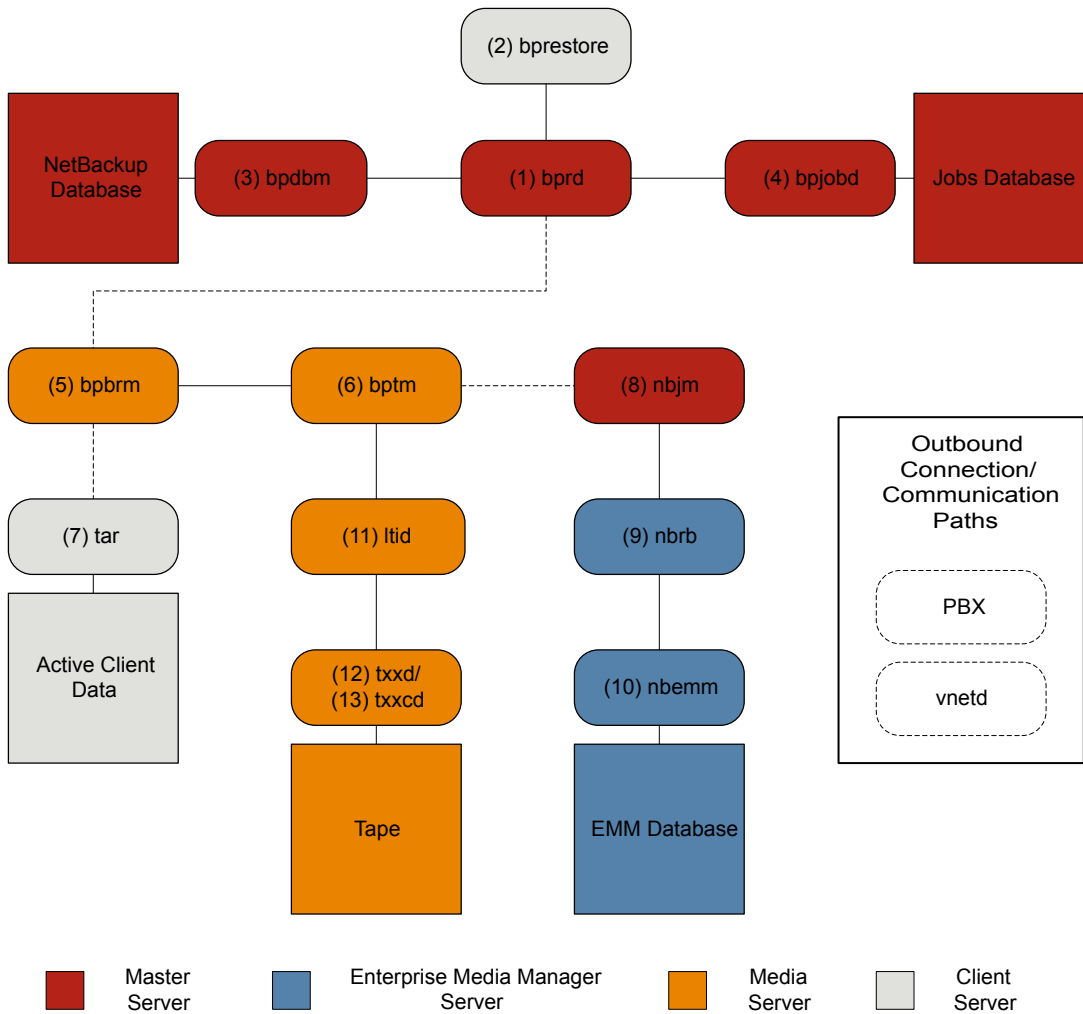
- [Restore process](#)
- [UNIX client restore](#)
- [Windows client restore](#)
- [About restore logging](#)
- [Sending restore logs to Veritas Technical Support](#)

Restore process

Understanding how the restore process works is a helpful first step in deciding which logs to gather for a particular issue. The restore process differs depending on whether you restore an image from tape or from disk.

[Figure 4-1](#) illustrates a restore from tape.

Figure 4-1 Restore from tape process flow



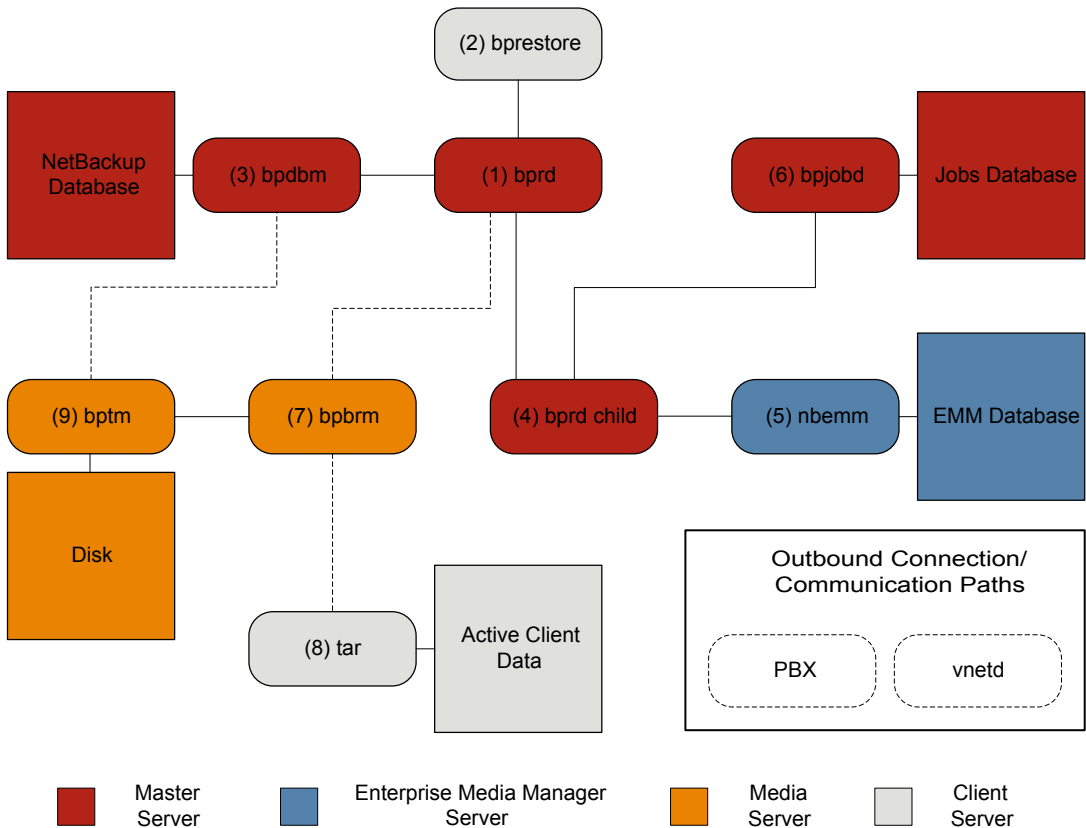
Restore procedure from tape

- 1 The (1) NetBackup request daemon (`bprd`) receives a restore request. This request can be initiated from the Backup, Archive, and Restore user interface or from the (2) command line (`bprestore`).
- 2 The `bprd` process launches two child processes: `MAIN bprd` and `MPX-MAIN-bprd`. The `MAIN bprd` process is used to identify images and media, while the `MPX-MAIN-bprd` process manages the restore operation. For simplicity's sake, these three processes are all referred to here as `bprd`.
- 3 The `bprd` service communicates with the (3) NetBackup Database Manager program (`bpdbm`) to get the information that is required to restore the files that have been requested.
- 4 Once it has the information it needs, `bprd` communicates with (4) `bpjobd`, and the job is added to the job list in the jobs database. The job is now visible in the Activity Monitor. It may show as "Active" even before resources are acquired.
- 5 The `bprd` service goes through Private Branch Exchange (`PBX`) and the NetBackup Legacy Network (`vnetd`) to start the (5) NetBackup backup and restore manager (`bpbrm`).
- 6 The `bpbrm` service starts the (6) tape management process (`bptm`) and provides the media information that is required for the restore. It also starts the (7) Tape Archive program (`tar`) on the client (through `PBX` and `vnetd`) and creates a connection between `tar` and `bptm`.
- 7 The `bptm` process sends a resource request to the (8) NetBackup Job Manager (`nbjm`) through `PBX` and `vnetd`.
- 8 The `nbjm` process sends the resource request to the (9) NetBackup Resource Broker (`nbrb`), which queries the (10) Enterprise Media Manager (`nbemm`). Once the resources have been allocated, `nbrb` notifies `nbjm`, which notifies `bptm`.
- 9 The `bptm` process makes a mount request to the (11) logical tape interface daemon (`ltid`). The `ltid` service calls on the (12) robotic drive daemon (`txxd`, where `xx` varies based on the type of robot being used). The `txxd` daemon communicates the mount request to the (13) robotic control daemon (`txxcd`), which mounts the media.
- 10 The `bptm` process reads the data to be restored from the media and delivers it to `tar`.
- 11 The `tar` process writes the data to the client disk.
- 12 When the restore is completed, `bptm` unmounts the media and notifies `nbjm`. The job now appears as "Done" in the Activity Monitor.

Some additional logs that are not included in the restore process flows but that can be of use in resolving restore problems include: `reqlib`, `daemon`, `robots`, and `acsssi`.

Figure 4-2 illustrates a restore from disk.

Figure 4-2 Restore from disk process flow



Restore procedure from disk

- 1 The (1) NetBackup request daemon (`bprd`) receives a restore request. This request can be initiated from the Backup, Archive, and Restore user interface or from the (2) command line (`bprestore`).
- 2 The `bprd` process contacts the (3) NetBackup Database Manager program (`bpdbm`) to identify the files, the client, and the media information for the restore.

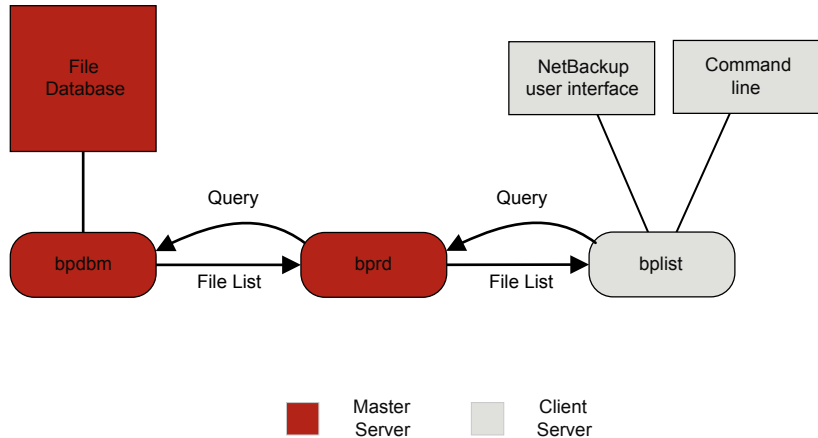
- 3 The `bprd` process initiates a (4) child `bprd` process. The child `bprd` process makes a call to the (5) Enterprise Media Manager (`nbemm`) to verify that the disk storage unit is available.
- 4 The child `bprd` process communicates with (6) `bpjobd` to allocate a `jobid`. The restore job is now visible in the Activity Monitor.
- 5 The `bprd` process starts the (7) NetBackup backup and restore manager (`bpbrm`) on the media server, through Private Branch Exchange (`PBX`) and the NetBackup Legacy Network Service (`vnetd`).
- 6 The `bpbrm` service uses `PBX` and `vnetd` to establish a connection with the (8) Tape Archive program (`tar`) on the client system. It also starts the (9) tape management process (`bptm`).
- 7 The `bptm` process makes a call to `bpdbm` (through `PBX` and `vnetd`) to get the fragment information and then mounts the disk.
- 8 The `bptm` process reads the backup image from the disk and streams the requested data to `tar`.
- 9 The `tar` process commits the data to the storage destination.

Each of the processes that is involved in a restore has an accompanying log file. These logs can be consulted to diagnose any issues that you encounter with your restore.

UNIX client restore

Before you start a restore, use the `bplist` program on the client to do the following: browse the file catalog to list the files available in the backup images, and select the desired files. You can start `bplist` directly from the command line, and the NetBackup user interface programs can use it.

To retrieve the file list, `bplist` sends a query to the request daemon (`bprd`) on the master server (see [Figure 4-3](#)). The request daemon then queries `bpdbm` for the information and transmits it to `bplist` on the client.

Figure 4-3 List operation - UNIX client

The following are the processing steps in a restore (in the order presented):

- When the user starts a restore, NetBackup invokes the client's `bprestore` program which sends a request to the request daemon, `bprd`. This request identifies the files and client. The request daemon then uses `bpcd` (client daemon) to start the backup and restore manager (`bpbm`).

Note: To restore Backup Exec images, `bpbm` initiates `mtfrd` instead of `ntar` on the clients. The server processes are the same as those used for NetBackup restores.

- If the disk device or tape device on which the data resides attaches to the master server, the following occurs: `bprd` starts the backup and restore manager on the master server. If the disk unit or tape unit connects to a media server, `bprd` starts the backup and restore manager on the media server.
- The backup and restore manager starts `bptm` and uses the client daemon (`bpcd`) to establish a connection between NetBackup `ntar` on the client and `bptm` on the server.
- For tape: The `bptm` process identifies which media is needed for the restore, based on the image catalog. `bptm` then requests the allocation of the required media from `nbrb` through `nbjm`. `nbjm` then asks `mds` (part of `nbenm`) for the

resources. `nbemm` allocates the media and selects and allocates an appropriate drive (for tape media).

`bptm` asks `ltid` to mount the tape in the drive.

For disk: `bptm` does not need to ask `nbrb` for an allocation, because disk inherently supports concurrent access. `bptm` uses the file path in a read request to the system disk manager.

- `bptm` directs the image to the client in one of two ways. If the server restores itself (server and client are on the same host), `nbtar` reads the data directly from shared memory. If the server restores a client that resides on a different host, it creates a child `bptm` process which transmits the data to `nbtar` on the client.

Note: Only the part of the image that is required to satisfy the restore request is sent to the client, not necessarily the entire backup image.

- The NetBackup `nbtar` program writes the data on the client disk.

Note: PBX must be running for NetBackup to operate (PBX is not shown in the next diagram). See the *NetBackup Troubleshooting Guide* for more information on how to resolve PBX problems.

Windows client restore

NetBackup supports the same types of operations on Windows clients as it does for UNIX clients.

The following are the Windows processes involved in restore operations:

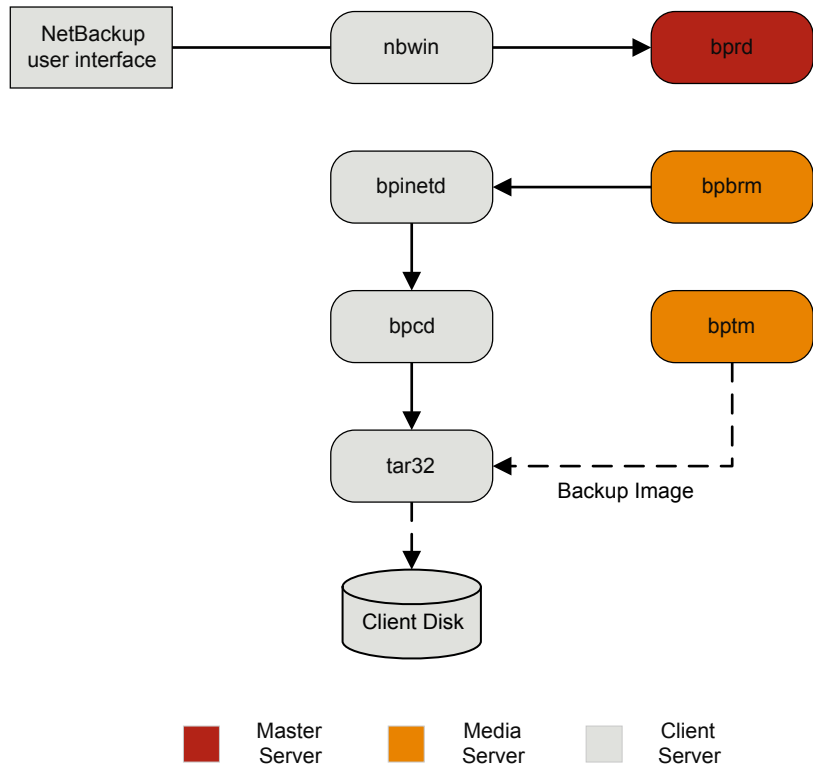
- `NBWIN` is the user interface program on the client. The `bpbackup` function and the `bparchive` function are merged into `NBWIN`.
- `BPINETD` serves the same purpose as `inetd` on UNIX clients.
- The NetBackup client daemon is called `BPCD`.
- `TAR32` is part of NetBackup for Windows and serves the same purpose as NetBackup `nbtar` on UNIX.

Note: To restore Backup Exec images, `bpbrm` invokes `mtfrd.exe` instead of `tar32.exe` on the clients. The server processes are the same as those used for NetBackup restores.

The server processes are the same as described for UNIX.

Figure 4-4 shows the client processes involved in these operations.

Figure 4-4 Restore - Windows client



About restore logging

A variety of logs exist to help diagnose any issues that occur with restores. Understanding how the restore process works is a helpful first step in deciding which logs to gather for a particular issue.

If you need assistance, send the logs to Veritas Technical Support.

See [“Sending restore logs to Veritas Technical Support”](#) on page 96.

The following are the common log files that are used in review of restore failures:

See [“bprd logging”](#) on page 153.

See [“bprestore logging”](#) on page 153.

See [“PBX logging”](#) on page 159.

See [“vnetd logging”](#) on page 162.

See [“bpdbm logging”](#) on page 152.

See [“bpjobd logging”](#) on page 152.

See [“bpbrm logging”](#) on page 150.

See [“bptm logging”](#) on page 153.

See [“tar logging”](#) on page 161.

See [“nbjm logging”](#) on page 155.

See [“nrb logging”](#) on page 157.

See [“nbemm logging”](#) on page 155.

See [“ltid logging”](#) on page 154.

See [“reqlib logging”](#) on page 160.

See [“robots logging”](#) on page 160.

See [“acsssi logging”](#) on page 149.

Sending restore logs to Veritas Technical Support

If you encounter a problem with a restore, you can send a problem report and the relevant logs to Veritas Technical Support for assistance.

See [“Logs to accompany problem reports for synthetic backups”](#) on page 110.

[Table 4-1](#) provides a list of logs and the recommended logging levels that Veritas Technical Support may need to diagnose certain restore issues.

Note: Veritas recommends that the diagnostic level for unified logging be set at the default level of 6.

See [“About global logging levels”](#) on page 50.

Table 4-1 Log to gather for specific restore issues

Type of problem	Log to gather
Problems with restore jobs from tape	<ul style="list-style-type: none"> ■ The <code>nbgm</code> log at debug level 5 ■ The <code>nbemm</code> log at debug level 1 ■ The <code>nbrb</code> log at debug level 4 ■ The <code>bpdbm</code> log at verbose 1 ■ The <code>bprd</code> log at verbose 5 ■ The <code>bpbrm</code> log at verbose 5 ■ The <code>tar</code> log at verbose 5 ■ The <code>bpcd</code> log at verbose 5 <p>If the problem is a media or a drive issue, Support may also need the following logs:</p> <ul style="list-style-type: none"> ■ The <code>reqlib</code> log ■ The <code>daemon</code> log ■ The <code>robots</code> log ■ The <code>acsssi</code> log (UNIX only)
Problems with restore jobs from disk	<ul style="list-style-type: none"> ■ The <code>bpdbm</code> log at verbose 1 ■ The <code>bprd</code> log at verbose 5 ■ The <code>bpbrm</code> log at verbose 5 ■ The <code>bptm</code> log at verbose 5 ■ The <code>bpdm</code> log at verbose 5 ■ The <code>tar</code> log at verbose 5 ■ The <code>bpcd</code> log at verbose 5

See [“Setting Media Manager debug logging to a higher level”](#) on page 53.

See [“About restore logging”](#) on page 95.

Advanced Backup and Restore Features

This chapter includes the following topics:

- [SAN Client Fiber Transport backup](#)
- [SAN Client Fiber Transport restore](#)
- [Hot catalog backup](#)
- [Hot catalog restore](#)
- [Synthetic backups](#)

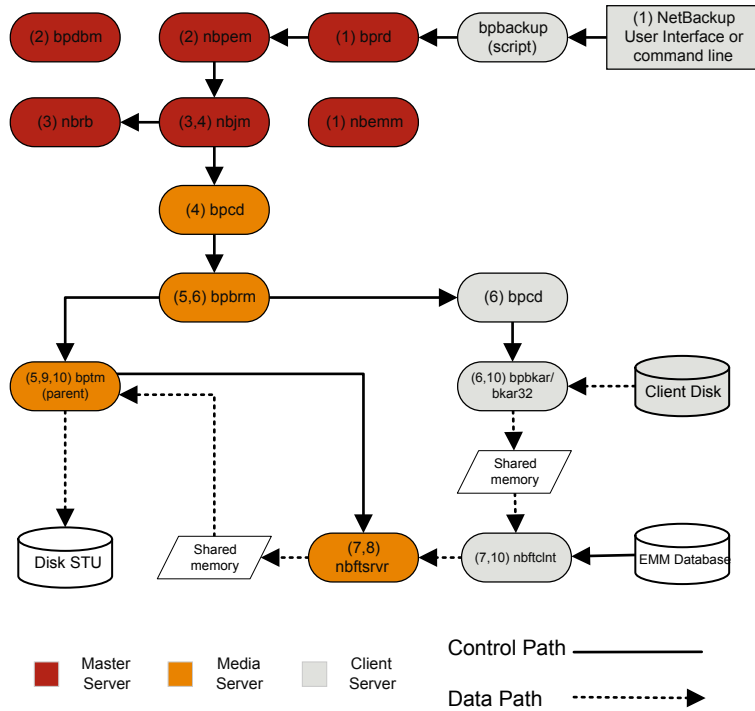
SAN Client Fiber Transport backup

The following shows a SAN client backup process.

For backups to disk, the SAN client feature provides high-speed data movement between NetBackup media servers and NetBackup SAN-attached clients. SAN-attached clients send backup data to the media server by means of Fibre Channel connections.

As part of SAN client, the FT Service Manager (FSM) is a domain layer service that resides on the master server. The FSM provides discovery, configuration, and event monitoring of SAN client resources. The FSM collects Fibre Channel information from the client and from the media server; FSM then populates the NetBackup relational database (NBDB) with the information. FSM runs as a sub-process of NBDB and writes log messages to the NBDB log. FSM interacts with the `nbftclnt` process on NetBackup clients and with the `nbftsrvr` process on media servers.

Figure 5-1 SAN client backup process flow



The processing steps for a SAN client backup operation are the following:

SAN client backup procedure

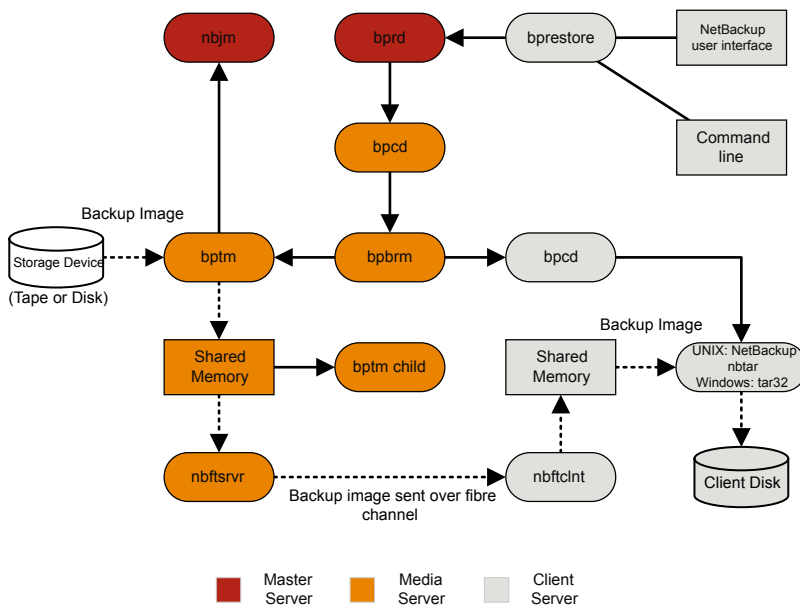
- 1 The NetBackup master server or primary client initiates the backup. The NetBackup request daemon (`bprd`) submits a backup request to the Policy Execution Manager (`nbpem`). `nbpem` processes the policy configurations.
All other daemons and programs are started as necessary including `nbpem`, `nbjm`, `nrb`, and `nbemm`.
- 2 The Policy Execution Manager service (`nbpem`) does the following:
 - Gets the policy list from `bpdbm`.
 - Builds a work list of all scheduled jobs.
 - Computes the due time for each job.

- Sorts the work list in order of due time.
 - Submits to `nbjm` all jobs that are currently due.
 - Sets a wake-up timer for the next due job.
 - When the job finishes, it recomputes the due time of the next job and submits to `nbjm` all of the jobs that are currently due.
- 3 The Job Manager service (`nbjm`) requests backup resources from the Resource Broker (`nbrb`), that returns information on the use of shared memory for the SAN client.
 - 4 The `nbjm` service starts the backup by means of the client daemon `bpcd`, which starts the backup and restore manager `bpbrm`.
 - 5 The `bpbrm` service starts `bptm`, which does the following:
 - Requests the SAN client information from `nbjm`.
 - Sends a backup request to the FT server process (`nbftsvr`).
 - Sends a backup request to the FT client process on the client (`nbftclnt`), that does the following: Opens a Fibre Channel connection to `nbftsvr` on the media server, allocates the shared memory, and writes the shared memory information to the backup ID file.
 - 6 The `bpbrm` service uses `bpcd` to start `bpbkar`, that does the following:
 - Reads the shared memory information from the BID file (waits for the file to exist and become valid).
 - Sends the information about files in the image to `bpbrm`.
 - Writes the file data to `bpbkar`, optionally compresses it, then writes the data to the shared buffer.
 - Sets the buffer flag when the buffer is full or the job is done.
 - 7 The FT client process (`nbftclnt`) waits for the shared memory buffer flag to be set. It then transfers the image data to the FT Server (`nbftsvr`) shared memory buffer, and clears the buffer flag.
 - 8 The `nbftsvr` service waits for data from `nbftclnt`; and writes the data is written to the shared memory buffer. When the transfer completes, `nbftsvr` sets the buffer flag.
 - 9 `bptm` waits for the shared memory buffer flag to be set, writes data from the buffer to the storage device, and clears the buffer flag.
 - 10 At the end of the job:
 - `bpbkar` informs `bpbrm` and `bptm` that the job is complete.

- `bptm` sends `bpbrm` the final status of the data write.
- `bptm` directs `nbftclnt` to close the Fibre Channel connection.
- `nbftclnt` closes the Fibre Channel connection and deletes the BID file.

SAN Client Fiber Transport restore

Figure 5-2 SAN client restore with Fibre Transport



The process flow for a SAN client restore is as follows (in the order presented).

- When the user starts a restore, NetBackup invokes the client's `bprestore` program that sends a request to the request daemon, `bprd`. This request identifies the files and client. The request daemon then uses `bpcd` (client daemon) to start the backup and restore manager (`bpbrm`).

Note: To restore Backup Exec images, `bpbrm` invokes `mtfrd.exe` instead of `tar32.exe` on the clients. The server processes are the same as those used for NetBackup restores.

- If the disk or tape where the data resides attaches to the master server, then `bprd` starts the backup and restore manager on the master server. If the disk unit or tape unit connects to a media server, `bprd` starts the backup and restore manager on the media server.
- `bpbrm` starts `bptm` and provides `bptm` with the backup ID and the `shmfat` (shared memory) flag.
- `bptm` does the following:
 - Requests the SAN client information from the Job Manager service (`nbjm`).
 - Sends a restore request to the FT server process (`nbftsrvr`).
 - Sends a restore request to the FT client process on the client (`nbftclnt`).
`nbftclnt` opens a Fibre Channel connection to `nbftsrvr` on the media server, allocates the shared memory, and writes the shared memory information to the backup ID file.
- `bpbrm` starts `tar` by means of `bpcd` and provides `tar` with the backup ID, socket information, and the `shmfat` (shared memory) flag.
- `bptm` does the following:
 - Reads the image from the storage device.
 - Creates a `bptm` child process. This process filters the backup image so that only the files that are selected for the restore are sent to the client.
 - Writes the image data to the shared buffer on the server.
 - When the buffer is full or the job is done, it sets the buffer flag (partial buffers may be sent to the client).
- `tar` does the following:
 - Sends the status and control information to `bpbrm`.
 - Reads the shared memory information from the local backup ID file (waits for the file to exist and become valid).
 - Waits for the buffer flag that indicates the data is ready to be read.
 - Reads the data from the buffer, extracts files, and restores them. When the `shmfat` (shared memory) flag is provided, `tar` considers the data to be already filtered.
- The FT Server process `nbftsrvr` waits for the shared memory buffer flag to be set. `nbftsrvr` then transfers the image data to the FT client (`nbftclnt`) shared memory buffer, and clears the buffer flag.

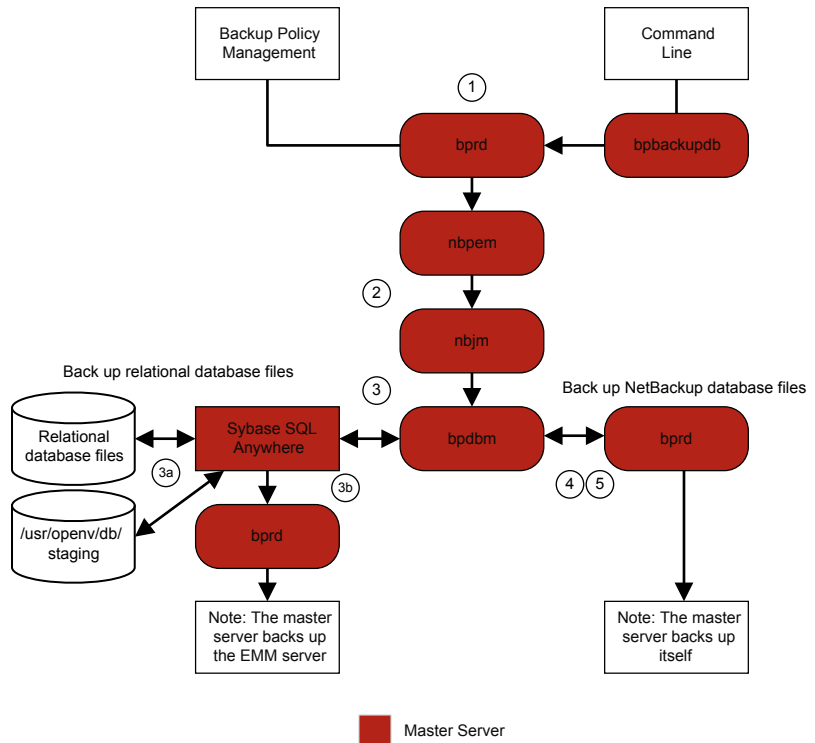
- The FT client (`nbftclnt`) waits for the data from `nbftsrvr` and writes the data to the shared memory buffer on the client. `nbftclnt` then sets the buffer flag.
- At the end of the job:
 - `bptm` informs `tar` and `bpbrm` that the job is complete.
 - `bptm` directs `nbftclnt` to close the Fibre Channel connection.
 - `nbftclnt` closes the Fibre Channel connection and deletes the BID file.

Hot catalog backup

The hot catalog backup is a policy-based backup, with all of the scheduling flexibility of a regular backup policy. This backup type is designed for highly active NetBackup environments where other backup activity usually takes place.

You can use an option in the NetBackup Administration Console to start a manual backup of the NetBackup catalogs. Or, you can configure a NetBackup policy to automatically back up its catalogs.

[Figure 5-3](#) shows the hot catalog backup process.

Figure 5-3 Hot catalog backup process

NetBackup initiates the following hot catalog backup jobs:

- A parent job that is started manually by the administrator or by a catalog backup policy schedule.
- A child job that copies NBDB to the staging directory and validates the information.
The SQL Anywhere files database agent makes an online copy of the relational database files to `/usr/opensv/db/staging`.
- A child job that backs up the NBDB database files.
After the files are in the staging area, the SQL Anywhere database agent backs them up in the same manner as an ordinary backup.
- A child job that backs up the NetBackup database files (all files in `/usr/opensv/netbackup/db`).

NetBackup creates the disaster recovery file, and emails it to the administrator if the email option was selected in the policy.

Consult the following logs for messages on hot catalog backup:

- `bpdbm`, `bpbkar`, `bpbm`, `bpcd`, `bpbackup`, `bprd`

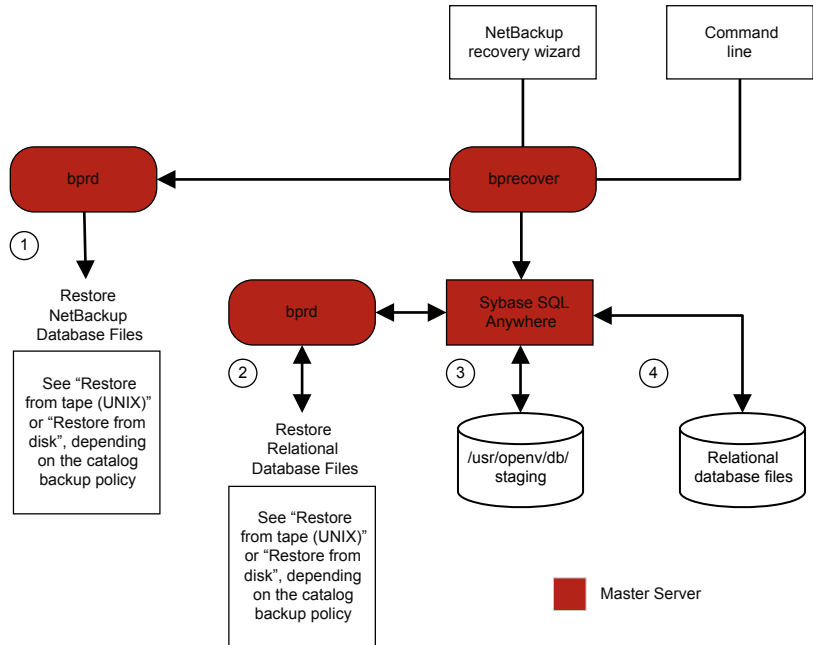
For messages pertaining only to the relational database files, see the `EMM server.log` file and the `bpdbm` log file in the following directories:

- **UNIX:** `/usr/opensv/netbackup/logs/bpdbm`
`/usr/opensv/db/log/server.log`
- **Windows:** `install_path\NetBackup\logs\bpdbm`
`install_path\NetBackupDB\log\server.log`

Hot catalog restore

You can start a catalog restore with the NetBackup Catalog Recovery Wizard in the NetBackup Administration Console, or with the `bprecover` command. More information is available in the "Disaster Recovery" chapter of the [NetBackup Troubleshooting Guide](#).

[Figure 5-4](#) illustrates the catalog restore and recovery process.

Figure 5-4 Catalog restore and recovery

A restore of the NetBackup database and relational database (NBDB) files from a hot catalog backup consists of the following steps (in the order presented):

- The NetBackup catalog image and configuration files are restored.
- The NBDB files are restored. The database files are restored to `/usr/opencv/db/staging` (UNIX), or to `install_path\NetBackupDB\staging` (Windows).
- After the files are restored to the staging directory, NBDB is recovered.
- The NBDB files are moved from the staging directory to a location that is determined by the following: The `bp.conf` file `VXDBMS_NB_DATA` setting on UNIX and by the corresponding registry key on Windows. The default location is `/usr/opencv/db/data` on UNIX, and `install_path\NetBackupDB\data` on Windows.

If the relational database files are relocated, they are moved from the staging directory to the `/usr/opencv/db/data/vxdbms.conf` file (UNIX) or the `install_path\NetBackupDB\data\vxdbms.conf` file (Windows). For information

on how to relocate the NetBackup relational database files after installation, see the [NetBackup Administrator's Guide, Volume I](#).

Synthetic backups

The typical NetBackup backup process accesses the client to create a backup. A synthetic backup is a backup image created without using the client. Instead, a synthetic backup process creates a full or a cumulative incremental image by using previously created backup images called component images.

Note: Synthetic archives do not exist.

For example, an existing full image and subsequent differential incremental images can be synthesized to create a new full image. The previous full image and the incrementals are the component images. The new synthetic full image behaves like a backup that is created through the traditional process. The new synthetic full image is a backup of the client that is as current as the last incremental. The synthetic image is created by copying the most current version of each file from the most recent component image that contains the file. A synthetic backup must be created in a policy with the **True Image Restore with Move Detection** option selected. This option enables the synthetic backup to exclude the files that have been deleted from the client file system from appearing in the synthetic backup.

Like a traditional backup, `nbpem` initiates a synthetic backup. It submits a request to `nbjm` to start the synthetic backup process and `nbjm` then starts `bpsynth`, which executes on the master server. It controls the creation of the synthetic backup image and the reading of the files that are needed from the component images. If directory `bpsynth` exists in the debug log directory, additional debug log messages are written to a log file in that directory.

`bpsynth` makes a synthetic image in several phases:

Table 5-1

Phase	Description
1 - Prepare catalog information and extents	<p>In phase 1, <code>bpsynth</code> makes a synthetic backup request to the database manager, <code>bpdbm</code>. It uses the entries and the TIR information from the catalogs of the component images to build the catalog for the new synthetic image. It also builds the extents to be copied from the component images to the synthetic image. The <code>bpdbm</code> service returns the list of extents to <code>bpsynth</code>. (An extent is the starting block number and the number of contiguous blocks within a specific component image.) A set of extents is typically copied from each component image onto the new synthetic image.</p> <p>The following figure shows how phase 1 operates:</p> <pre> graph TD nbpem([nbpem]) --> nbjm([nbjm]) nbjm --> bpsynth([bpsynth]) bpsynth -- "Request to make Synthetic backup" --> bpdbm([bpdbm]) bpdbm -- "Extents and media needed to form the synthetic backup" --> bpsynth bpdbm <--> Catalog[(Catalog)] style nbpem fill:#800000,color:#fff style nbjm fill:#800000,color:#fff style bpsynth fill:#800000,color:#fff style bpdbm fill:#800000,color:#fff style Catalog fill:#fff,stroke:#000 </pre> <p>The following figure shows how phase 1 operates:</p>
2 - Obtain resources	<p>In phase 2, <code>bpsynth</code> obtains write resources (storage unit, drive, and media) for the new image. It also reserves all the read media containing component images and obtains the drive for the first media to be read.</p> <p>When the component images reside on BasicDisk, no resource reservation is done.</p>

Table 5-1 (continued)

Phase	Description
3 - Copy data	<p>In phase 3, <code>bpsynth</code> starts the writer <code>bptm</code> (for tape and disk) on the media server to write the new synthetic image. It also starts a reader <code>bptm</code> (tape) or <code>bpdm</code> (disk) process for each component image on a media server that can access the component image. The reader process reads all extents for the component image.</p> <p>The following figure shows how phase 3 operates:</p> <pre> graph LR subgraph Master_Server [Master Server] bpsynth end subgraph Media_Server_1 [Media Server] parent_bptm1(parent bptm) child_bptm1(child bptm) end subgraph Media_Server_2 [Media Server] parent_bpdm2(parent bptm or bpdm) child_bpdm2(child bptm or bpdm) end bpsynth <--> parent_bptm1 bpsynth <--> parent_bpdm2 parent_bptm1 --> child_bptm1 parent_bpdm2 <--> child_bpdm2 child_bpdm2 -- Data flow --> parent_bpdm2 parent_bpdm2 --> bpsynth parent_bptm1 --> New_image[New image] Component_image[Component image(s)] --> child_bpdm2 </pre> <p> ■ Master Server ■ Media Server </p> <p>Note that <code>bpsynth</code> only starts the parent <code>bptm</code> (writer) and <code>bpdm</code> (reader) process on the media server. The parent in turn starts a child process. The parent and child communicate by means of buffers in shared memory.</p> <p>The <code>bpsynth</code> process sends the extents (starting block and count) for each component image to the corresponding child <code>bptm</code> or <code>bpdm</code> reader process.</p> <p>The parent <code>bptm</code> or <code>bpdm</code> reader process reads the data from the appropriate media into the shared buffers. The child <code>bptm</code> or <code>bpdm</code> reader process sends the data in the shared buffers to the child <code>bptm</code> writer process over a socket. The child <code>bptm</code> writer process writes the data into the shared buffers. The parent <code>bptm</code> writer process copies the data from the shared buffers to the media and notifies <code>bpsynth</code> when the synthetic image is complete.</p>
4 - Validate the image	<p>In phase 4, the <code>bpsynth</code> process validates the image. The new image is now visible to NetBackup and can be used like any other full or cumulative incremental backup.</p> <p>Synthetic backup requires that true image restore (TIR) with move detection be selected for each component image, and that the component images are synthetic images.</p>

Creating legacy log directories to accompany problem reports for synthetic backup

If the legacy log directories have not been created, you must create them. If the directories do not exist, the logs cannot be written to disk.

Table 5-2 Creating legacy log directories

Step	Action	Description
Step 1	Create directories on the master server.	Create the following directories: <i>install_path/netbackup/logs/bpsynth</i> <i>install_path/netbackup/logs/bpdbm</i> <i>install_path/netbackup/logs/vnetd</i>
Step 2	Create directories on the media server.	Create the following directories: <i>install_path/netbackup/logs/bpcd</i> <i>install_path/netbackup/logs/bptm</i>
Step 3	Change the Global logging level .	In Host Properties , select a master server and set the Global logging level to 5. See the NetBackup Troubleshooting Guide for more information on how to use the Host Properties window to access configuration settings. See “ Changing the logging level ” on page 52. See “ About global logging levels ” on page 50.
Step 4	Rerun the job.	Rerun the job and gather the logs from the directories that you created. The <i>bptm</i> logs are required only if the images are read from or written to a tape device or disk. The <i>bpdm</i> logs are needed only if the images are read from disk. If the images are read from multiple media servers, the debug logs for <i>bptm</i> or <i>bpdm</i> must be collected from each media server.

See “[Logs to accompany problem reports for synthetic backups](#)” on page 110.

Logs to accompany problem reports for synthetic backups

To debug problems with synthetic backups, you must include a complete set of logs in the problem report and additional items. Send all the information to Veritas Technical Support.

Include the following log types:

- Log files that unified logging creates

See [“Gathering unified logs for NetBackup”](#) on page 14.

- Log files that legacy logging creates
See [“Creating legacy log directories to accompany problem reports for synthetic backup”](#) on page 110.

Include the following additional items:

Try file

The try file is located in the following directory:

```
install_path/netbackup/db/jobs/trylogs/jobid.t
```

If the job ID of the synthetic backup job was 110, the try file is named `110.t`.

Policy attributes

Use the following command to capture the policy attributes:

```
install_path/netbackup/bin/admincmd/bppllist  
policy_name -L
```

where *policy_name* is the name of the policy for which the synthetic backup job was run.

List of storage
units

Capture the list of storage units from the following command:

```
install_path/netbackup/bin/admincmd/bpstulist -L
```

See [“Creating legacy log directories to accompany problem reports for synthetic backup”](#) on page 110.

Storage logging

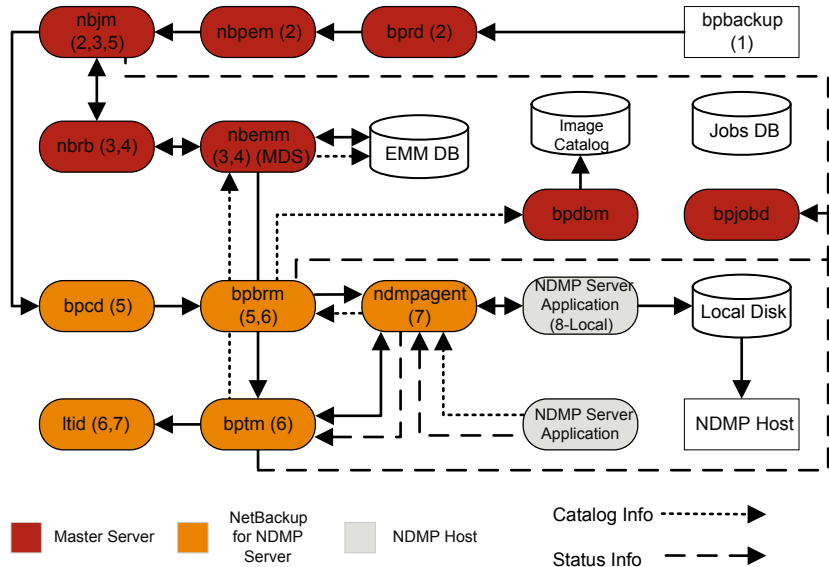
This chapter includes the following topics:

- [NDMP backup logging](#)
- [NDMP restore logging](#)

NDMP backup logging

The following shows an NDMP backup process.

Figure 6-1 NDMP backup process



The basic processing steps for an NDMP backup operation are the following:

NDMP backup procedure

- 1 The NetBackup administrator runs the `bpbbackup` command to start the backup job. Or, a scheduled policy that is created on the NetBackup Administration Console can initiate the job.
- 2 The `bpbbackup` process connects to the master server and creates the backup request. The NetBackup Request Manager (`bprd`) sends the backup request to the Policy Execution Manager (`nbpem`), who submits the job to the Job Manager (`nbjm`).
- 3 `nbjm` requests resources from the Resource Broker (`nbrb`) that are required to run the job. `nbrb` accesses the Media and Device Selection (MDS) of the Enterprise Media Management (`nbemm`) to evaluate the resources request. MDS queries the EMM database to identify the resources to use for this job.
- 4 MDS provides `nbrb` with a list of resources for the job, and `nbrb` passes it on to `nbjm`.
- 5 `nbjm` initiates communication with the media server that is associated with this backup job. It goes through the client service (`bpcd`) to start the Backup and Restore Manager (`bpbrm`) on the media server.
- 6 `bpbrm` starts the Tape Manager (`bptm`) on the media server. Eventually, the parent `bptm` process makes a request to `ltid` to mount the tape to be used for the backup job.
- 7 On the NetBackup for NDMP server, one of the following occurs: sends the necessary NDMP SCSI robotic commands to mount the requested tape on the storage device.
 - The NDMP agent service (`ndmpagent`) connects to the filer that issues the NDMP commands to mount the tape that is directly attached.
 - `ltid` on the media server issues the necessary NDMP SCSI robotic commands to mount the requested tape on the storage device.
- 8 One of the following occurs, depending on the type of NDMP backup:
 - Local backup. NetBackup sends the NDMP commands to have the NDMP server application perform the backup to tape. The data travels between the local disk and the tape drives on the NDMP host without crossing the LAN.
 - Three-way backup (not shown in the process flow diagram). NetBackup sends NDMP commands to the NDMP server application to perform the backup. The media server establishes NDMP communications with both NDMP servers. The data travels over the network from the NDMP server

that houses the data to be backed up to the NDMP server that writes the backup to its tape storage.

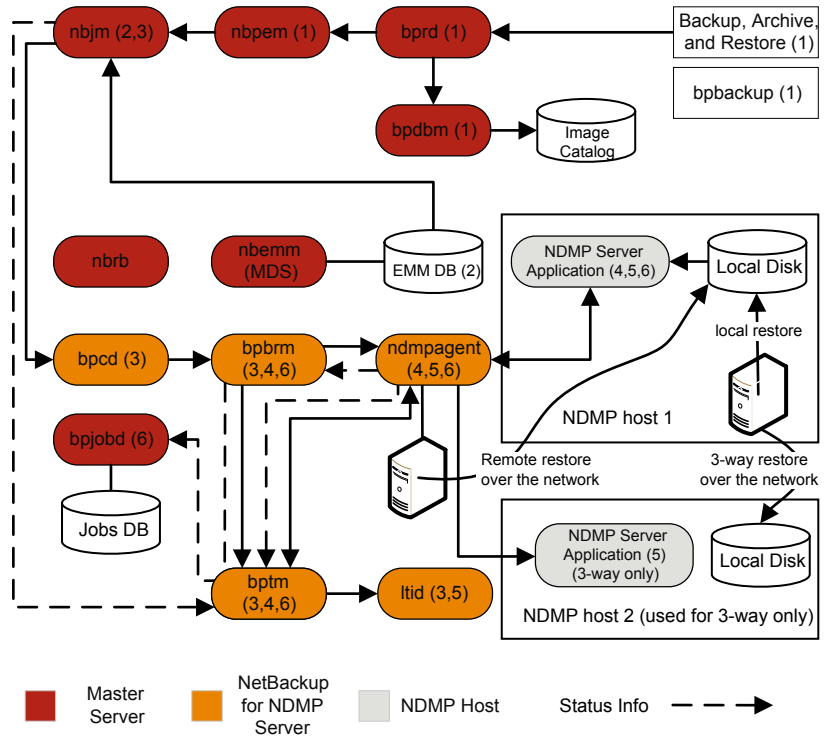
- Remote backup (not shown in the process flow diagram). The device that is used to write the backup is associated with a NetBackup storage unit. `bptm` on the NetBackup media server mounts a tape on a tape drive. NetBackup sends the NDMP commands to the NDMP server to initiate the backup to the non-NDMP media manager storage unit. The data travels over the network from the NDMP host to the NetBackup media server, which writes the data to the selected storage unit.

- 9 Throughout the backup operation and at its completion, the NDMP server sends status about the backup operation to the NetBackup for NDMP server. Several NetBackup processes send information about the job to `bpjobjd`, that uses this information to update the job status that you can view in the NetBackup Activity Monitor.

Status, catalog, and other job information movement is shown in dashed lines in the process flow diagram.

NDMP restore logging

The following shows an NDMP restore process.

Figure 6-2 NDMP restore process

The basic processing steps for an NDMP restore operation are as follows:

NDMP restore procedure

- 1 An administrator at the NetBackup Administration Console on a NetBackup master server or media server initiates a restore job by browsing the images catalog and by selecting the files to be restored from the NDMP images. This process is similar to selecting files to be restored from standard backup images. The NetBackup master server identifies the specific media that is required to perform the restore. In this diagram, the media is a tape volume.
- 2 After the master server identifies the data to be restored and the media required, it submits a restore job. The Job Manager (nbjm) then requests the required resources. This resource request causes the allocation of the media that contains the data to be restored. In this example, a tape drive is used during the restore operation.

- 3 The master server contacts the media server that participates in the restore job, and starts the Restore Manager (`bpbrm`) process to manage the restore job. `bpbrm` starts the Tape Manager process (`bptm`), that queries `nbjm` for the tape volume. Then, `bptm` requests that the logical tape interface daemon (`ltid`) mounts the tape.
- 4 On the NetBackup for NDMP server, the NDMP agent (`ndmpagent`) connects to the filer and issues NDMP commands to mount the tape that is directly attached, and `ltid` sends NDMP commands to mount the requested tape on the storage device. Or, the media server itself issues tape mount requests much like a regular media manager storage unit.
- 5 One of the following occurs, depending on the type of NDMP restore operation:
 - Local restore. NetBackup sends the NDMP commands to the NDMP server to initiate the restore operation from a tape drive to a local disk. The restore data travels from a tape drive to a local disk on the NDMP host without traversing the LAN.
 - Three-way restore. The NetBackup media server establishes NDMP communications with both of the NDMP servers that are involved in the restore. To initiate the restore of data from tape on one NDMP server to disk storage on the other NDMP server, the media server sends NDMP commands to both NDMP servers. The restore data travels over the network between the NDMP hosts.
 - Remote restore. NetBackup sends the NDMP commands to the NDMP server to prepare the server for the restore. `bptm` on the media server reads the restore data from tape and sends it over the network to the NDMP host where the data is written to disk storage.
- 6 The NDMP server sends status information about the restore operation to the NetBackup for NDMP server. Various NetBackup processes (`nbjm`, `bpbrm`, `bptm`, and others) send job status information to the master server. The Jobs Database Manager (`bpjobjd`) process on the master server updates the restore job status in the jobs database. You can view this status in the Activity Monitor.

NetBackup Deduplication logging

This chapter includes the following topics:

- [Deduplication backup process to the Media Server Deduplication Pool \(MSDP\)](#)
- [Client deduplication logging](#)
- [Deduplication configuration logs](#)
- [Media server deduplication/pdplugin logging](#)
- [Disk monitoring logging](#)
- [Logging keywords](#)

Deduplication backup process to the Media Server Deduplication Pool (MSDP)

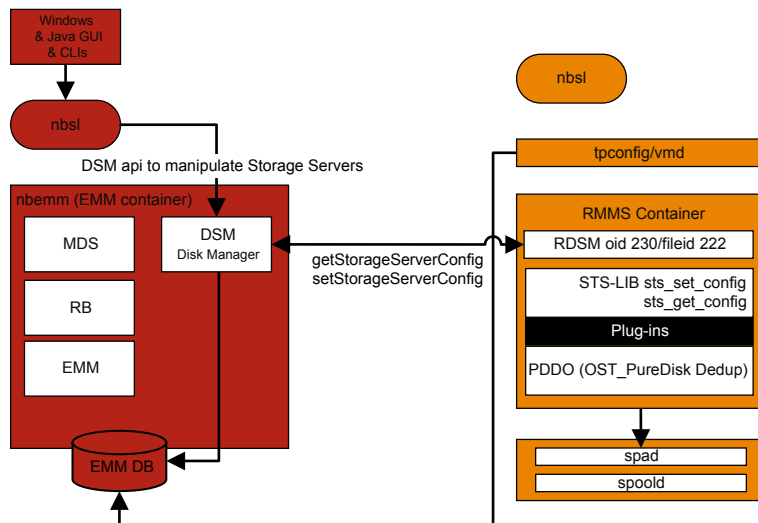
The deduplication backup process to the Media Server Deduplication Pool (MSDP) is as follows:

- The client `bpbkar` sends data to the NetBackup backup tape manager - the `bptm` process
- `pdvfs` (using `bptm` as a proxy) connects to the NetBackup Deduplication Manager (`spad`) to record metadata (image records) in the `spadb` mini-catalog and connects to the NetBackup Deduplication Engine (`spoold`) to store the image data in the `.bhd/.bin` files in the data directory (`dedup_path\data`)
- `spoold` writes `tlogs` to the `.tlog` files in the queue (`dedupe_path\queue`) directory and to the processed directory. The `tlog` data from the queue directory

is processed into the `crdb` later when the next content router queue processing job runs. Beginning with NetBackup 7.6, `.tlog` files do not contain additions to the database.

The functional overview is as follows:

Figure 7-1 Deduplication configuration for MSDP



In this scenario, the client backs up data directly to the media server and the media server deduplicates the data before it stores it locally. Ensure that this is on the correct media server which is not always the same as the MSDP storage server (due to load balancing).

For deduplication-specific logging, enable the following on the media server:

1. Verbose 5 bptm logging:
 - Create a log directory named `bptm` in `/usr/opensv/netbackup/logs` (Windows: `install_path\NetBackup\logs`)
 - Set the `bptm` log verbosity to 5 in the NetBackup Administration Console. To do this, click on **Host Properties > Logging** for the media server. If you use UNIX/Linux, set the `bptm` log verbosity to 5 in the `/usr/opensv/netbackup/bp.conf` file by appending the following line:

```
BPTM_VERBOSE = 5
```

- Edit the `pd.conf` configuration file that is located at:

Windows:

`install_path\NetBackup\bin\ost-plugins\pd.conf`

UNIX/Linux:

`/usr/opensv/lib/ost-plugins/pd.conf`

and uncomment and/or modify the following line:

`LOGLEVEL = 10`

Note: You can also modify `DEBUGLOG` in the `pd.conf` file to specify a path to which to log; however, we recommend leaving the `DEBUGLOG` entry commented out. The logging information (`PDVFS` debug logging) then logs to the `bptm` and `bpdm` logs.

2. Enable verbose `spad/spoold` logging (optional).

- Edit the `dedup_path\etc\puredisk\spa.cfg` and

`dedup_path\etc\puredisk\contentrouter.cfg` files so that the following line:

`Logging=long,thread` is changed to `Logging=full,thread`

- Ensure that you are on the correct media server and restart the MSDP storage server services.

Caution: If you enable verbose logging, it can impact the performance on MSDP.

3. Reproduce the backup failure.
4. Within the NetBackup Administration Console, click on **Activity Monitor > Jobs**, open the job details and click the **Detailed Status** tab. It displays the media server host name that ran the backup and the `bptm` process ID number (PID).
 - Find a line similar to `bptm (pid=value)`; this value is the `bptm` PID to locate in the `bptm` log.
5. Extract the `bptm` PID found in step 3 from the `bptm` log on the media server. This step only gathers the single-line entries; review the raw logs to see the multi-line log entries. In the following examples, 3144 is the `bptm` PID:
 - Windows command line:

`findstr "[3144." 092611.log > bptmpid3144.txt`

- UNIX/Linux command line:

```
grep "[3144\]" log.092611 > bptmpid3144.txt
```

6. Gather the `spoold` session logs that cover the dates from when the backup started and when it failed from the following logs:

Windows:

```
<dedup_path>\log\spoold\<mediasvr_IP_or_hostname>\bptm\Receive\MMDDYY.log
<dedup_path>\log\spoold\<mediasvr_IP_or_hostname>\bptm\Store\MMDDYY.log
```

UNIX/Linux:

```
<dedup_path>/log/spoold/<mediasvr_IP_or_hostname>/bptm/Receive/MMDDYY.log
<dedup_path>/log/spoold/<mediasvr_IP_or_hostname>/bptm/Store/MMDDYY.log
```

Client deduplication logging

Client deduplication logging uses the logs at the following location; select one of the following deduplication location options. On the applicable MSDP storage pool, edit `install_path\etc\puredisk\spa.cfg` and `install_path\etc\puredisk\contentrouter.cfg` and specify **Logging=full,thread** and then restart the `spad` and `spoold` services in order for the changes to take effect.

- The client-side log (NetBackup Proxy Service log) is as follows:

Windows:

```
install_path\NetBackup\logs\nbostpxy
```

UNIX/Linux:

```
/usr/opensv/netbackup/logs/nbostpxy
```

PBX (nbostpxy (OID450):

```
vxlogcfg -a -p 51216 -o 450 -s DebugLevel=6 -s DiagnosticLevel=6
```

- The media server log is as follows:

`bptm` and `storage_path\log\spoold\IP_address\nbostpxy.exe*`

Deduplication configuration logs

The following are the deduplication configuration logs.

NetBackup Administration Console for Windows wizard logging:

1. wingui (OID: 263):

```
# vxlogcfg -a -p 51216 -o 263 -s DebugLevel=6 -s DiagnosticLevel=6
```

2. On the applicable MSDP storage pool, edit

install_path\etc\puredisk\spa.cfg and

install_path\etc\puredisk\contentrouter.cfg. Specify

Logging=full,thread and then restart the spad and spoold services for the changes to take effect.

■ nbsl (OID: 132):

```
vxlogcfg -a -p 51216 -o 132 -s DebugLevel=6 -s DiagnosticLevel=6
```

■ dsm (OID: 178):

```
vxlogcfg -a -p 51216 -o 178 -s DebugLevel=6 -s DiagnosticLevel=6
```

3. Storage service (turn on STS logging, to log the msdp/pdplugin responses to NetBackup):

```
# vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6
```

4. Remote Monitoring & Management Service:

```
# vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
```

5. tpcommand (... \volmgr\debug\tpcommand)

6. *storage_directory\log\msdp-config.log*

Command-line configuration logging:

■ Administration log for nbdevquery (add *storage_server*)

■ tpcommand log for tpconfig (add credentials) (... \volmgr\debug\tpcommand)

■ *storage_directory\log\pdde-config.log*

■ Storage service (turn on STS logging, to log the msdp/pdplugin responses to NetBackup):

```
# vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6
```

■ Remote Monitoring and Management Service:

```
# vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
```

■ *storage_directory\log\pdde-config.log*

NetBackup Administration Console logging:

First, open the `Debug.Properties` file, in `C:\Program Files\VERITAS\Java` (for Windows) or `/usr/opensv/java` (for UNIX/Linux). Then, edit the file so the following lines are uncommented (or append the lines if they are not present). If you have a GUI that is running, be sure to restart it.

```
printcmds=true
printCmdLines=true
debugMask=0x0C000000
debugOn=true
```

The logs are located under `C:\Program`

`Files\VERITAS\NetBackup\logs\user_ops\nbjlogs` (Windows) or `/opt/opensv/netbackup/logs/user_ops/nbjlogs` (UNIX/Linux). Ensure that you look at the most recent log.

- Storage service (turn on STS logging, to log the `msdp/pdplugin` responses to NetBackup):

```
# vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6
```
- Remote Monitoring and Management Service:

```
# vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
```
- `tpcommand (... \volmgr\debug\tpcommand)`
- `storage_directory\log\msdp-config.log`

Media server deduplication/pdplugin logging

This topic describes the media server deduplication/pdplugin logging.

- Unless you are troubleshooting the Private Branch Exchange (PBX) communication between the client direct and its media server, reduce the unnecessary CORBA/TAO to zero (0) for deduplication logging by using the following command:

```
# vxlogcfg -a -p NB -o 156 -s DebugLevel=0 -s DiagnosticLevel=0
```

For backups:

- Enable verbose 5 `bptm` on the media servers to read/write backups
- Uncomment `LOGLEVEL = 10` in the media server `pd.conf` file

For duplications or replications:

- Enable verbose 5 `bpdm` on the media server(s) to read/write duplications
- Uncomment `LOGLEVEL = 10` in the media server `pd.conf` file

Caution: If you enable verbosity, it can affect performance.

- Enable trace level `spad` and `spoold` logging so that the failing duplication or replication job can be traced across `bpdm/pdvfs > source spad/spoold session log > source replication.log > target spad/spoold session logs`

Disk monitoring logging

STS logging should be configured on any media server that has credentials to communicate to the MSDP storage pool. `nbrmms` (OID: 222) should be configured on the master server and any applicable media servers. You can monitor the disks using the logs at the following location:

- Storage service (turn on the STS logging to show the response that NetBackup receives when it runs the MSDP plug-in):

```
# vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6
```
- Remote Monitoring and Management Service:

```
# vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
```

Logging keywords

Support uses the following keywords when it reviews the logs.

Keyword	Description
maximum fragment size	Should be 51200 KB or less
get_plugin_version	<code>libstspipd.dll</code> (pdplugin version)
get_agent_cfg_file_path_for_mount	Uses the PureDisk agent configuration file (note the <code>.cfg</code> file name); determines short name or FQDN.
emmlib_NdmpUserIdQuery	Used for backups, the credential check
Resolved	Name resolution of the remote CR
tag_nbu_dsid read	Checks if it read the <code>NBU_PD_SERVER</code> object correctly
Recommended routing table	CR routing table for the CR's to route fingerprint/so's; more useful when PDDO targets PureDisk.
for primary backups	Primary backup dsid
for opt-dup copies from	opt-dup dsid

Keyword	Description
this is opt-dup	opt-dup dsid
https	Web service calls to either SPA or CR to check if they completed

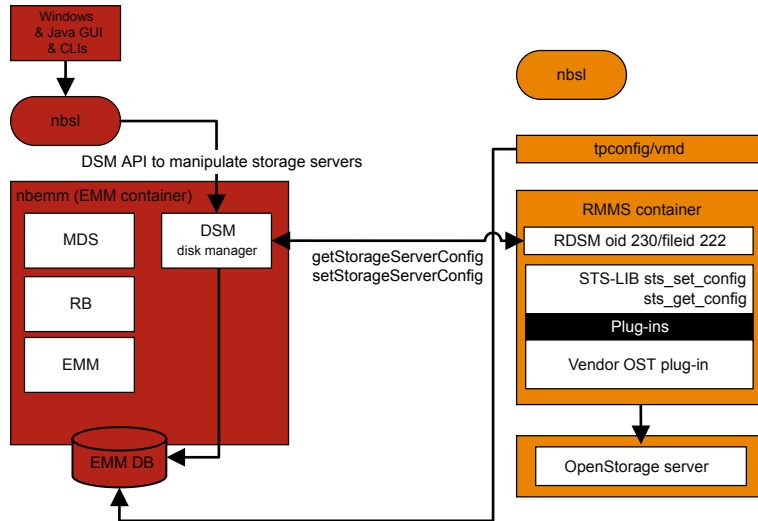
OpenStorage Technology (OST) logging

This chapter includes the following topics:

- [OpenStorage Technology \(OST\) backup logging](#)
- [OpenStorage Technology \(OST\) configuration and management](#)

OpenStorage Technology (OST) backup logging

The following shows the OpenStorage Technology (OST) configuration.

Figure 8-1 OST configuration

In this scenario, the client backs up the data directly to the media server and the media server accesses the vendor plug-in to transfer the data to the storage server.

For logging that is specific to OST, enable the following on the media server or plug-in host:

1. In the registry or `bp.conf` file, set `VERBOSE = 5`.
2. Ensure that the following directories exist under `/usr/opensv/netbackup/logs` (for Windows, use `install_path\NetBackup\logs`):
 - `bptm`
 - `bpbrm`
 - `bpstsinfo`
3. Create the `volmgr/debug/tpcommand` directory.
4. Put `VERBOSE` in the `vm.conf` file. See [“How to control the amount of information written to legacy logging files”](#) on page 46.
5. Set `DebugLevel=6` and `DiagnosticLevel=6` for the following processes:
 - `OID 178` (Disk Manager Service, `dsm`)

- OID 202 (Storage service, stssvc)
- OID 220 (Disk Polling Service, dps)
- OID 221 (Media Performance Monitor Service)
- OID 222 (Remote Monitoring & Management Service)
- OID 230 (Remote Disk Manager Service, rdsm)
- OID 395 (STS Event Manager, stsem)

These OIDs all log to the `nbrmms` unified log file on the media server.

6. Increase the vendor plug-in logging. Most vendors have their own plug-in logging in addition to what is logged within the NetBackup logs.
7. Reproduce the backup failure.
8. Within the NetBackup Administration Console, click on **Activity Monitor > Jobs**, open the job details and click the **Detailed Status** tab. It displays the media server host name that ran the backup and the `bptm` process ID number (PID).
 - Find a line similar to `bptm(pid=value)`; this value is the `bptm` PID to locate in the `bptm` log.
9. Extract the `bptm` PID found in step 8 from the `bptm` log on the media server. This step gathers only the single-line entries; review the raw logs to see the multi-line log entries. In the following examples, 3144 is the `bptm` PID:
 - Windows command line:

```
findstr "[3144." 092611.log > bptmpid3144.txt
```
 - UNIX/Linux command line:

```
grep "[3144\]" log.092611 > bptmpid3144.txt
```
10. Gather the vendor specific plug-in logs that cover the dates from when the backup started and when it failed.

OpenStorage Technology (OST) configuration and management

The OpenStorage Technology (OST) technology uses a plug-in architecture, similar to a software driver, that lets the third-party vendors direct the NetBackup data streams and metadata into their devices. The plug-in is developed and created by

the OST partner and it resides on the media server for use by NetBackup. NetBackup depends on the OST plug-in for a path to the storage server.

Communication to the storage server is through the network; name resolution on the media server and the storage server must be configured correctly. All supported vendor plug-ins can communicate over a TCP/IP network and some can also communicate to the disk storage on a SAN network.

To determine the capabilities of a disk appliance, NetBackup uses the plug-in to query the storage appliance. The capabilities can include deduplicated storage, optimized off-host duplication, and synthetic backups.

Each OST vendor can report different log messages. A review of the `bptm` log and/or plug-in log for a backup or a restore job is the best way to understand the specific calls made to the storage server through the plug-in.

The basic steps include the following:

- Claim the resource
- `sts open_server`
- Create the image
- `write`
- `close`
- `sts close_server`

The example of calls in a vendor plug-in log are as follows:

```
2016-03-14 09:50:57 5484: --> stspi_claim
2016-03-14 09:50:57 5484: --> stspi_open_server
2016-03-14 09:50:57 5484: <-- stspi_write_image SUCCESS
2016-03-14 09:50:57 5484: --> stspi_close_image
2016-03-14 09:50:59 5484: <-- stspi_close_server SUCCESS
```

To display the plug-in version, use the following commands:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/bpstsinfo -pi`
- **Windows:** `install dir\netbackup\bin\admincmd\bpstsinfo -pi`

To test the basic communication to the storage server, use the following commands:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/bpstsinfo -li -storage_server storage_server_name -stype OST_TYPE`
- **Windows:** `install dir\netbackup\bin\admincmd\bpstsinfo -li -storage_server storage_server_name -stype OST_TYPE`

To display the configured storage servers, use the following commands:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/nbdevquery -liststs -stype OST_TYPE -U`
- **Windows:** `install dir\netbackup\bin\admincmd\nbdevquery -liststs -stype OST_TYPE -U`

To show the configured disk pools, use the following commands:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/nbdevquery -listdp -stype OST_TYPE -U`
- **Windows:** `install dir\netbackup\bin\admincmd\nbdevquery -listdp -stype OST_TYPE -U`

To show the configured disk volumes, use the following commands:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/nbdevquery -listdv -stype OST_TYPE -U`
- **Windows:** `install dir\netbackup\bin\admincmd\nbdevquery -listdv -stype OST_TYPE -U`

Review the flags in the diskpool information, for example:

- `CopyExtents` - supports optimized duplications
- `OptimizedImage` - supports optimized synthetics and accelerator
- `ReplicationSource` - supports AIR (replication)
- `ReplicationTarget` - supports AIR (imports)

After the initial configuration of the diskpools, you must run the `nbdevconfig -updatedp` command as follows to recognize any new flag that the vendor added:

- **UNIX/Linux:** `/usr/opensv/netbackup/bin/admincmd/nbdevconfig -updatedp -stype OST_TYPE -dp diskpool -M master`
- **Windows:** `install dir\netbackup\bin\admincmd\nbdevconfig -updatedp -stype OST_TYPE -dp diskpool -M master`

To manually add the supported flags, you can use the following commands:

- `nbdevconfig -changests -storage_server storage_server_name -stype OST_TYPE -setattribute OptimizedImage`
- `nbdevconfig -changedp -stype OST_TYPE -dp diskpool name -setattribute OptimizedImage`

You should also review the following flag for the storage server:

- `OptimizedImage` - supports accelerator

To list the OpenStorage credentials for all of the media servers, use the following commands:

- **UNIX/Linux:** `/usr/openv/volmgr/bin/tpconfig -dsh -all_hosts`
- **Windows:** `install dir\volmgr\bin\tpconfig -dsh -all_hosts`

Storage lifecycle policy (SLP) and Auto Image Replication (A.I.R.) logging

This chapter includes the following topics:

- [About storage lifecycle policies \(SLPs\) and Auto Image Replication \(A.I.R.\)](#)
- [Storage lifecycle policy \(SLP\) duplication process flow](#)
- [Automatic Image Replication \(A.I.R.\) process flow logging](#)
- [Import process flow](#)
- [SLP and A.I.R. logging](#)
- [SLP configuration and management](#)

About storage lifecycle policies (SLPs) and Auto Image Replication (A.I.R.)

A storage lifecycle policy (SLP) contains instructions in the form of storage operations that are applied to the data.

The Auto Image Replication (A.I.R.) lets backups be replicated between the NetBackup domains. A.I.R. automatically creates the catalog entries in the target domain as the backups are replicated. Veritas recommends the use of A.I.R. instead of live catalog replication to populate the NetBackup catalog at a disaster recovery site.

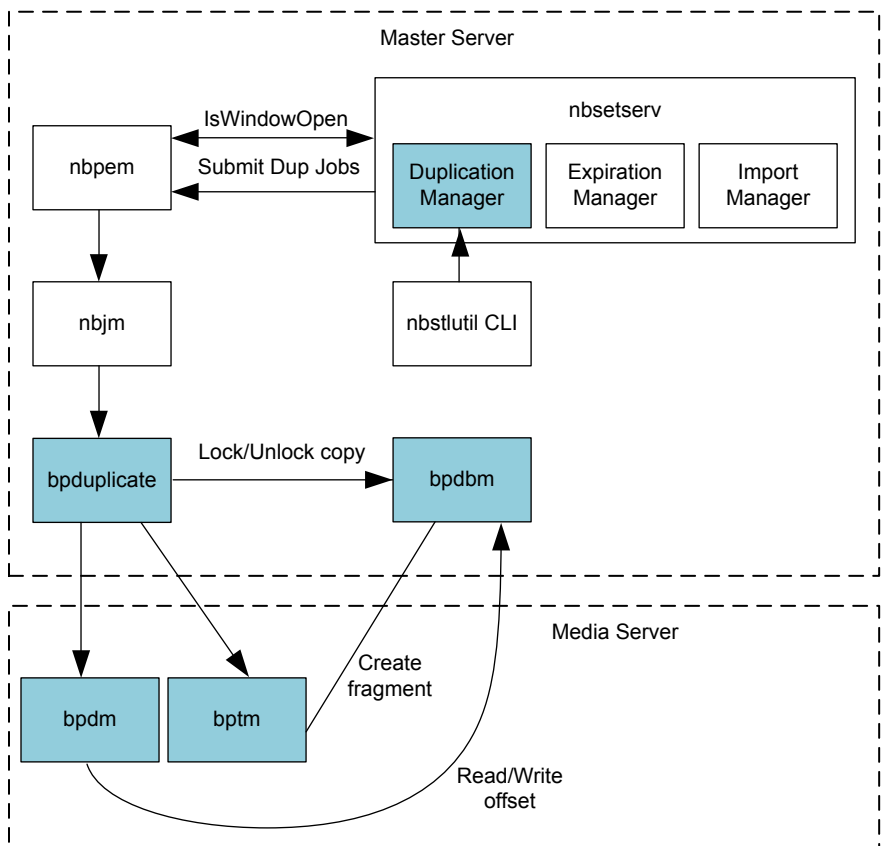
Understanding the storage lifecycle policy (SLP) operations (for example, backup, duplication, replication, import, and snapshot) can help determine which logs can be used to troubleshoot an issue. This topic primarily focuses on the Automatic Image Replication (A.I.R.) and duplication process flows. The process flow for other operations, like backups and snapshots, are covered in other topics of this guide.

See the [NetBackup Administrator's Guide, Volume I](#) for more information about SLPs and A.I.R.

Storage lifecycle policy (SLP) duplication process flow

The following figure describes the SLP duplication process flow.

Figure 9-1 SLP duplication process flow



The SLP duplication process flow is as follows:

1. The SLP manager (`nbstserv`) checks if the duplication window is open to submit duplication jobs. When it finds an SLP window open to submit a duplication job, it will process the relevant images managed by the SLP policies, batch them, and submit them to `nbpem` for further processing.
2. `nbpem` also checks if the SLP window is still open for the duplication operation. If it is, `nbpem` creates the duplication job structure and submits it to `nbjm`.
3. `nbjm` requests resources as it would for backups (not shown in the figure), and then invokes `bpduplicate`.
4. `bpduplicate` starts the required `bpdm` and/or `bptm` processes, media load operations occur (not shown in the diagram), the image is read from the local source storage, and then written to the local destination storage.
5. After the media server `bpdm/bptm` processes the exit, `bpduplicate` also exits.

Automatic Image Replication (A.I.R.) process flow logging

Note: For A.I.R. replications, only MSDP or OST disk-based storage units are used. The tape storage units cannot be used with A.I.R. The basic disk storage units are not supported with SLP.

The Automatic Image Replication (A.I.R.) process flow is as follows:

1. The SLP-controlled backup finishes. The backup image includes information about what SLP policy it will use for its secondary operation; for example, a replication or a duplication.
2. `nbstserv` on a regular interval (SLP parameter - Image Processing Interval) works to batch up images for the replication. The SLP manager (`nbstserv`) checks if the SLP window is open to submit replication jobs.
3. Next, `nbstserv` submits the batch to `nbpem`. `nbpem` passes the job to `nbjm`, which checks for resources from `nbrb` and `nbemm`. If the SLP window is open, `nbpem` passes the job to `nbjm`.
4. `nbjm` starts `nbreplicate` (`nbreplicate` appears in the `admin` log) and passes `nbreplicate` to `bpdm`.
5. `bpdm` makes the physical resource requests to `nbjm`.

6. The replication checks are run and the replication starts. `bpdm` lets the source storage server know when to initiate the replication. The source and target storage servers then communicate to perform the actual replication of data.

Note: For replications, one `bpdm` process controls the operation.

7. A replication event is sent to the remote or target storage server.
8. The replication finishes and the image copy records are updated.

Import process flow

The import process flow is as follows:

1. The media server that is responsible for monitoring the disk storage polls the storage for the A.I.R. import events. It is the `nbrmms` process that does the polling. The image associated with the import event is sent to the import manager (running within `nbstserv`) on the master server.
2. The import manager (OID 369) inserts the image records into the NBDB database.
3. On a regular interval, `nbstserv` looks for images that need to be imported. It batches up the images to be imported and sends the request to `nbpem`. `nbpem` passes the job to `nbjm` and then checks for resources from `nbrb` and `nbemm`.
4. `nbjm` starts `bpimport`. For replicated images, a fast import is run since most of the information that NetBackup needs for the image was brought in when the import event was received.
5. `bpimport` (admin log) starts `bpdm` on the media server.
6. `bpdm` obtains the physical resources needed from `nbjm`.
7. `bpdm` reads the image information and sends it to `bpdbm` on the master server.
8. The import of the image completes and is validated by `bpdbm`.

SLP and A.I.R. logging

`nbstserv` (master server):

```
vxlogcfg -a -p NB -o 226 -s DebugLevel=6 -s DiagnosticLevel=6
```

`importmgr` (master server, import manager logs within the 226 `nbstserv` log):

```
vxlogcfg -a -p NB -o 369 -s DebugLevel=6 -s DiagnosticLevel=6
```

`nbrmms` (logs on the media server responsible for monitoring the disk storage):

```
vxlogcfg -a -p NB -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
```

`stsem` (storage server event manager, `stsem` logs within the 222 `nbrmms` log):

```
vxlogcfg -a -p NB -o 395 -s DebugLevel=6 -s DiagnosticLevel=6
```

On the media servers that perform the duplication, view the appropriate `bpdm` and `bptm` legacy logs. On the media server that initiates the A.I.R. replication operation and on the media server that performs the subsequent import, you can view the `bpdm` legacy log for additional details.

```
bpdm (verbose 5)
```

```
bptm (verbose 5)
```

You can increase the plugin logging to get additional details within `bptm/bpdm` or the third-party vendors OST plugin log file regarding the duplication, replication, and import operations.

On the master server, the following legacy logs are also helpful to review:

- `admin` - (the `admin` log logs the `bpduplicate` or `nbrePLICATE` command for the job)
- `bpdbm` - (the NetBackup Database Manager program that contains backup policy information, such as files, media, and client information)

SLP configuration and management

To view the configured SLP policies using the CLI, run the following command:

```
nbstl -L -all_versions
```

To list the images that are under SLP control (that is, they are waiting for the completion of their secondary operations), use the following command:

```
nbstlutil list -image_incomplete
```

To display the SLP backlog, use the following command:

```
nbstlutil report
```

To display the SLP parameters using the CLI, the `bpgetconfig` command can be run on the master server:

- **UNIX:** `bpgetconfig | grep SLP`
- **Windows:** `bpgetconfig | findstr SLP`

To list images that have been replicated using A.I.R. (run on the source master server), use the following command:

```
nbstlutil repllist
```

To list images that are pending an A.I.R. import into the target environment (run on the target master server), use the following command:

```
nbstlutil pendimplist
```

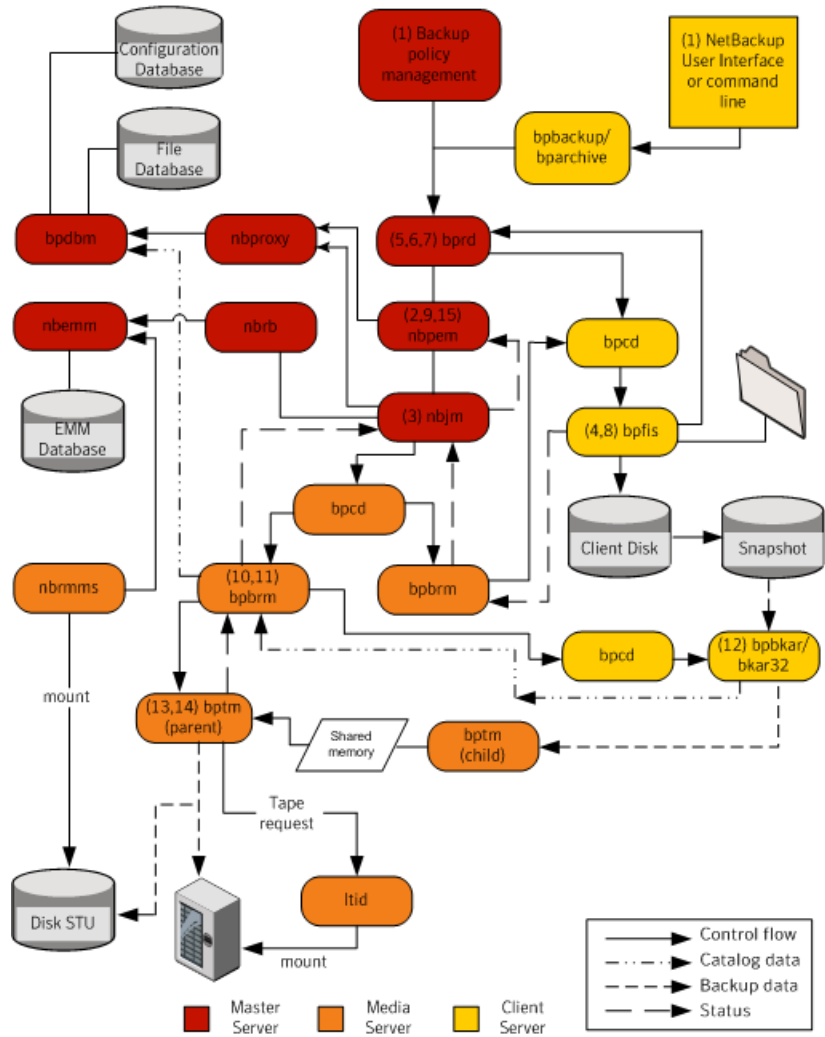
Snapshot technologies

This chapter includes the following topics:

- [Snapshot Client backup](#)
- [VMware backup](#)
- [Snapshot backup and Windows open file backups](#)

Snapshot Client backup

The following shows a typical snapshot backup process. In this scenario, the snapshot is created on the client and is then backed up to a storage unit (disk or tape) from that client. With the exception of Windows open file backups that do not use multiple data streams, all snapshots are created by a separate parent job, followed by a child job that backs up the snapshot. For non-multistreamed Windows Open File Backups, `bpbrm` using `bpcd` invokes `bpfis` to take a snapshot of individual drives. If you use System State or Shadow Copy Component backups, `bpbkars2` creates the snapshot using Volume Shadow Copy Service (VSS). Windows Open File Backups do not require a Snapshot Client license, although they do use Snapshot Client components, such as `bpfis`.



The basic processing steps for snapshot creation and backup are the following (this includes Windows open file backups that employ multiple data streams):

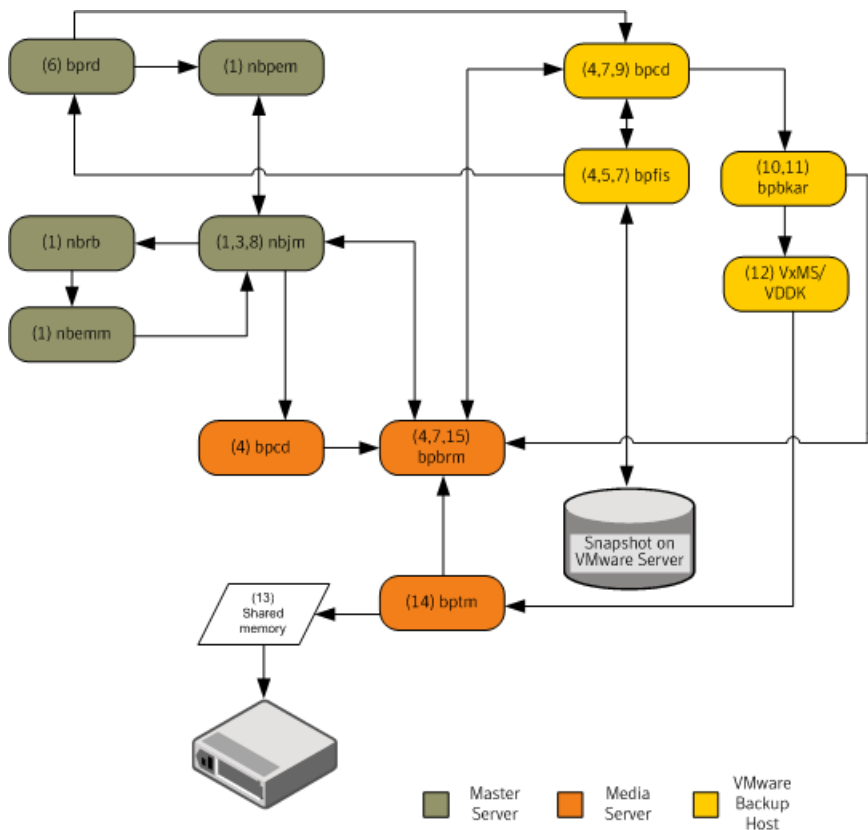
Snapshot Client backup procedure

- 1 The NetBackup master server or primary client initiates the backup, which causes the NetBackup request daemon (`bprd`) to submit a backup request to the Policy Execution Manager (`nbpem`). `nbpem` processes the policy configurations.
- 2 `nbpem` uses `nbjm` to start a parent job to create the snapshot. This job is separate from the job that backs up the snapshot.
- 3 `nbjm` starts an instance of `bpbrm` through `bpcd` on the media server. `bpbrm` starts `bpfis` through `bpcd` on the client.
- 4 `bpfis` creates a snapshot of the client data by means of a snapshot method.
- 5 `bpfis` contacts `bprd` to request transfer of `bpfis` state files from client to server. This operation is enabled by default.
- 6 `bprd` requests `bpcd` on the client to send a list of `bpfis` state files.
- 7 `bprd` copies each state file from the client to the master.
- 8 `bpfis` sends snapshot information and completion status to `bpbrm` and exits. `bpbrm`, in turn, reports the snapshot information and status to `nbjm` and exits. `nbjm` relays the information and status to `nbpem`.
- 9 `nbpem` submits to `nbjm` a child job for the backup with a file list derived from the snapshot information. `nbjm` starts `bpbrm` to back up the snapshot.
- 10 `bpbrm` starts `bpbkcar` on the client. `bpbkcar` sends the file catalog information to `bpbrm`, which relays it to the NetBackup file database (`bpdbm`) on the master server.
- 11 `bpbrm` starts the process `bptm` (parent) on the media server.
- 12 One of the following occurs: The next step depends on whether the media server backs up itself (`bptm` and `bpbkcar` are on the same host) or the media server backs up a client that resides on a different host.
 - If the media server backs up itself, `bpbkcar` stores the snapshot-based image block-by-block in shared memory on the media server.
 - If the media server backs up a client that resides on a different host, the `bptm` process on the server creates a child process of itself. The child receives the snapshot-based image from the client by means of socket communications and then stores the image block-by-block in shared memory.
- 13 The original `bptm` process takes the backup image from shared memory and sends it to the storage device (disk or tape).

- 14 bptm sends the backup completion status to bpbrm, which passes it to nbjm.
- 15 When nbpem receives the backup completion status from nbjm, nbpem tells nbjm to delete the snapshot. nbjm starts a new instance of bpbrm on the media server, and bpbrm starts a new instance of bpfis on the client. bpfis deletes the snapshot on the client, unless the snapshot is of the Instant Recovery type, in which case it is not automatically deleted. bpfis and bpbrm report their status and exit.

VMware backup

The following shows a VMware backup process.



The basic processing steps for a VMware backup operation are the following:

VMware backup procedure

- 1 The Policy Execution Manager (`nbpem`) triggers a backup job when the policy, schedule, and virtual machine are due and the backup window is open. The `nbpem` process, the Job Manager (`nbjm`), the Resource Broker (`nbrb`), and the Enterprise Media Manager (`nbemm`) together identify the resources (media server, storage unit, etc.) for the backup operation.
- 2 For a VMware Intelligent Policy (VIP), you can throttle the VMware resources that are used in the vSphere environment. For example, you can limit the resources to four concurrent backup jobs running from a vSphere datastore. This level of control tunes the number of backups to minimally influence the user and application experience on the vSphere platform.
- 3 `nbpem` uses `nbjm` to contact the selected media server and to start the Backup and Restore Manager (`bpbrm`) on it. A snapshot job (also referred to as the parent job) goes active in the Activity Monitor.
- 4 `nbjm` starts an instance of `bpbrm` through the client service (`bpcd`) on the media server. `bpbrm` starts the Frozen Image Snapshot (`bpfis`) through the client service (`bpcd`) on the VMware backup host. `bpfis` creates a snapshot of the VM data by using vCenter or ESX host depending on the configured credential servers.

`bpfis` armed with vADP contacts the vSphere host (vCenter) or the ESX/ESXi host for which credentials are stored in the NetBackup database and initiates the snapshot for the VM. For multiple VMs, `bpbrm` starts `bpfis` for each VM so that the snapshot operations occur in parallel. As in step 2, you can control the number of concurrent snapshots for a VIP by setting VMware resource limits in NetBackup. `bpfis` contacts the vSphere host by using the standard SSL port (the default is 443).
- 5 `bpfis` contacts the Request Manager (`bprd`) to request transfer of `bpfis` state files from the VMware Backup Host to the master server.
- 6 `bprd` requests `bpcd` on the VMware Backup Host to send a list of `bpfis` state files. `bprd` copies each state file from the VMware Backup Host to the master server.
- 7 `bpfis` sends snapshot information and completion status to `bpbrm`. `bpbrm` reports the snapshot information and status to `nbjm`. `nbjm` relays the information and status to `nbpem`.
- 8 `nbpem` submits a child job for the backup to `nbjm`, with a file list derived from the snapshot information. `nbjm` starts `bpbrm` to back up the snapshot.
- 9 `bpbrm` uses `bpcd` to start `bpbkar` on the VMware Backup Host.

10 The backup and archive manager (`bpbkar`) loads the Veritas Mapping Services (VxMS) which loads the VMware Disk Development Kit (VDDK) APIs. The APIs are used for reading from the vSphere datastore. VxMS maps the stream during run-time and identifies the contents of the vmdk file. `bpbkar` uses VxMS to send the file catalog information to `bpbrrm`, which relays it to the database manager `bpdbrm` on the master server.

11 `bpbrrm` also starts the process `bptm` (parent) on the media server.

The following shows the operation of the Veritas V-Ray within VxMS:

- Veritas V-Ray within VxMS generates the catalog of all the files inside the VMDK from both Windows and Linux VMs. The operation occurs while backup data is being streamed. `bpbrrm` on the media server sends this catalog information to the master server.
- The file system inode level also identifies unused and deleted blocks. For example, if the application on VM allocates 1 TB of space for a file, of which only 100 GB is currently used, the backup stream includes only that 100 GB. Similarly, if you delete a 1 TB file that was fully allocated in the past, VxMS skips the deleted blocks (unless the blocks are now allocated for a new file) from the backup stream. This optimization not only speeds up the backup stream, but reduces needed storage even when deduplication is not enabled.
- If the source side deduplication feature is enabled, the VMware backup host does the deduplication. The NetBackup deduplication plug-in using the mapping information that VxMS generates and sees the actual files in the file system within the VMDK. This V-Ray vision is established by the NetBackup deduplication plug-in that loads a dedicated stream handler that understands the VxMS mapping info.
- Because these operations occur on the VMware backup host, the ESX resources and the VM resources are not used. This setup is true off-host backup with no burden on the production vSphere. Even the source side deduplication occurs in an off-host system.

12 If the media server is the VMware Backup Host, `bpbkar` stores the snapshot-based image block-by-block in shared memory on the media server. If the media server is backing up a separate VMware Backup Host that is not the media server, the `bptm` process on the server creates a child process of itself. The child uses socket communications to receive the snapshot-based image from the VMware Backup Host and stores the image block-by-block in shared memory.

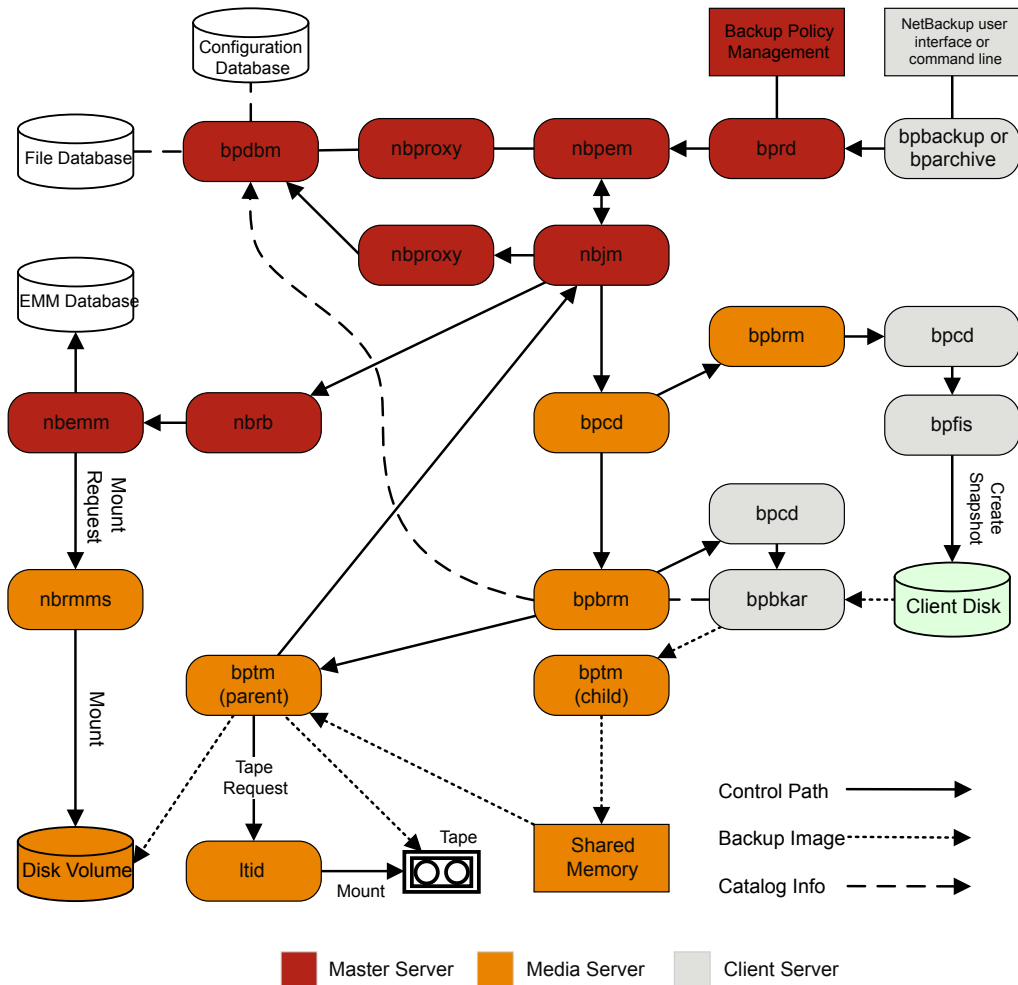
13 The original tape manager (`bptm`) process takes the backup image from shared memory and sends it to the storage device (disk or tape).

- 14 `bptm` sends backup completion status to `bpbrm`, which passes it to `nbjm` and `nbpem`.
- 15 `nbpem` tells `nbjm` to delete the snapshot. `nbjm` starts a new instance of `bpbrm` on the media server, and `bpbrm` starts a new instance of `bpfis` on the VMware Backup Host. `bpfis` deletes the snapshot on the vSphere environment. `bpfis` and `bpbrm` report their status and exit.

Snapshot backup and Windows open file backups

Figure 10-1 shows the overall snapshot backup process. PBX (not shown in the diagram) must be running for NetBackup to operate.

Figure 10-1 Snapshot backup and Windows open file backup using multiple data streams



A separate parent job creates all snapshots, then a child job backs up the snapshot.

The following sequence of operations is for snapshot creation and backup, including the Windows open file backups that employ multiple data streams:

- The NetBackup master server or primary client initiates the backup. This action causes the NetBackup request daemon `bprd` to submit a backup request to the Policy Execution Manager `nbpem`. `nbpem` processes the policy configurations.
- `nbpem` (through `nbjm`) starts a parent job to create the snapshot. This job is separate from the job that backs up the snapshot.
- `nbjm` starts an instance of `bpbrm` through `bpcd` on the media server, and `bpbrm` starts `bpfis` through `bpcd` on the client.
- `bpfis` creates a snapshot of the client's data by means of a snapshot method.
- When `bpfis` is finished, it sends snapshot information and completion status to `bpbrm` and exits. `bpbrm`, in turn, reports the snapshot information and status to `nbjm` and exits. `nbjm` relays the information and status to `nbpem`.
- `nbpem` submits a child job for the backup to `nbjm`, with a file list derived from the snapshot information. `nbjm` starts `bpbrm` to back up the snapshot.
- `bpbrm` starts `bpbkcar` on the client. `bpbkcar` sends the file catalog information to `bpbrm`, which relays it to the NetBackup file database `bpdbm` on the master server.
- `bpbrm` starts the process `bptm` (parent) on the media server.
- The next step depends on the following: Whether the media server backs up itself (`bptm` and `bpbkcar` on the same host), or the media server backs up a client on a different host. If the media server backs up itself, `bpbkcar` stores the snapshot-based image block by block in shared memory on the media server. If the media server backs up a client that resides on a different host, `bptm` on the server creates a child process of itself. The child receives the snapshot-based image from the client by means of socket communications and then stores the image block-by-block in shared memory.
- The original `bptm` process then takes the backup image from shared memory and sends it to the storage device (disk or tape).
 Information is available on how the tape request is issued.
 See "Media and device management process" in the *NetBackup Troubleshooting Guide*.
- `bptm` sends backup completion status to `bpbrm`, which passes it to `nbjm`.
- When `nbpem` receives backup completion status from `nbjm`, `nbpem` tells `nbjm` to delete the snapshot. `nbjm` starts a new instance of `bpbrm` on the media server,

and `bpbrm` starts a new instance of `bpfis` on the client. `bpfis` deletes the snapshot on the client, unless the snapshot is of the Instant Recovery type, in which case it is not automatically deleted. `bpfis` and `bpbrm` report their status and exit.

For more information, see the [NetBackup Snapshot Client Administrator's Guide](#). Note that Windows open file backups do not require Snapshot Client.

Locating logs

This chapter includes the following topics:

- [acsssi logging](#)
- [bpbackup logging](#)
- [bpbkar logging](#)
- [bpbrm logging](#)
- [bpcd logging](#)
- [bpcompatd logging](#)
- [bpdbm logging](#)
- [bpjobd logging](#)
- [bprd logging](#)
- [bprestore logging](#)
- [bptm logging](#)
- [daemon logging](#)
- [ltid logging](#)
- [nbemm logging](#)
- [nbjm logging](#)
- [nbpem logging](#)
- [nbproxy logging](#)
- [nbrb logging](#)

- [NetBackup web services logging](#)
- [NetBackup web server certificate logging](#)
- [PBX logging](#)
- [reqlib logging](#)
- [robots logging](#)
- [tar logging](#)
- [txxd and txxcd logging](#)
- [vnetd logging](#)

acsssi logging

On UNIX systems, the NetBackup ACS storage server interface (`acsssi`) communicates with the ACS library software host.

Log location	UNIX: <code>/usr/opensv/volmgr/debug/acsssi</code>
Server where it resides	media
How to access	The <code>acsssi</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

bpbackup logging

The `bpbackup` command-line executable is used to initiate user backups.

Log location	Windows: <code>install_path\NetBackup\logs\bpbackup</code> UNIX: <code>/usr/opensv/netbackup/logs/bpbackup</code>
Server where it resides	client

How to access	<p>The <code>bpbackup</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.</p> <p>See “About legacy logging” on page 36.</p>
---------------	--

See [“About backup logging”](#) on page 67.

bpbkar logging

The backup and archive manager (`bpbkar`) is used to read client data, which is sent to the media server to write to the storage media. It also collects metadata about the files that have been backed up to create the `files` file.

Log location	<p>Windows: <code>install_path\NetBackup\logs\bpbkar</code></p> <p>UNIX: <code>/usr/opensv/netbackup/logs/bpbkar</code></p>
Server where it resides	client
How to access	<p>The <code>bpbkar</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.</p> <p>See “About legacy logging” on page 36.</p>

See [“About backup logging”](#) on page 67.

bpbrm logging

The NetBackup backup and restore manager (`bpbrm`) manages the client and `bptm` process. It also uses the error status from the client and from `bptm` to determine the final status of backup and restore operations.

Log location	<p>Windows: <code>install_path\NetBackup\logs\bpbrm</code></p> <p>UNIX: <code>/usr/opensv/netbackup/logs/bpbrm</code></p>
Server where it resides	media

How to access

The `bpbxm` process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.

See [“About legacy logging”](#) on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

bpcd logging

The NetBackup client service (`bpcd`) authenticates remote hosts and launches processes on local hosts.

Log location

Windows: `install_path\NetBackup\logs\bpcd`
UNIX: `/usr/opensv/netbackup/logs/bpcd`

Server where it resides

media and client

How to access

The `bpcd` process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.

See [“About legacy logging”](#) on page 36.

See [“About backup logging”](#) on page 67.

bpcompatd logging

The NetBackup compatibility service (`bpcompatd`) creates connections between some multi-threaded processes and NetBackup legacy processes.

Log location

Windows: `install_path\NetBackup\logs\bpcompatd`
UNIX: `/usr/opensv/netbackup/logs/bpcompatd`

Server where it resides

master

How to access

The `bpcompatd` process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.

See [“About legacy logging”](#) on page 36.

See [“About backup logging”](#) on page 67.

bpdbm logging

The NetBackup Database Manager (bpdbm) manages the configuration, error, and file databases.

Log location	Windows: <i>install_path</i> \NetBackup\logs\bpdbm UNIX: /usr/openv/netbackup/logs/bpdbm
Server where it resides	master
How to access	The bpdbm process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

bpjobd logging

The bpjobd service manages the jobs database and relays job statuses to the Activity Monitor.

Log location	Windows: <i>install_path</i> \NetBackup\logs\bpjobd UNIX: /usr/openv/netbackup/logs/bpjobd
Server where it resides	master
How to access	The bpjobd process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

bprd logging

The NetBackup request daemon (`bprd`) responds to client and administrative requests for backups, restores, and archives.

Log location	Windows: <i>install_path</i> \NetBackup\logs\bprd UNIX: <i>/usr/opensv/netbackup/logs/bprd</i>
Server where it resides	master
How to access	The <code>bprd</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

bprestore logging

The `bprestore` command-line executable is used to initiate restores. It communicates with `bprd` on the master server.

Log location	Windows: <i>install_path</i> \NetBackup\logs\bprestore UNIX: <i>/usr/opensv/netbackup/logs/bprestore</i>
Server where it resides	client
How to access	The <code>bprestore</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About restore logging”](#) on page 95.

bptm logging

The NetBackup tape management process (`bptm`) manages the transfer of backup images between the client and the storage device (tape or disk).

Log location	Windows: <i>install_path</i> \NetBackup\logs\bptm UNIX: /usr/opensv/netbackup/logs/bptm
Server where it resides	media
How to access	The bptm process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

daemon logging

The daemon log includes debug information for the Volume Manager service (vmd) and its associated processes.

Log location	Windows: <i>install_path</i> \volmgr\debug\daemon UNIX: /usr/opensv/volmgr/debug/daemon
Server where it resides	master and media
How to access	The daemon log uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

ltid logging

The logical tape interface daemon (ltid), also called the NetBackup Device Manager, controls the reservation and assignment of tapes.

Log location	Windows: <i>install_path</i> \volmgr\debug\ltid UNIX: /usr/opensv/volmgr/debug/ltid
Server where it resides	media

How to access

The `ltid` process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.

See [“About legacy logging”](#) on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

nbemm logging

On the server that is defined as the master server, the NetBackup Enterprise Media Manager (`nbemm`) manages devices, media, and storage unit configuration. It supplies `nbrb` with a cache list of available resources, and manages the internal state of storage, (UP/DOWN) based on heartbeat information and disk polling.

Log location

Windows: `install_path\NetBackup\logs\nbemm`
UNIX: `/usr/opensv/logs/nbemm`

Server where it resides

master

How to access

The `nbemm` process uses the unified logging method. Use the `vxlogview` and `vxlogmgr` commands to view and manage the unified log files.

See [“About unified logging”](#) on page 13.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

nbjm logging

The NetBackup Job Manager (`nbjm`) accepts job requests from `nbpem` and from media commands, and it acquires the necessary resources for the jobs. It interacts with `bpjobd` to provide updates to the activity monitor states, starts the `bpbrrm` media manager service as needed, and updates the internal job states.

Log location

Windows: `install_path\NetBackup\logs\nbjm`
UNIX: `/usr/opensv/logs/nbjm`

Server where it resides

master

How to access	<p>The <code>nbjm</code> process uses the unified logging method. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files.</p> <p>See “About unified logging” on page 13.</p>
See “About backup logging” on page 67.	
See “About restore logging” on page 95.	

nbpem logging

The NetBackup Policy Execution Manager (`nbpem`) creates policy and client tasks and determines when jobs are run.

Log location	<p>Windows: <code>install_path\NetBackup\logs\nbpem</code></p> <p>UNIX: <code>/usr/openv/logs/nbpem</code></p>
Server where it resides	master
How to access	<p>The <code>nbpem</code> process uses the unified logging method. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files.</p> <p>See “About unified logging” on page 13.</p>

See [“About backup logging”](#) on page 67.

nbproxy logging

The proxy service `nbproxy` enables `nbpem` and `nbjm` to query master server catalogs.

Log location	<p>Windows: <code>install_path\NetBackup\logs\nbproxy</code></p> <p>UNIX: <code>/usr/openv/netbackup/logs/nbproxy</code></p>
Server where it resides	master
How to access	<p>The <code>nbproxy</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.</p> <p>See “About legacy logging” on page 36.</p>

See [“About backup logging”](#) on page 67.

nbrb logging

On the master server, the NetBackup Resource Broker (**nbrb**) locates logical and physical resources from a cached list of resources to satisfy storage units, media, and client reservations for jobs. It initiates drive queries every 10 minutes to check the state of the drives.

Log location	Windows: <i>install_path</i> \NetBackup\logs\nbrb UNIX: <i>/usr/opensv/logs/nbrb</i>
Server where it resides	master
How to access	The nbrb process uses the unified logging method. Use the vxlogview and vxlogmgr commands to view and manage the unified log files. See “About unified logging” on page 13.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

NetBackup web services logging

This topic describes the logs for the NetBackup web services.

Log location	<i>Web server logs</i> Windows: <i>install_path</i> \NetBackup\wmc\webserver\logs UNIX: <i>usr/opensv/wmc/webserver/logs</i> <i>Web applications logs</i> Windows: <i>install_path</i> \NetBackup\logs\nbwebsevice UNIX: <i>usr/opensv/logs/nbwebsevice</i>
Server where it resides	master

How to access	<p>The web services use the unified logging method for the web applications. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files.</p> <p>The NetBackup web server framework does not use the standard VxUL format. For more information on the format of these logs and how they are created, see the documentation for Apache Tomcat at http://tomcat.apache.org.</p> <p>See “About unified logging” on page 13.</p>
---------------	---

See the [NetBackup Troubleshooting Guide](#) for more information on how to access the web services logs.

NetBackup web server certificate logging

NetBackup creates the following logs when it generates and deploys the web server certificate during installation.

Log location	<p>Windows:</p> <pre>install_path\NetBackup\logs\nbatd install_path\NetBackup\logs\NBCert</pre> <p>C:\ProgramData\Symantec\NetBackup\InstallLogs\WMC_configureCerts_yyyymmdd_timestamp.txt</p> <p>UNIX:</p> <pre>usr/opensv/logs/nbatd usr/opensv/logs/nbcert usr/opensv/wmc/webserver/logs/configureCerts.log</pre>
Server where it resides	master
How to access	<p>The <code>nbcert</code> and <code>nbatd</code> logs use unified logging. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files. The <code>configureCerts.log</code> uses a simple logging style and not VxUL.</p> <p>See “About unified logging” on page 13.</p>

NetBackup creates the following logs when it renews the web server certificate.

Log location	<p>Windows:</p> <pre>install_path\NetBackup\logs\nbatd install_path\NetBackup\logs\nbservice</pre> <p>C:\ProgramData\Symantec\NetBackup\InstallLogs\WMC_configureCerts_yyyymmdd_timestamp.txt</p> <p>UNIX:</p> <pre>usr/openv/logs/nbatd install_path\NetBackup\logs\nbservice usr/openv/wmc/webserver/logs/configureCerts.log</pre>
Server where it resides	master
How to access	<p>The <code>nbweb service</code> (OID 466 and 484) and <code>nbatd</code> (OID 18) logs use unified logging. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files. The <code>configureCerts.log</code> uses a simple logging style and not VxUL.</p> <p>See “About unified logging” on page 13.</p>

See the [NetBackup Troubleshooting Guide](#) for more information on how to access the web services logs.

PBX logging

Private Branch Exchange (PBX) is the communication mechanism used by most NetBackup processes.

Log location	<p>Windows: <code>install_path\VxPBX\log</code></p> <p>UNIX: <code>/opt/VRTSspbx/log</code></p>
Server where it resides	master, media, and client
How to access	<p>The PBX process uses the unified logging method. Use the <code>vxlogview</code> and <code>vxlogmgr</code> commands to view and manage the unified log files. Note that the PBX product ID used to access the unified log files differs from the NetBackup product ID. The PBX product ID is 50936.</p> <p>See “About unified logging” on page 13.</p>

See the [NetBackup Troubleshooting Guide](#) for more information on how to access PBX logs.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

reqlib logging

The `reqlib` log includes debug information on the processes that request media management services from EMM or the Volume Manager service (`vmd`).

Log location	Windows: <code>install_path\volmgr\debug\reqlib</code> UNIX: <code>/usr/openv/volmgr/debug/reqlib</code>
Server where it resides	master and media
How to access	The <code>reqlib</code> log uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

robots logging

The `robots` log includes debug information on all robotic daemons, including the `txxd` and `txxcd` daemons.

Log location	Windows: <code>install_path\volmgr\debug\robots</code> UNIX: <code>/usr/openv/volmgr/debug/robots</code>
Server where it resides	media
How to access	The <code>robots</code> log uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“txxd and txxcd logging”](#) on page 161.

See [“About backup logging”](#) on page 67.

See [“About restore logging”](#) on page 95.

tar logging

The Tape Archive program (`tar`) writes restore data to the client disk. On Windows clients, the binary name is `tar32.exe` and on UNIX clients the binary name is `nbtar`.

Log location	Windows: <code>install_path\NetBackup\logs\tar</code> UNIX: <code>/usr/opensv/netbackup/logs/tar</code>
Server where it resides	client
How to access	The <code>tar</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process. See “About legacy logging” on page 36.

See [“About restore logging”](#) on page 95.

txxd and txxcd logging

The robotic daemon (`txxd`, where `xx` varies based on the type of robot being used) provides the interface between `ltid` and the tape library. The robotic control daemon (`txxcd`) provides the robotic control for the robot and communicates mount and unmount requests.

Log location	The <code>txxd</code> and <code>txxcd</code> processes do not have their own log files. Instead, errors are logged in the <code>robots</code> debug log and the system log. The system log is managed by <code>syslog</code> on UNIX and by the Event Viewer on Windows. See “About UNIX system logs” on page 10. See “Logging options with the Windows Event Viewer” on page 54.
How to access	The debug information is included by adding the word <code>VERBOSE</code> to the <code>vm.conf</code> file. See “How to control the amount of information written to legacy logging files” on page 46. On UNIX, debug information is also included by starting the daemon with the <code>-v</code> option (either by itself or through <code>ltid</code>).

- See [“robots logging”](#) on page 160.
- See [“About backup logging”](#) on page 67.
- See [“About restore logging”](#) on page 95.

vnetd logging

The NetBackup Legacy Network Service (`vnetd`) is a communication mechanism used to create firewall-friendly socket connections.

Log location	<p>Windows: <code>install_path\NetBackup\logs\vnetd</code></p> <p>UNIX: <code>/usr/opensv/logs/vnetd</code> or <code>/usr/opensv/netbackup/logs/vnetd</code> if the <code>vnetd</code> directory exists there. If the <code>vnetd</code> directory exists in both locations, logging occurs only in <code>/usr/opensv/netbackup/logs/vnetd</code>.</p>
Server where it resides	master, media, and client
How to access	<p>The <code>vnetd</code> process uses the legacy logging method. If legacy debug logging is not enabled on your NetBackup servers, you must create the appropriate directories for each process.</p> <p>See “About legacy logging” on page 36.</p>

- See [“About backup logging”](#) on page 67.
- See [“About restore logging”](#) on page 95.

Java-based administration console logging

This chapter includes the following topics:

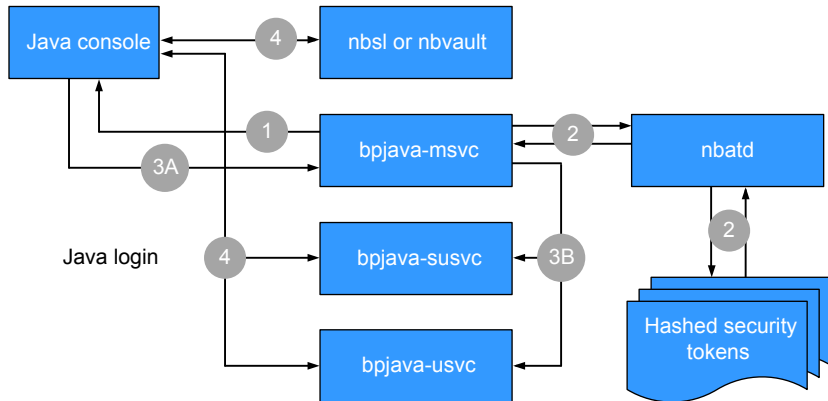
- [About the Java-based administration console logging](#)
- [Java-based administration console logging process flow](#)
- [Setting up a secure channel between the Java-based administration console and bpjava-*](#)
- [Setting up a secure channel between the Java-based administration console and either nbsl or nbvault](#)
- [Java-based administration console logging configuration on NetBackup servers and clients](#)
- [Java-based remote administration console logging on a Windows computer where NetBackup is not installed](#)
- [Configuring and gathering logs when troubleshooting Java GUI issues](#)
- [Undo logging](#)

About the Java-based administration console logging

Beginning with NetBackup 7.7, NetBackup provides only the Java-based administration console through which the administrator can manage NetBackup. The console can be run either directly on a supported Java-capable UNIX computer or on a Windows computer that has the **NetBackup Administration Console** installed.

Java-based administration console logging process flow

The Java-based administration console logging process flow is as follows:



The following steps describe the Java-based administration console login process:

1. The user initiates a login request to the Java-based administration console. The credentials are sent to `bpjava-msvc` over the Secure Sockets Layer (SSL) using the Server Security Certificate.
2. The `bpjava-msvc` process authenticates the token through `nbatd`, which reads the hashed security tokens on the server.
3. The following steps describe the process with the session certificate:
 - The `bpjava-msvc` process sends a response to the console login with a session token and a fingerprint of the session certificate.
 - The `bpjava-msvc` process initiates the appropriate `bpjava-*usvc` process and the session certificate and token are passed to one of the following processes:
 - `bpjava-susvc` for the **NetBackup Administration Console**
 - `bpjava-usvc` for the Backup, Archive, and Restore (BAR) interface
4. Various calls are made between the NetBackup Java-based administration console and `nbsl`, `bpjava-*usvc`, and `nbvault` (if configured) to populate the interface with the appropriate contents.

Setting up a secure channel between the Java-based administration console and bpjava-*

The following steps describe the process flow to set up a secure channel between the Java-based administration console and bpjava-*:

Note: The following processes are used: bpjava-msvc, which controls the login and authentication; bpjava-susvc, which is the administration console process; and bpjava-usvc, which is the client Backup, Archive, and Restore (BAR) interface.

1. The user initiates a login to the console. The credentials are sent to bpjava-msvc over the SSL (using the Server Security Certificate).
2. The bpjava-msvc process authenticates the user who uses the user credentials that were received in step 1.
3. After the user is authenticated, the bpjava-msvc process performs the following:
 - Generates the entities that are called the self-signed session certificate, the key, and the session token.
 - Launches the daemon bpjava-*usvc to gather more requests from the NetBackup Java-based administration console.
 - Passes the self-signed session certificate and the session token to bpjava-*usvc.

Note: The bpjava-*usvc process uses a session certificate as a Server Security Certificate for the SSL channel. It uses the session token to authenticate the Java-based administration console. The console does not use credentials while it connects to the bpjava-*usvc process. The Java-based administration console uses the session token for authentication.

- Sends the session token and the fingerprint of the session certificate to the Java-based administration console.
- Persists session token and user information to a secure directory (*install_path/var*; for example, *usr/openv/var*) in a file on the NetBackup host. This directory is accessible only to the root/administrator. The file name format is as follows:

```
hash(session token)_bpjava-*usvc_pid
```

Note: `msvc` saves this information so it can be used by `nbsl` or `nbvault` to authenticate the Java-based administration console.

- The `msvc` process stops the execution and exits.
4. `bpjava-*usvc` uses the session certificate to start the secure channel with the Java-based administration console. This secure channel is a one-way authenticated SSL channel. (Only the server certificate is present and there is no peer certificate. There is no certificate from the Java-based administration console side.)
 5. The Java-based administration console receives the session certificate as a part of the initial SSL handshake. It verifies the authenticity of the session certificate by using the pre-existing fingerprint of the session certificate (see step 3). The Java-based administration console calculates the fingerprint of the session certificate that was received from `bpjava-*usvc` due to the SSL handshake. It compares the new fingerprint with the fingerprint sent by `msvc`.
 6. Once the authenticity of the certificate is verified, the Java-based administration console sends the session token (received in step 3) to `bpjava-*usvc`.
 7. `bpjava-*usvc` verifies the received session token with the pre-existing one (see step 3).
 8. The success of the session token validation creates trust between `bpjava-*usvc` and the Java-based administration console.
 9. All further communication occurs between `bpjava-*usvc` and the Java-based administration console on this trusted secure channel.

Setting up a secure channel between the Java-based administration console and either nbsl or nbvault

The following steps describe the process flow to set up a secure channel between the Java-based administration console and either `nbsl` or `nbvault`:

1. Trust is already set up between the Java-based administration console and `bpjava-*`. The user information and session token already exist in a designated location with a name similar to the following:

```
hash(session token)_susvc_pid
```

See [“Setting up a secure channel between the Java-based administration console and bpjava-*”](#) on page 165.

2. The Java-based administration console sends a request to `nbsl/nbvault` for a secure connection.
3. `nbsl/nbvault` accepts the request and initiates a secure channel using the security certificate on the host. These daemons run with root/administrator privileges and can access the security certificate.
4. This secure channel is a one-way authenticated SSL channel where only the server certificate is present and there is no peer certificate. There is no certificate from the Java-based administration console side.
5. The trust options for the security certificate are as follows:
 - The Java-based administration console accepts the security certificate (or gives approval for this secure channel) if it trusts the NetBackup Certificate Authority (CA) who signed the security certificate.
 - If the Java-based administration console does not trust the CA who signed the security certificate, it displays a pop-up dialog box. This dialog box asks if the user trusts the CA who has signed the certificate (This is a one-time activity. After the user gives consent to trust the CA, the dialog box does not display again.).
6. The Java-based administration console sends a session token to `nbsl/nbvault`. See [“Setting up a secure channel between the Java-based administration console and bpjava-*”](#) on page 165.
7. `nbsl/nbvault` verifies this session token by performing the following procedure:
 - Generates a hash of the session token that was received
 - Searches for the file with the name that starts with this hash at the designated location
 - If the file is found, it extracts the PID from it (see step 1)
 - Checks to see if the PID is valid
8. The success of the verification creates a trust between `nbsl/nbvault` and the Java-based administration console.
9. All further communication occurs between `nbsl/nbvault` and the Java-based administration console on this trusted secure channel.

Java-based administration console logging configuration on NetBackup servers and clients

Java logging is automatically set up on systems on which the NetBackup client or server software is installed. The Java logs are located in the following pre-existing log directories:

- UNIX: `/usr/opensv/netbackup/logs/user_ops/nbjlogs`
- Windows: `install_directory\netbackup\logs\user_ops\nbjlogs`

Java-based remote administration console logging on a Windows computer where NetBackup is not installed

If the **NetBackup Remote Administration Console Installation (x64 only)** option is used when installing, NetBackup is not installed on the host and the Java activity is not logged by default.

You must make changes to the `setconf.bat` file to ensure that Veritas Technical Support can successfully log the Java operations; that is, `%nbjLogFileName%`.

The following changes to the `setconf.bat` file are required as there is no active path set against `NB_INSTALL_PATH` on the target host. The procedure is as follows:

1. Manually create the following directory structure in its entirety:

```
C:\Program Files\Veritas\NetBackup\logs\user_ops\nbjlogs
```

2. Edit the following file:

```
install_path\Veritas\Java\setconf.bat
```

3. Locate the following line (line 12 down from the top of the file) and remove the remark (REM) to activate it, as shown in the following example:

- Change line from: `REM SET NB_INSTALL_PATH=C:\\Program Files\\Veritas\\NetBackup`
- To: `SET NB_INSTALL_PATH=C:\\Program Files\\Veritas\\NetBackup`

4. Save the change to the `setconf.bat` file.
5. The next launch of the Java console produces a log within the following directory at the default verbosity of 3:

C:\Program Files\Veritas\NetBackup\logs\user_ops\nbjlogs

Configuring and gathering logs when troubleshooting Java GUI issues

Once installed, the log levels for the Java-based administration console are configured to gather a detailed set of logs.

The NetBackup Java GUI uses the `Debug.properties` file to determine which logging level to use.

On UNIX systems, the file is: `/usr/opensv/java/Debug.properties`

On Windows systems, the file is: `install_dir\VERITAS\Java\Debug.properties`

Specifically, the following settings are tuned to enable additional logging:

```
printcmds=true
debugMask=0x00040000
```

1. Gather the following Java console logs from the following pre-existing log directories on the system from which the GUI was started:
 - **UNIX:** `/usr/opensv/netbackup/logs/user_ops/nbjlogs`
 - **Windows:** `install_directory\netbackup\logs\user_ops\nbjlogs`
2. On the master server, log in through the Java-based administration console to create the `admin`, `bpjava-msvc`, `bpjava-susvc`, and `bpjava-usvc` log directories and enable VERBOSE 5 logging. You do not have to restart the NetBackup daemons for the logging level changes to take effect.

For UNIX systems, create the following directories:

- `/usr/opensv/netbackup/logs/admin`
- `/usr/opensv/netbackup/logs/bpjava-msvc`
- `/usr/opensv/netbackup/logs/bpjava-susvc`
- `/usr/opensv/netbackup/logs/bpjava-usvc`

3. Edit the `/usr/opensv/netbackup/bp.conf` file by adding the following lines:

```
ADMIN_VERBOSE = 5
BPJAVA-MSVC_VERBOSE = 5
BPJAVA-SUSVC_VERBOSE = 5
BPJAVA-USVC_VERBOSE = 5
```

4. For Windows systems, create the following directories:

- `install_dir\VERITAS\NetBackup\logs\admin`
- `install_dir\VERITAS\NetBackup\logs\bpjava-msvc`
- `install_dir\VERITAS\NetBackup\logs\bpjava-susvc`
- `install_dir\VERITAS\NetBackup\logs\bpjava-usvc`

5. Update the Windows registry at **hkey_local_machine > software > veritas > netbackup > current version > config** and add the following entries of type DWORD:

```
ADMIN_VERBOSE = 5
BPJAVA-MSVC_VERBOSE = 5
BPJAVA-SUSVC_VERBOSE = 5
BPJAVA-USVC_VERBOSE = 5
```

6. Run the following commands to set up detailed `nbatd` (OID 18) and `nbsl` (OID 132) logging. OID 137 (NetBackup libraries) and OID 156 (CORBA/ACE) write to the caller that requires access to either the libraries or CORBA/ACE, as follows:

```
vxlogcfg -a -p NB -o 18 -s DebugLevel=6
vxlogcfg -a -p NB -o 132 -s DebugLevel=6
vxlogcfg -a -p NB -o 137 -s DebugLevel=6
vxlogcfg -a -p NB -o 156 -s DebugLevel=6
```

7. Gather the `nbatd` and `nbsl` logs located in the following directory paths:

For UNIX:

- `/usr/opensv/logs/nbsl`
- `/usr/opensv/logs/nbatd`

For Windows:

- `install_dir\VERITAS\NetBackup\logs\nbsl`
- `install_dir\VERITAS\NetBackup\logs\nbatd`

8. Finally, gather the PBX logs, as follows:

- For UNIX: `/opt/VRTSspbx/log` (gather any logs that cover the current date/time)
- For Windows: `install_dir\VERITAS\pbx\log`

Undo logging

Ensure that you undo the logging after you have gathered the logs that relate to your troubleshooting issue.

To remove the log configuration settings, use the following commands:

```
vxlogcfg -r -p NB -o 18 -s DebugLevel=6
vxlogcfg -r -p NB -o 132 -s DebugLevel=6
vxlogcfg -r -p NB -o 137 -s DebugLevel=6
vxlogcfg -r -p NB -o 156 -s DebugLevel=6
```

On the master server, comment out the following Java `VERBOSE` entries in the `bp.conf` file (UNIX) or in the registry (Windows):

- `ADMIN_VERBOSE`
- `BPJAVA-MSVC_VERBOSE`
- `BPJAVA-SUSVC_VERBOSE`
- `BPJAVA-USVC_VERBOSE`

Index

A

- acssel, description 81
- acsssi logging 149
- acsssi, description 81
- admin log 43
- administration interface
 - activity logging 59
 - errors 57
- Application Event log 54
- application server status codes (Java interface) 58
- ascd, description 81
- Auto Image Replication (A.I.R.) 132
 - import process flow 135
 - logging 135
 - process flow logging 134
- avrd, description 82

B

- backup
 - and archive processes 66
 - and restore startup process 65
 - logging 62, 67
 - NetBackup catalogs 103
 - procedure, basic 63
 - process 62
 - deduplication 118
 - multiplexing 67
 - snapshot overview 144
 - synthetic processes 107
 - UNIX clients 67
- Backup, Archive, and Restore (BAR) interface 164
- backup_tape log 39
- barcode operations 77
- basic backup procedure 63
- bin
 - media and device management 80
- bjava-usvc process 164
- bp
 - UNIX client log 38
- bp.conf
 - file 67
- bparchive
 - log 38, 40
- bpbackup
 - log 38, 41
- bpbackup logging 149
- BPBACKUP_POLICY 67
- BPBACKUP_SCHED 67
- bpbkar
 - log 38, 41
 - logging 150
- bpbrm 146
 - log 43
 - logging 150
- bpcd
 - server log 43
 - UNIX client log 38, 41
- bpcd logging 151
- bpcompatd logging 151
- bpdjobs log 43
- bpdbm
 - log 44
 - logging 152
- bpdm log 43
- bpfis 146
- bphdb
 - log 39
- BPINETD 94
- bpinetd log 40
- bpinetd.log 40
- bpjava-*usvc process 164
- bpjava-msvc log 44, 60
- bpjava-msvc process 164
- bpjava-susvc process 164
- bpjava-usvc log 60
- bpjobd logging 152
- bplist
 - log 39, 41
- bpmount
 - log 39, 41
- bporaexp log 39
- bporaexp64 log 39
- bporaimp log 39

- bpوراimp64 log 39
- bprd log 44
- bprd logging 153
- bprestore
 - log 39, 41
 - logging 153
- bpsetconfig 50
- bpsynth 107
- bptm log 44
- bptm logging 153

C

- catalog backup 103
- change
 - logging level 52
- client
 - NetBackup
 - debug logs. *See* UNIX clients. *See* Windows and NetWare clients
- client deduplication
 - logging 121
- configuration
 - deduplication for MSDP 119
 - OpenStorage Technology (OST) 126
- configuration and management
 - OpenStorage Technology (OST) 129
- configuration logs
 - deduplication 121
- Configuring and gathering logs
 - when troubleshooting Java GUI issues 169

D

- daemon logging 154
- daemons
 - robotic 71
 - robotic control 71
- database backup (see catalog backup) 103
- DAYS_TO_KEEP_LOGS vm.conf setting 49
- debug level 53
- debug logs 59
 - NetBackup 80
 - vmd 45, 80
- debug.properties file 60
- deduplication
 - configuration logs 121
- deduplication backup process to MSDP 118
- deduplication configuration for MSDP 119

- directory structure
 - media and device management 79
- disk
 - monitoring logging 124
 - restore from 91
- disk space
 - for logs and temporary files 59
 - for logs files 33
- drive_mount_notify script 74
- drive_unmount_notify script 74
- driver directory 80

E

- EMM server 66
- enable debug logging 45
- Enterprise Media Manager (EMM) 66
- Error message types 57
- Event Viewer logging options 54
- eventlog 55
 - file entries 56
- exception errors in Java admin interface 57
- extra disk space for logs and temporary files 59

F

- fibre channel 98
- files
 - restore process 92
- FSM 98
- FT Service Manager 98
- functional overview
 - Media and Device Management
 - device management 73
 - volume management 73
 - media and device management
 - directories and files 79
 - NetBackup
 - restores 92
 - startup 65

G

- Global logging level 46, 52
- Global logging levels 50
- goodies directory 80

H

- help files
 - media and device management 80

- hostID
 - unified logging 16
- Hot catalog backup process 104
- Hot catalog restore 105

J

- Java interface
 - troubleshooting background 57
- Java-based administration console logging 163
 - configuration on NetBackup servers and clients 168
 - process flow 164
- Java-based remote administration console logging
 - on a Windows computer where NetBackup is not installed 168
- job ID search in unified logs 31

K

- Keep logs for setting 25
- keywords
 - logging 124

L

- legacy logging 37
 - client logs 38
 - configuring rotation 49
 - controlling size of 47
 - directories 38
 - locations 37
 - PC clients 40
 - rotation of 48
- levels for logging 50
- limiting the size of unified and legacy logs 12
- log level
 - UNIX clients 52
 - Windows clients 53
- log retention options 10
- logging
 - acsssi 149
 - backup 67
 - bpbackup 149
 - bpbkar 150
 - bpbrm 150
 - bpcd 151
 - bpcompatd 151
 - bpdbm 152
 - bpjobd 152
 - bprd 153

logging (*continued*)

- bprestore 153
- bptm 153
- changing location of 23
- client deduplication 121
- daemon 154
- disk monitoring 124
- Java-based administration console 163
- keywords 124
- legacy 37
- levels 50
- ltid 154
- media server deduplication/pdplugin 123
- nbemm 155
- nbjm 155
- nbpem 156
- nbproxy 156
- nbrb 157
- nbtar 161
- PBX 159
- reqlib 160
- restore 95
- robots 160
- setting level on PC clients 53
- synthetic backup 110
- tar 161
- txxd and txxcd 161
- vnetd 162

logs

- configuration
 - deduplication 121
- debug
 - enabling detailed 59
- file retention 25
- overview 8
- PC client activity
 - bparchive 40
 - bpbackup 41
 - bpbkar 41
 - bpcd 41
 - bpinstd 40
 - bpplst 41
 - bpmount 41
 - bprestore 41
 - tar 41
 - user_ops 42
- reports
 - NetBackup 9

- logs (*continued*)
 - server activity
 - acssi 45
 - admin 43
 - bpbrm 43
 - bpcd 43
 - bpdbjobs 43
 - bpdbm 44
 - bpdm 43
 - bpjava-susvc 44
 - bprd 44
 - bpsynth 44
 - bptm 44
 - daemon 45
 - ltid 45
 - nbatd 17, 44
 - nbazd 44
 - nbjm 18
 - nbpem 18
 - reqlib 45
 - robots 45
 - syslogs 44
 - tpcommand 46
 - setting log size retention 12
 - setting retention period 47
 - system 10
 - UNIX client activity
 - backup_tape 39
 - bp 38
 - bparchive 38
 - bpbackup 38
 - bpbkar 38
 - bpcd 38
 - bphdb 39
 - bpjava-msvc 44
 - bplist 39
 - bpmount 39
 - bprestore 39
 - nbtar 40
 - user_ops 40
 - Windows Event Viewer logging option 54
 - ltid 47
 - ltid logging 154
 - ltid, description 82

M

- MAX_LOGFILE_SIZE 50
- MAX_NUM_LOGFILES 50
- MaxLogFileSizeKB 33–34, 36

- Media and device management
 - process 73
- media and device management 71
- media and device management components 79
- Media Server Deduplication Pool (MSDP) 118
- media server deduplication/pdplugin logging 123
- Messages, error 57
- misc file 80
- mklogdir.bat 37
- moving log locations 23
- MSDP
 - deduplication configuration 119
- multiplexed backups 67

N

- nbatd log 44
- nbazd log 44
- nbemm 66
- nbemm logging 155
- nbftclnt 98, 100, 102
- nbftsrvr 98, 100, 102
- nbjm 18, 66, 107, 146
- nbjm logging 155
- nbpem 18, 66–67, 107, 146
- nbpem logging 156
- nbproxy logging 156
- nbrb 66
- nbrb logging 157
- nbsl 164
- nbstar logging 161
- nbvault 164
- NBWIN 94
- NDMP backup logging 112
- NDMP backup procedure 114
- NDMP restore logging 115
- NDMP restore procedure 116
- NetBackup
 - process descriptions 65
 - product ID 16
- NetBackup Administration Console
 - debug logging 59
 - errors 57
- NetBackup Status Collection daemon. *See* vmscd
- network daemon (vnetd) 45
- NumberOfFiles 33
- NumberOfLogFiles 36

O

- OpenStorage Technology (OST)
 - configuration 126
 - configuration and management 129
- operating system errors 58
- originator IDs
 - list of 17
- originatorID
 - unified logging 16

P

- PBX logging 159
- Private Branch Exchange (PBX) 123
- process descriptions
 - NetBackup 65
- product ID for NetBackup 16
- productID
 - unified logging 16

Q

- query string 27

R

- raw partitions
 - restore process 92
- reports
 - NetBackup 9
- reqlib logging 160
- restore logging 95
- restore logs
 - send to Veritas Technical Support 96
- restore procedure from disk 91
- restore procedure from tape 90
- restore process 88
 - UNIX client 92
 - Windows client 94
- retention
 - of logs 25
- retention limits for logs
 - setting 54
- robot drive selection 73
- robotic
 - control daemons 72
 - daemons 72
- robots logging 160
- RolloverMode 36
- rotation
 - legacy logging 48

- rotation (*continued*)

- of logs 24
 - unified logging 16

S

- SAN client backup procedure 99
- SAN client backup process flow 99
- SAN Client Fiber Transport backup 98
- SAN Client Fiber Transport restore 101
- Secure Sockets Layer (SSL) 164
- sending backup logs 69
- server
 - NetBackupdebug logs 37
- Server Security Certificate 164
- session certificate 164
- set retention limits for logs 54
- setting up a secure channel
 - between the Java console and bpjava-* 165
 - between the Java console and either nbsl or nbvault 166
- Shared Storage Option management process 75
- snapshot
 - backup process overview 146
- Snapshot backup 144
- Snapshot Client backup 138
- Snapshot Client backup procedure 140
- SSO. *See* Shared Storage Option
- startup
 - NetBackup 65
- startup process 71
 - media and device management 71
- Status Collection Daemon 38
- stderr 57
- stdout 57
- storage lifecycle policy (SLP) 132
 - configuration and management 136
 - duplication process flow 133
 - logging 135
- synthetic backup
 - logs 110
- synthetic backups 107
- syslogd 10
- system logs 10

T

- tape
 - restore from 90

- tar
 - log files 15
 - log on Windows clients 41
 - logging 161
- TAR32 94
- tl4d, description 83
- tl8cd, description 84
- tl8d, description 83
- tl8cd, description 85
- tl8d, description 84
- tlhcd, description 85
- tlhd, description 85
- tlmd, description 86
- tpautoconf 46
- tpconfig 46
- Troubleshooting error messages in the NetBackup
 - Administration Console for UNIX 57
- Troubleshooting Java GUI issues 169
- try file 111
- tshd, overview 86
- txxd and txxcd logging 161

U

- undo logging 171
- unified logging 13
 - changing location of 23
 - client logs 38
 - configuring settings 34
 - controlling disk space usage 33
 - controlling number of log files 33
 - controlling size of 34
 - deleting logs 32
 - file name format 16
 - file rotation 24
 - format of files 26
 - listing settings 36
 - location 13
 - message types 15
 - NetBackup product ID 16
 - processes using 17
 - retention 25
 - setting level on PC clients 53
 - settings levels 50
 - submitting to Technical Support 14
 - tar log files 15
- UNIX client restore 92
- UNIX clients
 - processes that use legacy logging 38
- UNIX system logs 10

- upload directory 15
- user-directed backups 67
- user_ops log 40, 42, 44

V

- VERBOSE 47
- verbose flag 47
- VERBOSE level 52
- Veritas Technical Support
 - restore logs 96
 - send backup logs to 69
- Veritas V-Ray 143
- vm.conf 47
- vm.conf file 81
- vmd 45
 - debug logging 45
 - overview 86
- vmgcd 38
 - logging 46
- vmgcd directory 38
- vmgcd, overview 87
- VMware backup 141
- VMware backup procedure 142
- vnetd log 45
- vnetd logging 162
- vSphere 142
- vxlogcfg 23
- vxlogcfg command 34, 36, 52
- vxlogmgr command 31, 33
- vxlogview command 26
 - query string overview 27
 - with job ID option 31

W

- Windows client
 - restore 94
- Windows Event Viewer 54
- Windows open file backups 144, 146

X

- XML 39