Symantec NetBackup™ Logging Reference Guide

Documentation version: 7.7

Legal Notice

Copyright © 2015 Symantec Corporation. All rights reserved.

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

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 Symantec Corporation 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. SYMANTEC CORPORATION 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 Symantec 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.
Technical Support

Technical Support maintains support centers globally. Technical Support’s primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within the company to answer your questions in a timely fashion.

Our support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about our support offerings, you can visit our website at the following URL:

support.symantec.com

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

support.symantec.com

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Technical Support
  - Recent software configuration changes and network changes

**Licensing and registration**

If your product requires registration or a license key, access our technical support Web page at the following URL:

support.symantec.com

**Customer service**

Customer service information is available at the following URL:

support.symantec.com

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Advice about technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs, DVDs, or manuals
Support agreement resources

If you want to contact us regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan  customercare_apj@symantec.com
Europe, Middle-East, and Africa  semea@symantec.com
North America and Latin America  supportsolutions@symantec.com
Contents

Technical Support ................................................................. 4

Chapter 1 Using logs .............................................................. 11
  About logs ........................................................................... 11
  About UNIX system logs ..................................................... 13
  About log retention in NetBackup .......................................... 13
  About limiting the size of unified and legacy logs .................... 15
  About unified logging ......................................................... 15
    Gathering unified logs for NetBackup .................................. 17
    Types of unified logging messages ..................................... 18
    File name format for unified logging .................................. 19
    Originator IDs for the entities that use unified logging .......... 20
    About changing the location of unified log files .................. 26
    About rolling over unified log files .................................... 26
    About recycling unified log files ....................................... 28
    About using the vxlogview command to view unified logs ....... 29
    About query strings used with the vxlogview command ........... 30
    Examples of using vxlogview to view unified logs ............... 33
    Examples of using vxlogmgr to manage unified logs ............. 34
    Examples of using vxlogecfg to configure unified logs .......... 37
  About legacy logging .......................................................... 39
    UNIX client processes that use legacy logging ..................... 41
    PC client processes that use legacy logging ....................... 43
    File name format for legacy logging .................................... 45
    Directory names for legacy debug logs for servers ............... 46
    Directory names for legacy debug logs for media and device
      management ...................................................................... 48
    How to control the amount of information written to legacy logging
      files .............................................................................. 49
    About limiting the size and the retention of legacy logs .......... 50
    Configuring legacy log rotation .......................................... 52
    Creating legacy log directories to accompany problem reports for
      synthetic backup ............................................................. 53
  About global logging levels .................................................. 54
  Changing the logging level .................................................. 56
Chapter 6  Storage logging ................................................................. 116
   NDMP backup logging .............................................................. 116
   NDMP restore logging ............................................................ 118

Chapter 7  NetBackup Deduplication logging .................................... 121
   Deduplication backup process to the Media Server Deduplication Pool (MSDP) ................................................................. 121
   Client deduplication logging .................................................... 124
   Deduplication configuration logs .............................................. 124
   Media server deduplication/pdplugin logging .......................... 126
   Disk monitoring logging .......................................................... 127
   Logging keywords ................................................................. 127

Chapter 8  Snapshot technologies .................................................. 129
   Snapshot Client backup ......................................................... 129
   VMware backup ..................................................................... 132
   Snapshot backup and Windows open file backups ...................... 135

Chapter 9  Locating logs ............................................................... 138
   acscci logging ................................................................. 139
   bpbackup logging ............................................................. 139
   bpkar logging ................................................................. 140
   bpbm logging ................................................................... 140
   bpcd logging ................................................................. 141
   bpcplg logging .................................................................. 141
   bcompd logging .................................................................. 141
   bpdbm logging ............................................................... 141
   bjobd logging ................................................................. 142
   bprd logging ................................................................. 142
   bprestore logging ............................................................. 143
   bptm logging ................................................................. 143
   daemon logging .............................................................. 144
   ltid logging ................................................................. 144
   nbemm logging ............................................................. 145
   nbjm logging ................................................................. 145
   nbpem logging ............................................................... 146
   nbproxy logging ............................................................. 146
   nbrii logging ................................................................. 146
   PBX logging ................................................................. 147
   reqlib logging ............................................................... 147
   robots logging .............................................................. 148
   tar logging ................................................................. 148
txxd and txgcd logging ................................................................. 149
vnetd logging ............................................................................. 149

Index .............................................................................................. 151
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
- Logs to accompany problem reports for synthetic backups
- 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.
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 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
### Log retention options in NetBackup

<table>
<thead>
<tr>
<th>Log retention option</th>
<th>Use this option...</th>
<th>Reference link</th>
</tr>
</thead>
</table>
| Keep logs up to GB                   | To limit the size of unified and legacy logs.  
When the log size across NetBackup processes grows up to this configuration value, the older logs are deleted.  
This option is available on the **NetBackup Administration Console > NetBackup Management > Host Properties > Logging** dialog box. | See “About limiting the size of unified and legacy logs” on page 15.                                                      |
| NumberOfLogFiles                     | To limit the number of unified log files that you want to retain for a NetBackup process.  
When the number of log files exceeds this configuration value, the oldest log files become eligible for deletion during log cleanup.  
This option can be set using a command-line interface. | See “About recycling unified log files” on page 28.                                                                        |
| MaxLogFileSizeKB and other vxlogcfg options | To prevent unified log files from becoming too large.  
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.  
These options can be set using a command-line interface. | See “About rolling over unified log files” on page 26.                                                                        |
| 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 50.                                                   |
Table 1-1  Log retention options in NetBackup (continued)

<table>
<thead>
<tr>
<th>Log retention option</th>
<th>Use this option...</th>
<th>Reference link</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_LOGFILE_SIZE and MAX_NUM_LOGFILES</td>
<td>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.</td>
<td>See “Configuring legacy log rotation” on page 52.</td>
</tr>
</tbody>
</table>

**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.

**Note:** In the case of the NetBackup server, the recommended value for the **Keep logs up to GB** option is 25 GB or above. For NetBackup clients, the recommended value is 5 GB or greater.

See “About log retention in NetBackup” on page 13.

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/openv/netbackup/logs/nbutils`

---

### About unified logging

Unified logging and legacy logging are the two forms of debug logging used in NetBackup. Unified logging creates log file names and messages in a standardized format. All NetBackup processes use either unified logging or legacy logging.
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.

See “About legacy logging” on page 39.

Server processes and client processes use unified logging.

See “Originator IDs for the entities that use unified logging” on page 20.

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/openv/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:

- `vxlogcfg`: Modifies the unified logging configuration settings.
  - See “Examples of using vxlogcfg to configure unified logs” on page 37.
- `vxlogmgr`: Manages the log files that the products that support unified logging generate.
  - See “Examples of using vxlogmgr to manage unified logs” on page 34.
- `vxlogview`: Displays the logs that unified logging generates.
  - See “Examples of using vxlogview to view unified logs” on page 33.

See the **NetBackup Commands Reference Guide** for a complete description about these commands. The guide is available through the following URL:

http://www.symantec.com/docs/DOC5332

These commands are located in the following directory:

- **Windows**: `install_path\NetBackup\bin`
- **UNIX**: `/usr/openv/netbackup/bin`
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/openv/logs/bmrsetup/51216-157-2202872032-050125-0000000.log
   /usr/openv/logs/nbemm/51216-111-2202872032-050125-0000000.log
   /usr/openv/logs/nbrb/51216-118-2202872032-050125-0000000.log
   /usr/openv/logs/nbjm/51216-117-2202872032-050125-0000000.log
   /usr/openv/logs/nbpem/51216-116-2202872032-050125-0000000.log
   /usr/openv/logs/nbsl/51216-132-2202872032-050125-0000000.log
   Total 6 file(s)

   Copying
   /usr/openv/logs/bmrsetup/51216-157-2202872032-050125-0000000.log ... 
   Copying
   /usr/openv/logs/nbemm/51216-111-2202872032-050125-0000000.log ... 
   Copying
   /usr/openv/logs/nbrb/51216-118-2202872032-050125-0000000.log ... 
   Copying
   /usr/openv/logs/nbjm/51216-117-2202872032-050125-0000000.log ... 
   Copying
   /usr/openv/logs/nbpem/51216-116-2202872032-050125-0000000.log ... 
   Copying
   /usr/openv/logs/nbsl/51216-132-2202872032-050125-0000000.log ...
3 Change to the /upload directory and list its contents.

```sh
# 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.

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

---

**Types of unified logging messages**

The following message types can appear in unified logging files:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **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:  

```
05/02/10 11:02:01.717 [Warning] V-116-18 failed to connect to nbjm, will retry
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **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:  

```
05/05/09 14:14:30.347 V-116-71 [JobScheduler::doCatIncr] no configured session based incremental catalog schedules
```
Debug log messages are intended primarily for Symantec 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:

```
10/29/09 13:11:28.065 [taolog] TAO (12066|1) -
Transport_Cache_Manager::bind_i, 0xffbfc194 ->
0x7179d0 Transport[12]
```

---

## 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/openv/logs/nb pem/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

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>51216</td>
<td>Product ID</td>
<td>Identifies the product. The NetBackup product ID is 51216. The product ID is also known as the entity ID.</td>
</tr>
<tr>
<td>116</td>
<td>Originator ID</td>
<td>Identifies the log writing entity, such as a process, service, script, or other software. The number 116 is the originator ID of the nb pem process (the NetBackup policy execution manager).</td>
</tr>
<tr>
<td>2201360136</td>
<td>Host ID</td>
<td>Identifies the host that created the log file. Unless the file was moved, this ID is the host where the log resides.</td>
</tr>
<tr>
<td>041029</td>
<td>Date</td>
<td>Shows the date when the log was written in YYMMDD format.</td>
</tr>
<tr>
<td>00000000000</td>
<td>Rotation</td>
<td>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 26.</td>
</tr>
</tbody>
</table>
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 20..

Windows  
\textit{install\_path\NetBackup\logs}

UNIX  
\textit{/usr/openv/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 may 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>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>extit{nbatd}</td>
<td>The authentication service (\textit{nbatd}) is a service (daemon) that verifies the user identity and issues credentials. These credentials are used for Secure Sockets Layer (SSL) communication. The (\textit{nbatd}) directory is created under the \textit{usr/netbackup/sec/at/bin} directory (UNIX) or the \textit{install_path\NetBackup\sec\at\bin} directory (Windows).</td>
</tr>
<tr>
<td>103</td>
<td>extit{pbx_exchange}</td>
<td>The Private Branch Exchange (PBX) service provides single-port access to clients outside the firewall that connect to Symantec product services. Service name: VRTSPbx. It writes logs to \textit{/opt/VRTSPbx/log} (UNIX) or \textit{install_path\VxPBX\log} (Windows). The PBX product ID is 50936.</td>
</tr>
<tr>
<td>111</td>
<td>extit{nbemm}</td>
<td>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.</td>
</tr>
<tr>
<td>Originator ID</td>
<td>Entity</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>116</td>
<td>nbpem</td>
<td>The NetBackup Policy Execution Manager (nbpem) creates policy and client tasks and determines when jobs are due to run. It runs only on the master server.</td>
</tr>
<tr>
<td>117</td>
<td>nbjm</td>
<td>The NetBackup Job Manager (nbjm) accepts the jobs that the Policy Execution Manager submits and acquires the necessary resources. It runs only on the master server.</td>
</tr>
<tr>
<td>118</td>
<td>nbrb</td>
<td>The NetBackup Resource Broker (nbrb) maintains a cache list of available resources and uses that list to locate the physical and logical resources that are required for a backup or tape restore. It initiates a SQL call to nbemm to update the database, and then passes the allocation information to nbjm. It runs only on the master server.</td>
</tr>
<tr>
<td>119</td>
<td>bmrd</td>
<td>The NetBackup Bare Metal Restore (BMR) master server daemon.</td>
</tr>
<tr>
<td>121</td>
<td>bmrsavecfg</td>
<td>The BMR Save Configuration is a data collection utility that runs on the NetBackup client, not the server.</td>
</tr>
<tr>
<td>122</td>
<td>bmrc</td>
<td>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.</td>
</tr>
<tr>
<td>123</td>
<td>bmrs</td>
<td>The BMR Server Utility.</td>
</tr>
<tr>
<td>124</td>
<td>bmrcreatefloppy</td>
<td>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.</td>
</tr>
<tr>
<td>125</td>
<td>bmrssrt</td>
<td>The BMR Create SRT utility creates a shared resource tree. It runs on the BMR boot server.</td>
</tr>
<tr>
<td>126</td>
<td>bmrprep</td>
<td>The BMR Prepare to Restore utility prepares the BMR servers for a client restoration.</td>
</tr>
<tr>
<td>127</td>
<td>bmrsetup</td>
<td>The BMR Setup Commands utility sets up BMR installation, configuration, and upgrade processes.</td>
</tr>
<tr>
<td>128</td>
<td>bmrcommon</td>
<td>The BMR Libraries and Common Code catalog provides log messages to the BMR libraries.</td>
</tr>
<tr>
<td>129</td>
<td>bmrconfig</td>
<td>The BMR Edit Configuration utility modifies the client configuration.</td>
</tr>
<tr>
<td>130</td>
<td>bmrcreatepkg</td>
<td>The BMR Create Package utility adds Windows drivers, service packs, and hot fixes to the BMR master server for restore operations.</td>
</tr>
</tbody>
</table>
### Table 1-3
Originator IDs for the server entities that use unified logging

<table>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>131</td>
<td>bmrrst</td>
<td>The BMR Restore utility restores Windows BMR clients. It runs on the restoring client for Windows systems only.</td>
</tr>
<tr>
<td>132</td>
<td>nbsl</td>
<td>The NetBackup Service Layer facilitates the communication between the NetBackup graphical user interface and NetBackup logic. nbsl is required to run Symantec NetBackup OpsCenter, an application that manages and monitors multiple NetBackup environments. This process runs only on the master server.</td>
</tr>
<tr>
<td>134</td>
<td>ndmpagent</td>
<td>The NDMP agent daemon manages NDMP backups and restores. It runs on the media server.</td>
</tr>
<tr>
<td>137</td>
<td>libraries</td>
<td>The libraries control the logging level in the NetBackup libraries. The application and diagnostic messages are for customer use; debug messages are intended for Symantec engineering.</td>
</tr>
<tr>
<td>140</td>
<td>mmui</td>
<td>The media server user interface is used for the Enterprise Media Manager (EMM).</td>
</tr>
<tr>
<td>142</td>
<td>bmrepadm</td>
<td>The BMR External Procedure process manages the BMR external procedures that are used during a restore operation.</td>
</tr>
<tr>
<td>143</td>
<td>mds</td>
<td>The EMM Media and Device Selection process manages the media selection component and device selection component of the Enterprise Media Manager (EMM).</td>
</tr>
<tr>
<td>144</td>
<td>da</td>
<td>The EMM Device Allocator is used for shared drives.</td>
</tr>
<tr>
<td>146</td>
<td>NOMTRS</td>
<td>The Symantec OpsCenter reporting service is part of Symantec OpsCenter.</td>
</tr>
<tr>
<td>147</td>
<td>NOMClient</td>
<td>The Symantec OpsCenter Client is part of Symantec OpsCenter.</td>
</tr>
<tr>
<td>148</td>
<td>NOMServer</td>
<td>The Symantec OpsCenter Server is part of Symantec OpsCenter</td>
</tr>
<tr>
<td>151</td>
<td>ndmp</td>
<td>The NDMP message log (ndmp) handles NDMP protocol messages, avrd, and robotic processes.</td>
</tr>
<tr>
<td>154</td>
<td>bmrovradm</td>
<td>The BMR Override Table Admin Utility manages the custom override functions for Bare Metal Restore.</td>
</tr>
</tbody>
</table>
Table 1-3  Originator IDs for the server entities that use unified logging

<table>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>ace</td>
<td>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 Symantec engineering. If Symantec Technical Support instructs you to increase the logging level, increase the level for originator ID 137 to 4 or higher. <strong>Warning:</strong> A debug logging level greater than 0 generates large amounts of data.</td>
</tr>
<tr>
<td>158</td>
<td>ncfrai</td>
<td>Remote access interface for NetBackup clients.</td>
</tr>
<tr>
<td>159</td>
<td>ncftfi</td>
<td>Transmater for NetBackup clients.</td>
</tr>
<tr>
<td>163</td>
<td>nbsvcmon</td>
<td>The NetBackup Service Monitor monitors the NetBackup services that run on the local computer and tries to restart a service that unexpectedly terminates.</td>
</tr>
<tr>
<td>166</td>
<td>nbvault</td>
<td>The NetBackup Vault Manager manages NetBackup Vault. nbvault must be running on the NetBackup Vault server during all NetBackup Vault operations.</td>
</tr>
<tr>
<td>178</td>
<td>dsm</td>
<td>The Disk Service Manager (DSM) performs set and get operations on disk storage and disk storage units.</td>
</tr>
<tr>
<td>199</td>
<td>nbftsrvr</td>
<td>The Fibre Transport (FT) server process runs on media servers that are configured for NetBackup Fibre Transport. On the server side of the FT connection, nbftsrvr controls data flow, processes SCSI commands, manages data buffers, and manages the target mode driver for the host bus adapters. nbftsrvr is part of SAN Client.</td>
</tr>
<tr>
<td>200</td>
<td>nbftclnt</td>
<td>The Fibre Transport (FT) client process runs on the client and is part of SAN Client.</td>
</tr>
<tr>
<td>201</td>
<td>fsm</td>
<td>The FT Service Manager (FSM) is a component of the Enterprise Media Manager (EMM) and is part of SAN Client.</td>
</tr>
<tr>
<td>202</td>
<td>stssvc</td>
<td>The Storage service manages the storage server and runs on the media server.</td>
</tr>
<tr>
<td>210</td>
<td>ncfive</td>
<td>Exchange Firedrill Wizard for NetBackup clients.</td>
</tr>
</tbody>
</table>
Table 1-3  Originator IDs for the server entities that use unified logging *(continued)*

<table>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>219</td>
<td>rsrcevtmgr</td>
<td>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.</td>
</tr>
<tr>
<td>220</td>
<td>dps</td>
<td>Disk polling service for NetBackup clients.</td>
</tr>
<tr>
<td>221</td>
<td>mpms</td>
<td>The Media Performance Monitor Service (MPMS) runs on every media server within RMMS and gathers CPU load and free memory information for the host.</td>
</tr>
<tr>
<td>222</td>
<td>nbrmms</td>
<td>Remote monitoring and Management Service (RMMS) is the conduit through which EMM discovers and configures disk storage on media servers.</td>
</tr>
<tr>
<td>226</td>
<td>nbstserv</td>
<td>The Storage services controls the lifecycle image duplication operations.</td>
</tr>
<tr>
<td>230</td>
<td>rdsm</td>
<td>The Remote Disk Service Manager interface (RDSM) runs within the Remote Manager and Monitor Service. RDMS runs on media servers.</td>
</tr>
<tr>
<td>231</td>
<td>nbevtmgr</td>
<td>The Event Manager Service provides asynchronous event management services for cooperating participants.</td>
</tr>
<tr>
<td>248</td>
<td>bmrlauncher</td>
<td>The BMR Launcher Utility in the Windows BMR Fast Restore image configures the BMR environment.</td>
</tr>
<tr>
<td>261</td>
<td>aggs</td>
<td>Artifact Generator Generated Source.</td>
</tr>
<tr>
<td>263</td>
<td>wingui</td>
<td>The NetBackup Administration Console for Windows</td>
</tr>
<tr>
<td>271</td>
<td>nbecmsg</td>
<td>Legacy error codes.</td>
</tr>
<tr>
<td>272</td>
<td>expmgr</td>
<td>The Expiration Manager handles the capacity management and the image expiration for storage lifecycle operations.</td>
</tr>
<tr>
<td>286</td>
<td>nbkms</td>
<td>The Encryption Key Management Service is a master server-based symmetric service that provides encryption keys to the media server NetBackup Tape Manager processes.</td>
</tr>
<tr>
<td>293</td>
<td>nbaudit</td>
<td>NetBackup Audit Manager.</td>
</tr>
<tr>
<td>294</td>
<td>nbauditmsgs</td>
<td>NetBackup Audit Messages.</td>
</tr>
<tr>
<td>309</td>
<td>ncf</td>
<td>NetBackup Client Framework.</td>
</tr>
</tbody>
</table>
Table 1-3  Originator IDs for the server entities that use unified logging

<table>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>311</td>
<td>ncfnbservercom</td>
<td>NetBackup Client/Server Communications.</td>
</tr>
<tr>
<td>317</td>
<td>ncfbedspi</td>
<td>NetBackup Client Beds Plug-in.</td>
</tr>
<tr>
<td>318</td>
<td>ncfwinpi</td>
<td>NetBackup Client Windows Plug-in.</td>
</tr>
<tr>
<td>321</td>
<td>dbaccess</td>
<td>NetBackup Relational Database access library.</td>
</tr>
<tr>
<td>348</td>
<td>ncforaclepi</td>
<td>NetBackup Client Oracle Plug-in.</td>
</tr>
<tr>
<td>351</td>
<td>ncflive</td>
<td>Live Browse Client.</td>
</tr>
<tr>
<td>352</td>
<td>ncfgre</td>
<td>Granular restore.</td>
</tr>
<tr>
<td>355</td>
<td>ncfstore</td>
<td>NetBackup TAR Plug-in.</td>
</tr>
<tr>
<td>356</td>
<td>ncfvxmmspi</td>
<td>NetBackup Client VxMS Plug-in.</td>
</tr>
<tr>
<td>357</td>
<td>ncfnbrestore</td>
<td>NetBackup Restore.</td>
</tr>
<tr>
<td>359</td>
<td>ncfnbbrowser</td>
<td>NetBackup Browser.</td>
</tr>
<tr>
<td>360</td>
<td>ncforautil</td>
<td>NetBackup Client Oracle utility.</td>
</tr>
<tr>
<td>361</td>
<td>ncfdb2pi</td>
<td>NetBackup Client DB2 Plug-in.</td>
</tr>
<tr>
<td>362</td>
<td>nbars</td>
<td>NetBackup Agent Request Services.</td>
</tr>
<tr>
<td>363</td>
<td>dars</td>
<td>Database Agent Request Server process call</td>
</tr>
<tr>
<td>366</td>
<td>ncfnbcs</td>
<td>NetBackup Client Service.</td>
</tr>
<tr>
<td>369</td>
<td>impmgr</td>
<td>NetBackup Import Manager.</td>
</tr>
<tr>
<td>371</td>
<td>nbim</td>
<td>Indexing Manager.</td>
</tr>
<tr>
<td>372</td>
<td>nbhsm</td>
<td>Hold service.</td>
</tr>
<tr>
<td>375</td>
<td>ncfnbsearchserverpi</td>
<td>NetBackup Client Search Server Plug-in.</td>
</tr>
<tr>
<td>377</td>
<td>ncfnbdiscover</td>
<td>NetBackup Client Component Discovery.</td>
</tr>
<tr>
<td>380</td>
<td>ncfnbquiescence</td>
<td>NetBackup Client Component Quiescence/Unquiescence.</td>
</tr>
<tr>
<td>381</td>
<td>ncfnbdboffline</td>
<td>NetBackup Client Component Offline/Online.</td>
</tr>
<tr>
<td>386</td>
<td>ncfvmwarepi</td>
<td>NetBackup NCF VMware Plug-in.</td>
</tr>
</tbody>
</table>
Originator IDs for the server entities that use unified logging

<table>
<thead>
<tr>
<th>Originator ID</th>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>387</td>
<td>nbrntd</td>
<td>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 56.</td>
</tr>
<tr>
<td>395</td>
<td>stsem</td>
<td>STS Event Manager.</td>
</tr>
<tr>
<td>396</td>
<td>nbutils</td>
<td>NetBackup Utilities.</td>
</tr>
<tr>
<td>400</td>
<td>nbdisco</td>
<td>NetBackup Discovery.</td>
</tr>
<tr>
<td>401</td>
<td>ncfmssqlpi</td>
<td>NetBackup Client MSSQL plug-in.</td>
</tr>
<tr>
<td>402</td>
<td>ncfexchangepi</td>
<td>NetBackup Client Exchange plug-in.</td>
</tr>
<tr>
<td>403</td>
<td>ncfsharepointpi</td>
<td>NetBackup Client SharePoint plug-in.</td>
</tr>
<tr>
<td>412</td>
<td>ncffilesyspi</td>
<td>NetBackup Client File System plug-in.</td>
</tr>
</tbody>
</table>

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/openv/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 13.
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 unified log files

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MaxLogFileSizeKB</code></td>
<td>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>.</td>
</tr>
<tr>
<td><code>RolloverAtLocalTime</code></td>
<td>Specifies the time of day at which the log file is rolled over, if the <code>RolloverMode</code> is set to <code>LocalTime</code>.</td>
</tr>
<tr>
<td><code>RolloverPeriodInSeconds</code></td>
<td>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>.</td>
</tr>
<tr>
<td><code>MaxLogFileSizeKB or RolloverAtLocalTime</code></td>
<td>Specifies that the log file rollover occurs whenever the file size limit or the local time limit is reached, whichever is first.</td>
</tr>
<tr>
<td><code>MaxLogFileSizeKB or RolloverPeriodInSeconds</code></td>
<td>Specifies that the log file rollover occurs whenever the file size limit or the periodic time limit is reached, whichever is first.</td>
</tr>
</tbody>
</table>

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
MaxLogFileSizeKB=256
RolloverAtLocalTime=22:00
```

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:
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 20.
- 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 13.

<table>
<thead>
<tr>
<th>Limit the number of log files</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The following example sets to 8000 the maximum number of log files that are allowed for all unified logging originators in the NetBackup (product ID 51216). When the number of log files exceeds 8000 for a particular originator, the oldest log files become eligible for deletion during log cleanup.</td>
</tr>
<tr>
<td></td>
<td><code># vxlogcfg -a -p 51216 -o ALL -s NumberOfLogFiles=8000</code></td>
</tr>
<tr>
<td></td>
<td>See “Examples of using vxlogcfg to configure unified logs” on page 37.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specify the number of days the log files are kept</th>
<th>Use the <code>Keep logs for days</code> 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.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In the NetBackup Administration Console, in the left pane, expand <code>NetBackup Management &gt; Host Properties &gt; Master Servers</code>. Double-click the server you want to change. A new dialog box appears. In the left pane, click <code>Logging &gt; Keep logs for days</code>.</td>
</tr>
</tbody>
</table>
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 34.

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/openv/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>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| 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' |
<p>| PID        | Long Integer | Provide the process ID | PID = 1234567 |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>TID</td>
<td>Long Integer</td>
<td>Provide the thread ID</td>
<td>TID = 2874950</td>
</tr>
<tr>
<td>STDATE</td>
<td>Long Integer or string</td>
<td>Provide the start date in seconds or in the locale-specific short date and time format. For example, a locale may have format 'mm/dd/yy hh:mm:ss AM/PM'</td>
<td>STDATE = 98736352</td>
</tr>
<tr>
<td>ENDATE</td>
<td>Long Integer or string</td>
<td>Provide the end date in seconds or in the locale-specific short date and time format. For example, a locale may have format 'mm/dd/yy hh:mm:ss AM/PM'</td>
<td>ENDATE = 99736352</td>
</tr>
<tr>
<td>PREVTIME</td>
<td>String</td>
<td>Provide the hours in 'hh:mm:ss' format. This field should be used only with operators =, &lt;, &gt;, &gt;=, and &lt;=</td>
<td>PREVTIME = '2:34:00'</td>
</tr>
<tr>
<td>SEV</td>
<td>Integer</td>
<td>Provide one of the following possible severity types:</td>
<td>SEV = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = INFO</td>
<td>SEV = INFO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = WARNING</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = ERR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = CRIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = EMERG</td>
<td></td>
</tr>
<tr>
<td>MSGTYPE</td>
<td>Integer</td>
<td>Provide one of the following possible message types:</td>
<td>MSGTYPE = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = DEBUG (debug messages)</td>
<td>MSGTYPE = DIAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = DIAG (diagnostic messages)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = APP (application messages)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 =CTX (context messages)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = AUDIT (audit messages)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 1-5
**Data types for fields (continued)**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTX</td>
<td>Integer or string</td>
<td>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 !=.</td>
<td>CTX = 78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CTX = 'ALL'</td>
</tr>
</tbody>
</table>

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:

- `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:

- 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**

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`(PRODID == 51216) &amp;&amp; ((PID == 178964)</td>
<td></td>
</tr>
<tr>
<td>((prodid = 'NBU') &amp;&amp; ((stdate &gt;= '11/18/09 0:0:0 AM') &amp;&amp; (endate &lt;= '12/13/09 13:0:0 PM')))</td>
<td></td>
</tr>
</tbody>
</table>
Table 1-6  Examples of query strings (continued)

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(STDATE &lt;= ‘04/05/09 0:0:0 AM’)</td>
<td>Retrieves the log messages that were logged on or before 2009-05-04 for all the installed Symantec products.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display all the attributes of the log messages</td>
<td><code>vxlogview -p 51216 -d all</code></td>
</tr>
</tbody>
</table>
| Display specific attributes of the log messages                     | Display the log messages for NetBackup (51216) that show only the date, time, message type, and message text:  
|                                                                     | `vxlogview --prodid 51216 --display D,T,m,x` |
| Display the latest log messages                                     | Display the log messages for originator 116 (nbpem) that were issued during the last 20 minutes. Note that you can specify `-o nbpem instead of -o 116:  
|                                                                     | # vxlogview -o nbpem -t 00:20:00 |
| Display the log messages from a specific time period               | Display the log messages for nbpem that were issued during the specified time period:  
|                                                                     | # vxlogview -o nbpem -b "05/03/05 06:51:48 AM" -e "05/03/05 06:52:48 AM" |
### Table 1-7 Example uses of the vxlogview command (continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display results faster</td>
<td>You can use the <code>-i</code> option to specify an originator for a process:</td>
</tr>
<tr>
<td></td>
<td># vxlogview -i nbpem</td>
</tr>
<tr>
<td></td>
<td>The <code>vxlogview -i</code> option searches only the log files that the specified process (nbpem) 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you use the <code>-i</code> option with a process that is not a service, <code>vxlogview</code> returns the message &quot;No log files found.&quot; 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.</td>
</tr>
<tr>
<td></td>
<td>The <code>-i</code> option displays entries for all OIDs that are part of that process including libraries (137, 156, 309, etc.).</td>
</tr>
<tr>
<td>Search for a job ID</td>
<td>You can search the logs for a particular job ID:</td>
</tr>
<tr>
<td></td>
<td># vxlogview -i nbpem</td>
</tr>
<tr>
<td></td>
<td>The <code>jobid=</code> search key should contain no spaces and must be lowercase.</td>
</tr>
<tr>
<td></td>
<td>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 (nbpem). 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>.</td>
</tr>
</tbody>
</table>

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

http://www.symantec.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

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the log files</td>
<td>List all unified log files for the <code>nbrb</code> service:</td>
</tr>
<tr>
<td></td>
<td># vxlogmgr -s -o nbrb</td>
</tr>
<tr>
<td></td>
<td>/usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log</td>
</tr>
<tr>
<td></td>
<td>/usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log</td>
</tr>
<tr>
<td></td>
<td>/usr/openv/logs/nbrb/51216-118-1342895976-050505-00.log</td>
</tr>
<tr>
<td></td>
<td>Total 3 file(s)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete the oldest log files</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td># vxlogcfg -a -p 51216 -o nbrb -s NumberOfLogFiles=1</td>
</tr>
<tr>
<td></td>
<td># vxlogmgr -d -o nbrb -a</td>
</tr>
<tr>
<td></td>
<td>Following are the files that were found:</td>
</tr>
<tr>
<td></td>
<td>/usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log</td>
</tr>
<tr>
<td></td>
<td>/usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log</td>
</tr>
<tr>
<td></td>
<td>Total 2 file(s)</td>
</tr>
<tr>
<td></td>
<td>Are you sure you want to delete the file(s)? (Y/N):</td>
</tr>
<tr>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Deleting /usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log ...</td>
</tr>
<tr>
<td></td>
<td>Deleting /usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log ...</td>
</tr>
<tr>
<td>Delete the newest log files</td>
<td>Delete all the unified log files that NetBackup created in the last 15 days:</td>
</tr>
<tr>
<td></td>
<td># vxlogmgr -d --prodid 51216 -n 15</td>
</tr>
<tr>
<td></td>
<td>Make sure that you roll over (rotate) the log files before you recycle them.</td>
</tr>
<tr>
<td>Delete the log files for a specific originator</td>
<td>Delete all unified log files for originator <code>nbrb</code>:</td>
</tr>
<tr>
<td></td>
<td># vxlogmgr -d -o nbrb</td>
</tr>
<tr>
<td></td>
<td>Make sure that you roll over (rotate) the log files before you recycle them.</td>
</tr>
<tr>
<td>Delete all the log files</td>
<td>Delete all unified log files for NetBackup:</td>
</tr>
<tr>
<td></td>
<td># vxlogmgr -d -p NB</td>
</tr>
<tr>
<td></td>
<td>Make sure that you roll over (rotate) the log files before you recycle them.</td>
</tr>
</tbody>
</table>
Table 1-8  Example uses of the vxlogmgr command (continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control the number of log</td>
<td>You can use the <code>vxlogmgr</code> command with the <code>vxlogcfg</code> command's <code>NumberOfLogFiles</code> option to manually delete log files.</td>
</tr>
<tr>
<td>files</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td><code># vxlogmgr -a -d</code></td>
</tr>
<tr>
<td></td>
<td>The following command keeps the two most recent log files of all PBX originators:</td>
</tr>
<tr>
<td></td>
<td><code># vxlogmgr -a -d -p ics</code></td>
</tr>
<tr>
<td></td>
<td>The following deletes the older log files for the <code>nbrb</code> service only:</td>
</tr>
<tr>
<td></td>
<td><code># vxlogmgr -a -d -o nbrb</code></td>
</tr>
<tr>
<td>Control disk space</td>
<td>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.</td>
</tr>
<tr>
<td>usage</td>
<td>The disk space that a given originator uses can be calculated as follows:</td>
</tr>
<tr>
<td></td>
<td><code>NumberOfFiles for originator * MaxLogFileSizeKB for originator</code></td>
</tr>
<tr>
<td></td>
<td>The total disk space that unified logs consume is the sum of the disk space that each originator consumes. If none of the originators overrides the <code>NumberOfFiles</code> and <code>MaxLogFileSizeKB</code> settings, then the total disk space that unified logging consumes is as follows:</td>
</tr>
<tr>
<td></td>
<td><code>Number of originators * default MaxLogFileSizeKB * default NumberOfFiles</code></td>
</tr>
<tr>
<td></td>
<td>Use the <code>vxlogcfg</code> command to list the current unified logging settings.</td>
</tr>
<tr>
<td></td>
<td>For example, assume the following:</td>
</tr>
<tr>
<td></td>
<td>1. <code>vxlogmgr -a -d -p NB</code> is configured as a <code>cron</code> job with a frequency of one hour.</td>
</tr>
<tr>
<td></td>
<td>2. No originators override default settings for <code>MaxLogFileSizeKB</code> or <code>NumberOfFiles</code>.</td>
</tr>
<tr>
<td></td>
<td>3. The number of active NetBackup originators on the host is 10. (Typical of a NetBackup master server that is not running BMR or NDMP.)</td>
</tr>
<tr>
<td></td>
<td>4. The default <code>MaxLogFileSizeKB</code> is equal to 51200.</td>
</tr>
<tr>
<td></td>
<td>5. The default <code>NumberOfFiles</code> is equal to 3.</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>`10 * 51200 * 3 KB = 1,536,000 KB of additional disk space used each hour.</td>
</tr>
</tbody>
</table>

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.

- The vxlogcfg options for robust file logging (MaxLogFileSizeKB and NumberOfLogFiles) also affect certain legacy logs.
  
  See “About limiting the size and the retention of legacy logs” on page 50.

- 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>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the maximum log file size</td>
<td>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. You can change the maximum file size with the MaxLogFileSizeKB option. The following command changes the default maximum log size to 100000 KB for the NetBackup product: # vxlogcfg -a -p 51216 -o Default -s MaxLogFileSizeKB=100000 For MaxLogFileSizeKB to be effective, the RolloverMode option must be set to FileSize: # vxlogcfg -a --prodid 51216 --orgid Default -s RolloverMode=FileSize MaxLogFileSizeKB can be set per originator. An originator that is not configured uses the default value. The following example overrides the default value for service nbrb (originator 118): # vxlogcfg -a -p 51216 -o nbrb -s MaxLogFileSizeKB=1024</td>
</tr>
</tbody>
</table>
Table 1-9  Example uses of the vxlogcfg command (continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set log recycling</td>
<td>The following example sets automatic log file deletion for nbemm logs (originator ID 111):</td>
</tr>
<tr>
<td></td>
<td># vxlogcfg -a --prodid 51216 --orgid 111 -s</td>
</tr>
<tr>
<td></td>
<td>RolloverMode=FileSize MaxLogFileSizeKB=512000</td>
</tr>
<tr>
<td></td>
<td>NumberOfLogFiles=999 LogRecycle=TRUE</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Set debug level and diagnostic level</td>
<td>The following example sets the default debug level and diagnostic level of product ID NetBackup (51216):</td>
</tr>
<tr>
<td></td>
<td># vxlogcfg -a --prodid 51216 --orgid Default -s</td>
</tr>
<tr>
<td></td>
<td>DebugLevel=1 DiagnosticLevel=6</td>
</tr>
</tbody>
</table>
### Table 1-9  Example uses of the vxlogcfg command (continued)

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

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 15.
In legacy debug logging, each process creates log files of debug activity in its own logging directory. Each log file grows to a certain size before the NetBackup process closes it and creates a new log file. For more details on the legacy log file name format, refer to the following section:

See “File name format for legacy logging” on page 45.

The NetBackup legacy debug log directories are located in the following directories:

Windows

\install_path\NetBackup\logs
\install_path\Volmgr\debug

UNIX

/usr/openv/netbackup/logs
/usr/openv/volmgr/debug

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 (the \bpbrm, \bpcm, \bptdm, and \bptm directories). To enable logging for all NetBackup processes that use legacy logging, you must create the log file directories that do not exist already.

See “Directory names for legacy debug logs for servers” on page 46.
See “Directory names for legacy debug logs for media and device management” on page 48.

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/openv/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.symantec.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.

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

Windows

\install_path\Volmgr\debug\vmscd\n
UNIX

/usr/openv/volmgr/debug/vmscd
As an alternative, you can restart `vmscd` 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 the debug log directories at once:

- **Windows**: `Install_path\NetBackup\Logs\mklogdir.bat`
- **UNIX**: `usr/openv/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**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bp</code></td>
<td>Menu driven client-user interface program.</td>
</tr>
<tr>
<td><code>bparchive</code></td>
<td>Archive program. Also useful for debugging <code>bp</code>.</td>
</tr>
<tr>
<td><code>bpbackup</code></td>
<td>Backup program. Also useful for debugging <code>bp</code>.</td>
</tr>
<tr>
<td><code>bpbbkar</code></td>
<td>Program that is used to generate backup images.</td>
</tr>
<tr>
<td><code>bpcd</code></td>
<td>NetBackup client daemon or manager.</td>
</tr>
<tr>
<td><code>bpclimagelist</code></td>
<td>Command-line utility that produces a status report on client NetBackup images or removable media.</td>
</tr>
<tr>
<td><code>bpclntcmd</code></td>
<td>Command-line utility on clients that test NetBackup system functionality and enables Fibre Transport services.</td>
</tr>
<tr>
<td><code>bphdb</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>bpjava-msvc</code></td>
<td>The NetBackup-Java application server authentication service that <code>inetd</code> starts during startup of the NetBackup Java interface applications. This program authenticates the user that started the application.</td>
</tr>
<tr>
<td>Directory</td>
<td>Associated process</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bpjava-usvc</td>
<td>The NetBackup program that bpjava-msvc starts upon successful logon through the logon dialog box that is presented when a NetBackup-Java interface is started. This program services all requests from the Java administration and user interfaces on the host where bpjava-msvc is running.</td>
</tr>
<tr>
<td>bplist</td>
<td>Program that lists backed up and archived files. Also useful for debugging bp. On pre-7.6 versions of NetBackup, the bpclntcmd command and the bpclimagelist command send their debug log messages to the bplist directory. On NetBackup 7.6, bpclntcmd and bpclimagelist send their debug log messages to the bpclntcmd and bpclimagelist directory, respectively.</td>
</tr>
<tr>
<td>bpmount</td>
<td>Program that determines local mount points and wildcard expansion for multiple data streams.</td>
</tr>
<tr>
<td>bporaexp</td>
<td>Command-line program on clients to export Oracle data in XML format. Communicates with bprd on server.</td>
</tr>
<tr>
<td>bporaexp64</td>
<td>64-bit command-line program on clients to export Oracle data in XML format. Communicates with bprd on server.</td>
</tr>
<tr>
<td>bporaimp</td>
<td>Command-line program on clients to import Oracle data in XML format. Communicates with bprd on server.</td>
</tr>
<tr>
<td>bporaimp64</td>
<td>64-bit command-line program on clients to import Oracle data in XML format. Communicates with bprd on server.</td>
</tr>
<tr>
<td>bprestore</td>
<td>Restore program. Also useful for debugging bp.</td>
</tr>
<tr>
<td>db_log</td>
<td>For more information on these logs, see the NetBackup guide for the database-extension product that you use.</td>
</tr>
<tr>
<td>mtfrd</td>
<td>These logs have information about the mtfrd process, which is used for phase 2 imports and restores of Backup Exec media.</td>
</tr>
<tr>
<td>tar</td>
<td>tar process during restores.</td>
</tr>
</tbody>
</table>
Table 1-10  UNIX client processes that use legacy logging (continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_ops</td>
<td>The user_ops 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 (jbpSA) 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. In addition, on NetBackup-Java capable platforms, the NetBackup Java interface log files are written in a subdirectory that is called nbjlogs. All files in the user_ops directory hierarchy are removed according to the setting of the KEEP_LOGS_DAYS configuration option.</td>
</tr>
</tbody>
</table>

PC client processes that use legacy logging

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

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

<table>
<thead>
<tr>
<th>Directory</th>
<th>NetBackup client</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpinetd</td>
<td>Windows2003</td>
<td>Client service logs. These logs have information on the bpinetd32 process.</td>
</tr>
<tr>
<td>bparchive</td>
<td>Windows 2003</td>
<td>Archive program that is run from the command line.</td>
</tr>
<tr>
<td>bpbackup</td>
<td>Windows 2003</td>
<td>The backup program that is run from the command line.</td>
</tr>
<tr>
<td>Directory</td>
<td>NetBackup client</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bpbkar</td>
<td>Windows 2003</td>
<td>Backup and archive manager. These logs have information on the bpbkar32 process.</td>
</tr>
<tr>
<td>bpcd</td>
<td>All Windows clients</td>
<td>NetBackup client daemon or manager. These logs have information on communications between the server and client.</td>
</tr>
<tr>
<td>bpjava-msvc</td>
<td></td>
<td>The NetBackup-Java application server authentication service that the Client Services service starts during startup of the NetBackup Java interface applications. This program authenticates the user that started the application. (On all Windows platforms.)</td>
</tr>
<tr>
<td>bpjava-usvc</td>
<td></td>
<td>NetBackup program that bpjava-msvc starts upon successful logon through the logon dialog box that is presented when a NetBackup-Java interface is started. This program services all requests from the Java administration and user interfaces on the NetBackup host where bpjava-msvc is running. (On all Windows platforms.)</td>
</tr>
<tr>
<td>bplist</td>
<td>Windows 2003</td>
<td>List program that is run from the command line.</td>
</tr>
<tr>
<td>bpmount</td>
<td>Windows 2003</td>
<td>The program that is used to collect drive names on the client for multistreaming clients.</td>
</tr>
<tr>
<td>bprestore</td>
<td>Windows 2003</td>
<td>The restore program that is run from the command line.</td>
</tr>
<tr>
<td>tar</td>
<td>Windows 2003</td>
<td>tar process. These logs have information about the tar32 process.</td>
</tr>
</tbody>
</table>
Table 1-11  PC client processes that use legacy logging (continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>NetBackup client</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_ops</td>
<td>Windows 2003</td>
<td>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 (jbpSA) 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. In addition, on NetBackup-Java capable platforms, the NetBackup Java interface log files are written in a subdirectory that is called <code>nbjlogs</code>. All files in the <code>user_ops</code> directory hierarchy are removed according to the setting of the KEEP_LOGS_DAYS configuration option.</td>
</tr>
</tbody>
</table>

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`.

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

In versions of NetBackup earlier than 7.7, log file names are of the following format:

- On Windows: `mmddyy_nnnnn.log`
- On Windows: `mmddyy.log`
- On UNIX: `log.mmddyy`

Retention of all logs files in the legacy debug log directories is managed 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 50.

**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.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Administrative commands</td>
</tr>
<tr>
<td>bpbbrm</td>
<td>NetBackup backup and restore manager</td>
</tr>
<tr>
<td>bpcd</td>
<td>NetBackup client daemon or manager. The NetBackup Client service starts this process.</td>
</tr>
<tr>
<td>bpjobd</td>
<td>NetBackup jobs database manager program</td>
</tr>
<tr>
<td>bpdm</td>
<td>NetBackup disk manager</td>
</tr>
</tbody>
</table>
### Directory names for legacy debug logs (continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bpdbm</strong></td>
<td>NetBackup Database Manager. This process runs only on master servers. On Windows systems, it is the NetBackup Database Manager service.</td>
</tr>
<tr>
<td><strong>bpjava-msvc</strong></td>
<td>The NetBackup-Java application server authentication service that is started when the NetBackup Java interface applications start. On UNIX servers, \texttt{inetd} starts it. On Windows servers, the Client Services service starts it. This program authenticates the user that started the application.</td>
</tr>
<tr>
<td><strong>bpjava-susvc</strong></td>
<td>The NetBackup program that \texttt{bpjava-msvc} starts upon successful logon through the logon dialog box that is presented when a NetBackup-Java interface starts. This program services all requests from the Java user interfaces on the NetBackup master or media server host where the \texttt{bpjava-msvc} program runs (all Windows platforms).</td>
</tr>
<tr>
<td><strong>bprd</strong></td>
<td>NetBackup request daemon or manager. On Windows systems, this process is called the NetBackup Request Manager service.</td>
</tr>
<tr>
<td><strong>bpsynth</strong></td>
<td>The NetBackup process for synthetic backup. \texttt{nbjm} starts \texttt{bpsynth}. \texttt{bpsynth} runs on the master server.</td>
</tr>
<tr>
<td><strong>bptm</strong></td>
<td>NetBackup tape management process</td>
</tr>
<tr>
<td><strong>nbatd</strong></td>
<td>Authentication daemon (UNIX and Linux) or service (Windows). \texttt{nbatd} authenticates access to interfaces of NetBackup services or daemons.</td>
</tr>
<tr>
<td><strong>nbazd</strong></td>
<td>Authorization daemon (UNIX and Linux) or service (Windows). \texttt{nbazd} authorizes access to interfaces of NetBackup services or daemons.</td>
</tr>
<tr>
<td><strong>syslogs</strong></td>
<td>System log</td>
</tr>
<tr>
<td></td>
<td>You must enable system logging to troubleshoot \texttt{ltid} or robotic software. See the \texttt{syslogd} man page.</td>
</tr>
<tr>
<td><strong>user_ops</strong></td>
<td>The \texttt{user_ops} directory is created during the install of NetBackup on all servers and clients. NetBackup Java interface programs use it for the following: temporary files and for job and progress log files that the \texttt{Backup, Archive, and Restore} program (\texttt{jbpSA}) generates. This directory must exist for successful operation of any of the Java programs and must have public read, write, and execute permissions. \texttt{user_ops} contains a directory for every user that uses the Java programs. In addition, on NetBackup-Java capable platforms, the NetBackup Java interface log files are written in the \texttt{nbjlogs} subdirectory. All files in the \texttt{user_ops} directory hierarchy are removed according to the setting of the \texttt{KEEP_LOGS_DAYS} configuration option.</td>
</tr>
</tbody>
</table>
Table 1-12  Directory names for legacy debug logs (continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnetd</td>
<td>The Symantec network daemon, used to create firewall-friendly socket connections. Started by the <code>inetd(1M)</code> process. Note: Logging occurs in either the <code>/usr/openv/logs</code> directory or the <code>/usr/openv/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/openv/netbackup/logs/vnetd</code>.</td>
</tr>
</tbody>
</table>

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

See “Multiplexed backup process” on page 71.

On UNIX systems, also refer to the `README` file in the `/usr/openv/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

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td>acsissi</td>
<td>UNIX only. Debug information on transactions between NetBackup and the StorageTek ACSLS server.</td>
</tr>
<tr>
<td>daemon</td>
<td>Debug information for <code>vmd</code> (NetBackup Volume Manager service, Windows) and its associated processes (<code>oprd</code> and <code>rdevmi</code>). Stop and restart <code>vmd</code> after creating the directory.</td>
</tr>
<tr>
<td>ltid</td>
<td>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.</td>
</tr>
<tr>
<td>reqlib</td>
<td>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.</td>
</tr>
<tr>
<td>robots</td>
<td>Debug information on all robotic daemons, which includes <code>tldcd</code>, <code>tl8cd</code>, and <code>tl4d</code> daemons. Stop and restart robotic daemons.</td>
</tr>
</tbody>
</table>
### Table 1-13

<table>
<thead>
<tr>
<th>Directory</th>
<th>Associated process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tpcommand</strong></td>
<td>Debug information for device configuration, including the <code>tpconfig</code> and the <code>tpautoconf</code> commands and the NetBackup Administration Console.</td>
</tr>
<tr>
<td><strong>vmscd</strong></td>
<td>Debug information for the NetBackup Status Collection daemon. Stop and restart <code>vmscd</code> after creating the directory.</td>
</tr>
</tbody>
</table>

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

- **Windows**: `install_path\Volmgr\debug`
- **UNIX**: `/usr/openv/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/openv/volmgr/debug/daemon`

See “File name format for legacy logging” on page 45.

See “About limiting the size and the retention of legacy logs” on page 50.

See “Directory names for legacy debug logs for media and device management” on page 48.

### 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 56.

**Note:** This setting also affects unified logging.
On UNIX, add a `VERBOSE` entry in the `/usr/openv/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.
- `<processname>_VERBOSE = -3` the corresponding process only logs informational, warning, and error messages in the debug logs.

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  install_path\Volmgr\n
UNIX     /usr/openv/volmgr/
```

**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 13.

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 26.

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):

- bpbm
- bpcd
- bpdbm
- bpdm
- bprd
- bptm
- nbproxy

For the logs created by 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 infinite retention. The vm.conf file is located in the following directory:

Windows  \install_path\Volmgr\
UNIX  /usr/openv/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 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 13.

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 `MAX_NUM_LOGFILES` options. These options are part of the `bpsetconfig` command, which is located in the following directory:

  Windows: \install_path\NetBackup\bin\admincmd\Windows
  UNIX: /usr/openv/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.

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>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | Create directories on the master server. | Create the following directories:  
`install_path/netbackup/logs/bpsynth`  
`install_path/netbackup/logs/bpdbm`  
`install_path/netbackup/logs/vnetd` |
Table 1-14  Creating legacy log directories (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Create directories on the media server.</td>
<td>Create the following directories:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>install_path/netbackup/logs/bpcd</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>install_path/netbackup/logs/bptm</em></td>
</tr>
<tr>
<td>Step 3</td>
<td>Change the Global logging level.</td>
<td>In <em>Host Properties</em>, select a master server and set the <strong>Global logging level</strong> to 5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See the <em>NetBackup Troubleshooting Guide</em> for more information on how to use the Host Properties window to access configuration settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Changing the logging level” on page 56.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “About global logging levels” on page 54.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Rerun the job.</td>
<td>Rerun the job and gather the logs from the directories that you created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The <em>bptm</em> logs are required only if the images are read from or written to a tape device or disk. The <em>bpdm</em> logs are needed only if the images are read from disk.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the images are read from multiple media servers, the debug logs for <em>bptm</em> or <em>bpdm</em> must be collected from each media server.</td>
</tr>
</tbody>
</table>

See “Logs to accompany problem reports for synthetic backups” on page 58.

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 1-15 describes all logging levels and the detail that each level includes.

Table 1-15  Global logging levels

<table>
<thead>
<tr>
<th>Logging level</th>
<th>Description</th>
</tr>
</thead>
</table>


### Global logging levels (continued)

<table>
<thead>
<tr>
<th>Logging level</th>
<th>Description</th>
</tr>
</thead>
</table>
| Minimum logging| Includes very important, low-volume diagnostic messages and debug messages. The Host Properties Logging page or Logging Assistant can set minimum logging. Legacy logs use the following values to represent minimum logging:  
  - Windows: Registry displays the following hexadecimal value: 0xffffffff  
  - UNIX: The `bp.conf` file displays `VERBOSE = 0 (global)`. `<processname>_VERBOSE = 0` represents using the global default for an individual process. If the global VERBOSE value is set to a value other than 0, an individual process can be decreased by using the value -1. For example, `processnameVERBOSE = -1`. Unified logging uses the value 1 to represent minimum logging. |
| Disable logging | The Host Properties Logging page or Logging Assistant can set disable logging. Legacy logs use the following values to represent disabled logging:  
  - UNIX: The `bp.conf` file displays `VERBOSE=-2 (global) or <processname>_VERBOSE = -2` for an individual process.  
  - Windows: Registry displays the following hexadecimal value: 0xfffffffe. Unified logging uses the value 0 to represent disabled logging. |
| 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 Logging Assistant.

See “Changing the logging level” on page 56.

See “Examples of using vxlogcfg to configure unified logs” on page 37.

- 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 49.

- 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 48.

- 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.
In the properties dialog box, in the left pane, click **Logging**.

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 54.

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. 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 **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 37.

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 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.

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 Symantec Technical Support.

Include the following log types:

- Log files that unified logging creates
  See “Gathering unified logs for NetBackup” on page 17.

- Log files that legacy logging creates
  See “Creating legacy log directories to accompany problem reports for synthetic backup” on page 53.

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 53.

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

NetBackup Windows master servers can be configured so messages from NetBackup reports are written to the Windows Event Viewer Application log. You can see these messages in the Application log and also use third-party tools to monitor the Application log for these messages.

To route unified logging application and diagnostic messages for an originator to the Application log, set the LogToOslog value to true for that originator.

The following example routes the application and diagnostic messages for nbrb to the Windows Event Log:
# vxlogcfg -a -o nbrb -p NB -s "LogToOslog=true"

**Note:** For this setting to take effect, restart NetBackup services.

To enable the logging tool, 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
  ```

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

**Severity**
- 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**
- 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**
- 1 = Unknown
- 2 = Debug
- 4 = Info
- 8 = Warning
- 16 = Error
- 32 = Critical
You can configure the eventlog file to log the messages that include several different severities and types. Consider the results that the following entry in the eventlog file produces:

Entry 56 255

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)

Consider the following example message that 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 may appear in the status line (bottom) of the NetBackup Administration Console window. They also may appear in the log file that contains the stdout or the stderr messages that the Java APIs or the NetBackup Administration Console write. Symantec does not document Java exception errors.

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

Table 1-16  Error message types

<table>
<thead>
<tr>
<th>Error type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetBackup status codes and messages</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> A status code does not always accompany the error message.</td>
</tr>
<tr>
<td></td>
<td>To find the status code, look up the NetBackup message in the alphabetical listing and click the link to see a full description.</td>
</tr>
<tr>
<td></td>
<td>See the Status Codes Reference Guide.</td>
</tr>
<tr>
<td>NetBackup Administration Console: application</td>
<td>These messages have status codes in the 500 range. Messages with status codes 500, 501, 502, 503 and 504 begin with &quot;Unable to login, status: &quot;. Messages with status codes 511 and 512 may or may not begin with &quot;Unable to login, status: &quot;.</td>
</tr>
<tr>
<td>server status codes and messages</td>
<td><strong>Note:</strong> A status code does not always accompany the error message.</td>
</tr>
<tr>
<td></td>
<td>See the Status Codes Reference Guide.</td>
</tr>
</tbody>
</table>
Table 1-16  Error message types (continued)

<table>
<thead>
<tr>
<th>Error type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java exceptions</td>
<td>Either the Java APIs or NetBackup Administration APIs generate these exceptions. These messages begin with the name of the exception. For example:</td>
</tr>
<tr>
<td></td>
<td>java.lang.ClassCastException</td>
</tr>
<tr>
<td></td>
<td>vrtx.nbu.NBUCommandExecutionException</td>
</tr>
<tr>
<td>Java exceptions</td>
<td>Java exceptions usually appear in one of the following places:</td>
</tr>
<tr>
<td></td>
<td>■ The status line (bottom) of the NetBackup Administration window</td>
</tr>
<tr>
<td></td>
<td>■ The log file that the jnbSA or jbpSA commands generate</td>
</tr>
<tr>
<td></td>
<td>■ The output file of the Windows Display Console .bat file if it is set up</td>
</tr>
<tr>
<td></td>
<td>See “Troubleshooting error messages in the NetBackup Administration Console” on page 62.</td>
</tr>
<tr>
<td>Operating system</td>
<td>Any messages that do not match those in the NetBackup documentation are most likely messages from the operating system.</td>
</tr>
<tr>
<td>errors</td>
<td></td>
</tr>
</tbody>
</table>

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/openv/netbackup/logs/user_ops

■ On the host where the console was started

■ In /usr/openv/netbackup/logs/user_ops/nbjlogs

If space is not available in the respective file systems, you may 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 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>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | Create logging directories | On the NetBackup client or server that is specified in the logon dialog box, create the following directories:  
- bpjava-msvc  
- \(\text{bpjava-susvc}\) (if a NetBackup server)  
- \(\text{bpjava-usvc}\) (if a NetBackup client)  
Create the directories in the following locations:  
- \(\text{install}\_\text{path}\backslash\text{NetBackup}\backslash\text{logs}\) (Windows)  
- \(\text{/usr/openv/netbackup/logs}\) (UNIX)  
See “About unified logging” on page 15.  
See “About legacy logging” on page 39. |
| Step 2 | Edit the \(\text{Debug.properties}\) file | Add the following line to the \(\text{Debug.properties}\) file:  
debugMask=2  
The \(\text{Debug.properties}\) file can be found in the following locations:  
- \(\text{/usr/openv/java}\)  
  Change the file on the UNIX machine where you run the jnbSA or jbpSA commands. The log file name is displayed in the xterm window where you ran the jnbSA or jbpSA commands.  
- \(\text{install}\_\text{path}\backslash\text{VERITAS}\backslash\text{java}\)  
  Change the file at this location if you use the NetBackup Java Windows Display Console. |
| Step 3 | Edit the \(\text{nbjava.bat}\) file | Perform this step if you use the Windows Display Console on a host where NetBackup is not installed.  
Edit the \(\text{nbjava.bat}\) file to redirect output to a file.  
The \(\text{nbjava.bat}\) file is located in \(\text{install}\_\text{path}\backslash\text{VERITAS}\backslash\text{java}\)  
See the \(\text{nbjava.bat}\) file for details. |

This detailed debug logging provides more information than the NetBackup Administration Console logging that you can configure in the Administration Console itself. See the NetBackup Administrator’s Guide, Volume I:  
http://www.symantec.com/docs/DOC5332
Backup process and logging

This chapter includes the following topics:

- Backup process
- NetBackup process descriptions
- About backup logging
- Sending backup logs to Symantec 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.
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 may be of use in resolving backup problems include: `bpbackup`, `reqlib`, `daemon`, `robots`, and `acsissi`.

**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.

**Backup and restore startup process**

When the NetBackup master server starts up, a script automatically starts all services, daemons, and programs that are required by NetBackup. (The start-up commands that are used by the script 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.

As of NetBackup version 7.6, remote EMM servers are no longer supported. 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 sub components 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 `nbpe` 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 `nbpe`, 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 `bpare` command on the command line. This action invokes the client’s `bpbackup` or `bpare` program, which sends a request to the request daemon `bprd` on the master server. When `bprd` receives the user request, it contacts `nbpe`, which checks the policy configurations for schedules. By default `nbpe` 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 are the common log files that are used in review of media and master server failures:

See “`nbpe` logging” on page 146.

See “`nbpr` logging” on page 146.

See “`bpd` logging” on page 141.
See “bprd logging” on page 142.
See “nbjm logging” on page 145.
See “bpjobd logging” on page 142.
See “nrbp logging” on page 146.
See “nbemmm logging” on page 145.
See “bpcompatd logging” on page 141.
See “PBX logging” on page 147.
See “vnetd logging” on page 149.
See “bpcd logging” on page 141.
See “bpbrm logging” on page 140.
See “bpbkar logging” on page 140.
See “bptm logging” on page 143.
See “ltid logging” on page 144.
See “txxd and txxcd logging” on page 149.

The additional logs that are not included in the backup process flow but that may be helpful in resolving backup problems include: bpbackup, reqlib, daemon, robots, and acsssi.

See “bpbackup logging” on page 139.
See “reqlib logging” on page 147.
See “daemon logging” on page 144.
See “robots logging” on page 148.
See “acsssi logging” on page 139.

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

See “Sending backup logs to Symantec Technical Support” on page 72.

### Sending backup logs to Symantec Technical Support

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

See “Logs to accompany problem reports for synthetic backups” on page 58.

*Table 2-1* provides a list of logs and the recommended logging levels that Symantec Technical Support may need to diagnose certain backup issues.
Note: Symantec recommends that the diagnostic level for unified logging be set at the default level of 6.

See “About global logging levels” on page 54.

### Table 2-1 Logs to gather for specific backup issues

<table>
<thead>
<tr>
<th>Type of problem</th>
<th>Logs to gather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problems with backup scheduling</td>
<td>- The nbpem log at debug level 5</td>
</tr>
<tr>
<td></td>
<td>- The nbjm log at debug level 5</td>
</tr>
<tr>
<td></td>
<td>- The nbproxy log at verbose 4</td>
</tr>
<tr>
<td></td>
<td>- The bpdbm log at verbose 2</td>
</tr>
<tr>
<td></td>
<td>- The bprd log at verbose 5</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The bprd log is only needed for problems with manual or user-initiated backups.</td>
</tr>
<tr>
<td>Problems with queued backup jobs that don't go active</td>
<td>- The nbpem log at debug level 3</td>
</tr>
<tr>
<td></td>
<td>- The nbjm log at debug level 5</td>
</tr>
<tr>
<td></td>
<td>- The nbrb log at debug level 4</td>
</tr>
<tr>
<td></td>
<td>- The nbproxy log at verbose 4</td>
</tr>
<tr>
<td></td>
<td>- The bpdbm log at verbose 2</td>
</tr>
<tr>
<td></td>
<td>- The nbemm logs at the default levels</td>
</tr>
<tr>
<td></td>
<td>- The mds log at debug level 2</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The mds log writes to the nbemm log.</td>
</tr>
<tr>
<td>Problems with active backup jobs that don't write</td>
<td>- The nbjm log at debug level 5</td>
</tr>
<tr>
<td></td>
<td>- The nbrb log at debug level 4</td>
</tr>
<tr>
<td></td>
<td>- The bpdbm log at verbose 2</td>
</tr>
<tr>
<td></td>
<td>- The bpbrm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bptm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpcd log at verbose 5</td>
</tr>
<tr>
<td>If the problem is a tape load or unload issue, Support may also need the following logs:</td>
<td>- The ltid log</td>
</tr>
<tr>
<td></td>
<td>- The reqlib log</td>
</tr>
<tr>
<td></td>
<td>- The daemon log</td>
</tr>
<tr>
<td></td>
<td>- The robots log</td>
</tr>
<tr>
<td></td>
<td>- The acsssi log (UNIX only)</td>
</tr>
</tbody>
</table>
See “Setting Media Manager debug logging to a higher level” on page 57.
See “About backup logging” on page 71.
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 77.

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 84.

TL8, TLH, and TLD require following types of daemons:
Each host with a robotic drive attached must have a robotic daemon. These daemons provide the interface between \texttt{ltid} 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 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.
Starting media and device management

At system startup, the server automatically starts ltid, which starts applicable robotic daemons.

To start processes manually, enter:
On UNIX: /usr/openv/netbackup/bin/bp.start_all
On Windows: <install_path>\NetBackup\bin\bpup

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 69.

The resulting request to mount a device is passed from nbjm to nbrb, which acquires the physical resources from nbemm (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 shared by itself and ltid. Drive busy status also appears in the Device Monitor.
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.
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 oprd, 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 oprd, ltid, and avrd in the media server’s shared memory.

Figure 3-3 illustrates the Shared Storage Option management process.
Barcode operations

Barcode reading is mainly a function of the robot hardware rather than media and device management. When a robot has a barcode reader, it scans any barcode that may be 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 83.

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).
Media and device management components

This topic shows the file and directory structure and the programs and daemons associated with the 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 equivalent files and directories that are located in the directory where NetBackup is installed (by default, C:\Program Files\VERITAS).
Figure 3-5  Media and device management directories and files

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

<table>
<thead>
<tr>
<th>File or directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Commands, scripts, programs, daemons, and files required for media and device management. Three subdirectories under bin are available. driver: Contains the SCSI drivers used on various platforms to control robotics. goodies: Contains the <code>vm.conf</code> script and scan utility.</td>
</tr>
<tr>
<td>debug</td>
<td>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.</td>
</tr>
<tr>
<td>help</td>
<td>Help files that are used by media and device management programs. These files are in ASCII format.</td>
</tr>
<tr>
<td>misc</td>
<td>Lock files and temporary files that are required by various components of media and device management.</td>
</tr>
</tbody>
</table>
Table 3-1  Media and device management directories and files (continued)

<table>
<thead>
<tr>
<th>File or directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm.conf</td>
<td>Media and device management configuration options.</td>
</tr>
</tbody>
</table>

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 discussed in this table reside under `/usr/openv/volmgr/bin`. On Windows, they reside under `install_path\volmgr\bin`.

**Note:** The following table contains references to the system log. This log is managed by syslog on UNIX (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

<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| acsd              | The Automated Cartridge System daemon interfaces with the Automated Cartridge System. It communicates with the server that controls the ACS robotics through the acsssi process (UNIX) or the STK Libattach Service (Windows). Also, for UNIX, see the acsssi and acssel programs.  
**Started By:** Starting ltid (or on UNIX, independently by using the `/usr/openv/volmgr/bin/ascd` command.  
**Stopped By:** Stopping ltid (or on UNIX, independently by finding the PID (process id) and then using the kill command).  
**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the vm.conf file. On UNIX, debug information is also included by starting the daemon with the -v option: this option can also be used through ltid, or by putting VERBOSE in the vm.conf file. |
| acssel            | Available only on UNIX.  
See the [NetBackup Device Configuration Guide](http://example.com). |
| acsssi            | Available only on UNIX.  
See the [NetBackup Device Configuration Guide](http://example.com). |
<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| avrd             | The automatic-volume-recognition daemon controls automatic volume assignment and label scanning. This lets NetBackup read labeled tape volumes and automatically assign the associated removable media to requesting processes.  

**Started By:** Starting `ltid` (or on UNIX, independently by using the `/usr/openv/volmgr/bin/avrd` command).  

**Stopped By:** Stopping `ltid`, (or on UNIX, independently by finding the PID (process id) and then using the `kill` command).  

**Debug Log:** All errors are logged in the system log. Debug information is included by adding `VERBOSE` to the `vm.conf` file. On UNIX, debug information is also included by aborting `avrd` and starting the daemon with the `-v` option. |
| ltid             | The device demon (UNIX) or NetBackup Device Manager service (Windows) controls the reservation and assignment of tapes.  

**Started By:** `/usr/openv/volmgr/bin/ltid` command on UNIX or Stop/Restart Device Manager Service command in Media and Device Management window on Windows.  

**Stopped By:** `/usr/openv/volmgr/bin/stopltid` command on UNIX or Stop/Restart Device Manager Service command in the Media and Device Management window on Windows.  

**Debug Log:** Errors are logged in the system log and `ltid` debug log. Debug information is included if the daemon is started with the `-v` option (available only on UNIX) or adding `VERBOSE` to the `vm.conf` file. |
| tl4d             | The Tape Library 4MM daemon is the interface between `ltid` and the Tape Library 4MM and communicates with the robotics through a SCSI interface.  

**Started By:** Starting `ltid` (or on UNIX, independently by using the `/usr/openv/volmgr/bin/tl4d` command).  

**Stopped By:** Stopping `ltid` (or on UNIX, independently by finding the PID (process id) and then using the `kill` command).  

**Debug Log:** All errors are logged in the system log. Debug information is included by adding `VERBOSE` to the `vm.conf` file. On UNIX, debug information is also included by starting the daemon with the `-v` option (either by itself or through `ltid`). |
### Table 3-2  Media and device management daemons and programs (continued)

<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| tl8d              | 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 may be attached to different hosts than the robotic control. tl8d is the interface between the local ltid 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 ltid and then to the local tl8d (all on the same host). tl8d then forwards the request to tl8cd on the host that is controlling the robot (could be on another host).  
**Started By:** Starting ltid (or on UNIX, independently by using the /usr/openv/volmgr/bin/tl8d command).  
**Stopped By:** Stopping ltid (or on UNIX, independently by finding the PID (process id) and then using the kill command.  
**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the vm.conf file. On UNIX, debug information is also included by starting the daemon with the -v option (either by itself or through ltid). |
| tl8cd             | 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.  
**Started By:** Starting ltid (or on UNIX, independently by using the /usr/openv/volmgr/bin/tl8d command).  
**Stopped By:** Stopping ltid or by using the tl8cd -t command.  
**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the vm.conf file. On UNIX, debug information is also included by starting the daemon with the -v option (either by itself or through ltid). |
Table 3-2  Media and device management daemons and programs (continued)

<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| **tldd**          | The Tape Library DLT daemon works in conjunction with tldcd to handle requests to TLD robots (Tape Library DLT and Tape Stacker DLT). tldd provides the interface between the local ltid and the robotic control (tldcd) in the same manner as explained previously for tl8d.  

**Started By:** Starting ltid (or on UNIX, independently by using the /usr/openv/volmgr/bin/tldd command).  

**Stopped By:** Stopping ltid (or on UNIX, independently by finding the PID (process id) and then using the kill command).  

**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the vm.conf file. On UNIX, debug information is also included by starting the daemon with the -v option (either by itself or through ltid). |
| **tlhd**          | The Tape Library Half-inch daemon works in conjunction with tlhdcd to handle requests to 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 manner as explained previously for tl8d.  

**Started By:** Starting ltid (or on UNIX, independently by using the /usr/openv/volmgr/bin/tlhd command).  

**Stopped By:** Stopping ltid (or on UNIX, independently by finding the PID (process id) and then using the kill command).  

**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included by adding VERBOSE to the vm.conf file. On UNIX, debug information is also included by starting the daemon with the -v option (either by itself or through ltid). |
<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| tlhcd             | The tape library half-inch control daemon provides robotic control for a TLH robot that is in an IBM Automated tape library (ATL) in a similar manner to that which was explained previously for tl8cd.  
**Started By:** Starting `ltid` (or on UNIX, independently by using the `/usr/openv/volmgr/bin/tlhcd` command).  
**Stopped By:** Stopping `ltid` or by using the `tlhcd -t` command.  
**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included if the daemon is started with the `-v` option (either by itself or through `ltid`). The `-v` option is available only on UNIX. Also, add the VERBOSE option to the `vm.conf` file. |
| tlmd              | The Tape Library Multimedia daemon is the interface between `ltid` 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.  
**Started By:** Starting `ltid` or independently by using the `/usr/openv/volmgr/bin/tlmd` command.  
**Stopped By:** Stopping `ltid` or independently by finding the PID (process id) and then using the `kill` command.  
**Debug Log:** Errors are logged in the system log and robots debug log. Debug information is included if the daemon is started with the `-v` option (either by itself or through `ltid`). The `-v` option is available only on UNIX. Also, add the VERBOSE option to the `vm.conf` file. |
| tshd              | The Tape Stacker Half-inch daemon is the interface between `ltid` and the half-inch-cartridge stacker and communicates with the robotics through a SCSI interface. This robot is not supported on Windows.  
**Started By:** Starting `ltid` (or on UNIX, independently by using the `/usr/openv/volmgr/volmgr/bin/tshd` command).  
**Started By:** `tpconfig` command.  
**Stopped By:** Quit option from within the utility on UNIX. On Windows, `tpconfig` is only a command-line interface that runs to completion (no quit option).  
**Debug Log:** `tpcommand` debug logs. |
Table 3-2  Media and device management daemons and programs (continued)

<table>
<thead>
<tr>
<th>Program or daemon</th>
<th>Description</th>
</tr>
</thead>
</table>
| **vmd**           | The Volume Manager daemon (NetBackup Volume Manager service on Windows) allows remote administration and control of Media and Device Management.  
  **Started By:** Starting ltid  
  **Stopped By:** Terminate Media Manager Volume Daemon option.  
  **Debug Log:** System log and also a debug log if the daemon or reqlib debug directories exist |
| **vmscd**         | The Media Manager Status Collector Daemon keeps the EMM server database up-to-date with the actual status of drives attached to 5.x servers.  
  **Started By:** the EMM server.  
  **Stopped By:** the EMM server.  
  **Debug Log:** /usr/openv/volmgr/debug/vmscd (UNIX), install_path\Volmgr\debug\vmscd (Windows) |
Restore process and logging

This chapter includes the following topics:

- Restore process
- UNIX client restore
- SAN client restore
- Windows client restore
- About restore logging
- Sending restore logs to Symantec 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

- (2) bprestore
- (3) bpdbm
- (1) bprd
- (4) bpjobd
- (5) bpbrm
- (6) bptm
- (8) nbjm
- (7) tar
- (11) ltid
- (9) nbrb
- (12) txxd/
- (13) txxcd
- (10) nbemm
- Tape
- EMM Database

Outbound Connection/Communication Paths
- PBX
- vnetd

Color Legend:
- Master Server
- Enterprise Media Manager Server
- Media Server
- Client Server
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 may 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.
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 (`bpbrm`).

  **Note:** To restore Backup Exec images, `bpbrm` initiates `mtfrd` instead of `tar` 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 `tar` 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 `nbemm`) 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), `tar` 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 `tar` 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 tar` 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.

---

**SAN client restore**

Figure 4-4 shows the server and client components that are used in a restore of a SAN client over Fibre Channel.
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 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 (`bpbrm`).

**Note:** To restore Backup Exec images, `bpbrm` invoke `mtfrd` instead of `tar` 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 `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 shared memory, and writes 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 buffer is full or job is done, sets 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.

### 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 `tar` 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 4-5** shows the client processes involved in these operations.
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 Symantec Technical Support.

See “Sending restore logs to Symantec Technical Support” on page 102.

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

See “bprd logging” on page 142.

See “bprestore logging” on page 143.
Sending restore logs to Symantec Technical Support

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

See “Logs to accompany problem reports for synthetic backups” on page 58.

Table 4-1 provides a list of logs and the recommended logging levels that Symantec Technical Support may need to diagnose certain restore issues.

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

See “About global logging levels” on page 54.
Table 4-1  Log to gather for specific restore issues

<table>
<thead>
<tr>
<th>Type of problem</th>
<th>Log to gather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problems with restore jobs from tape</td>
<td>- The nbjm log at debug level 5</td>
</tr>
<tr>
<td></td>
<td>- The nbemm log at debug level 1</td>
</tr>
<tr>
<td></td>
<td>- The nbrb log at debug level 4</td>
</tr>
<tr>
<td></td>
<td>- The bpdbm log at verbose 1</td>
</tr>
<tr>
<td></td>
<td>- The bprd log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpprm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The tar log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpcd log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>If the problem is a media or a drive issue,</td>
</tr>
<tr>
<td></td>
<td>Support may also need the following logs:</td>
</tr>
<tr>
<td></td>
<td>- The reqlib log</td>
</tr>
<tr>
<td></td>
<td>- The daemon log</td>
</tr>
<tr>
<td></td>
<td>- The robots log</td>
</tr>
<tr>
<td></td>
<td>- The acsssi log (UNIX only)</td>
</tr>
<tr>
<td>Problems with restore jobs from disk</td>
<td>- The bpdbm log at verbose 1</td>
</tr>
<tr>
<td></td>
<td>- The bprd log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpprm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bptm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpdm log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The tar log at verbose 5</td>
</tr>
<tr>
<td></td>
<td>- The bpcd log at verbose 5</td>
</tr>
</tbody>
</table>

See “Setting Media Manager debug logging to a higher level” on page 57.

See “About restore logging” on page 101.
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.
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, nbrb, 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 wakeup timer for the next due job.
- When the job finishes, re-computes the due time of the next job and submits to `nbjm` all jobs that are currently due.

3 The Job Manager service (`nbjm`) requests backup resources from the Resource Broker (`nbrb`), which returns information on the use of shared memory for 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 (`nbftsrvr`).
   - Sends a backup request to the FT client process on the client (`nbftclnt`), which does the following: Opens a Fibre Channel connection to `nbftsrvr` on the media server, allocates shared memory, and writes shared memory information to the backup ID file.

6 The `bpbrm` service uses `bpcd` to start `bpbkar`, which 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 `tar`, 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 (`nbftsrvr`) shared memory buffer, and clears the buffer flag.

8 The `nbftsrvr` service waits for data from `nbftclnt`; and writes the data is written to the shared memory buffer. When the transfer completes, `nbftsrvr` 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**

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 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 (`bpbrm`).

**Note:** To restore Backup Exec images, `bpbrm` invoke `mtfrd` instead of `tar` 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 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 shared memory, and writes 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 buffer is full or job is done, sets 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

Hot catalog backup is policy based, 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 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.
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/openv/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/openv/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, bpbkm, 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/openv/netbackup/logs/bpdbm`  
  `usr/openv/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 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.
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/openv/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/openv/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/openv/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 only previously created backup images, called component images.

*Note:* Synthetic archives do not exist.

For example, an existing full image and subsequent differential incremental images may 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:
In phase 1, bpsynth makes a synthetic backup request to the database manager, bpdbm. 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 bpdbm service returns the list of extents to bpsynth. (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.

The following figure shows how phase 1 operates:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Prepare catalog information and extents</td>
<td>In phase 1, bpsynth makes a synthetic backup request to the database manager, bpdbm. 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 bpdbm service returns the list of extents to bpsynth. (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.</td>
</tr>
<tr>
<td>2 - Obtain resources</td>
<td>In phase 2, bpsynth 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. When the component images reside on BasicDisk, no resource reservation is done.</td>
</tr>
</tbody>
</table>
Table 5-1  (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 - Copy data</td>
<td>In phase 3, bpsynth starts the writer bptm (for tape and disk) on the media server to write the new synthetic image. It also starts a reader bptm (tape) or bpdm (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. The following figure shows how phase 3 operates:</td>
</tr>
<tr>
<td></td>
<td>Note that bpsynth only starts the parent bptm (writer) and bpdm (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.</td>
</tr>
<tr>
<td></td>
<td>The bpsynth process sends the extents (starting block and count) for each component image to the corresponding child bptm or bpdm reader process.</td>
</tr>
<tr>
<td></td>
<td>The parent bptm or bpdm reader process reads the data from the appropriate media into the shared buffers. The child bptm or bpdm reader process sends the data in the shared buffers to the child bptm writer process over a socket.</td>
</tr>
<tr>
<td></td>
<td>The child bptm writer process writes the data into the shared buffers. The parent bptm writer process copies the data from the shared buffers to the media and notifies bpsynth when the synthetic image is complete.</td>
</tr>
<tr>
<td>4 - Validate the image</td>
<td>In phase 4, the bpsynth process validates the image. The new image is now visible to NetBackup and can be used like any other full or cumulative incremental backup.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
Storage logging

This chapter includes the following topics:

- NDMP backup logging
- NDMP restore logging

NDMP backup logging

The following shows an NDMP backup process.

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

1 The NetBackup administrator runs the `bpbackup` command to start the backup job. Or, a scheduled policy that is created on the NetBackup Administration Console can initiate the job.

2 The `bpbackup` 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 (`nbem`), 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 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 `bpjobd`, who uses this information to update the job status which 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.
The basic processing steps for a NDMP restore operation are the following:

**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 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 and the Job Manager (nbjm) requests the required resources. This resource request causes 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`), which 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 (`bpjobd`) 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 `bpbkarc` sends data to the NetBackup backup tape manager - the `bptm` process
- `pvdvs` (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` may write `tlogs` to the `.tlog` files in the queue (`dedupe_path\queue`) directory and to the processed directory. The `tlog` data from the queue directory
will be processed into the crdb later when the next content router queue processing job runs. Beginning with NetBackup 7.7, .tlog files no longer contain additions to the database.

The functional overview is as follows:

**Figure 7-1** Deduplication client backup to MSDP

In this scenario, the client is backing up data directly to the media server and the media server is deduplicating the data before storing 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 on the media server:

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

   ```
   BPTM_VERBOSE = 5
   ```
1. Edit the /usr/openv/lib/ost-plugins/pd.conf file (Windows: install_path\NetBackup\bin\ost-plugins\pd.conf) file so the following lines are uncommented:
   Windows:
   DEBUGLOG = C:\pdplugin.log
   UNIX/Linux:
   DEBUGLOG = /var/log/puredisk/pdplugin.log
   LOGLEVEL = 10

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 may have a performance impact 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 hostname that ran the backup and the bptm process id number (pid).
   - Find a line similar to bptm(pid=value); this 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 "\[3144\]" 092611.log > 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/openv/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 Admininstration 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 msdp/pdplugin responses to NBU):
   
   ```
   # 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 msdp/pdplugin responses to NBU):
  ```
  # 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`

Java Administration Console logging:

Open the `Debug.Properties` file, in `C:\Program Files\VERITAS\Java` (for Windows) or `/usr/openv/java` (for UNIX/Linux) and 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/openv/netbackup/logs/user_ops/nbjlogs` (UNIX/Linux). Ensure that you
are looking at the most recent log.

- **Storage Service** (turn on STS logging, to log the `msdp/pdplugin` responses to
  NBU):
  ```
  # 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 `bpsm` 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 impact 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 running 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**

The following keywords are used by Support when reviewing the logs.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum fragment size</td>
<td>Should be 51200 KB or less</td>
</tr>
<tr>
<td>get_plugin_version</td>
<td><code>libstspipd.dll</code> (pdplugin version)</td>
</tr>
<tr>
<td>get_agent_cfg_file_path_for_mount</td>
<td>Uses the PureDisk agent configuration file (note the <code>.cfg</code> filename); determines shortname or FQDN</td>
</tr>
<tr>
<td>emmlib_NdmpUserldQuery</td>
<td>Used for backups, the credential check</td>
</tr>
<tr>
<td>Resolved</td>
<td>Name resolution of the remote CR</td>
</tr>
<tr>
<td>tag_nbu_dsid read</td>
<td>Checks if it read the NBU_PD_SERVER object correctly</td>
</tr>
<tr>
<td>Recommended routing table</td>
<td>CR routing table for the CR's to route fingerprint/so's; more useful when PDDO targets PureDisk</td>
</tr>
<tr>
<td>for primary backups</td>
<td>Primary backup dsid</td>
</tr>
<tr>
<td>for opt-dup copies from</td>
<td>opt-dup dsid</td>
</tr>
<tr>
<td>this is opt-dup</td>
<td>opt-dup dsid</td>
</tr>
<tr>
<td><strong>Keyword</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>https</td>
<td>Webservice calls to either SPA or CR to check if they completed</td>
</tr>
</tbody>
</table>
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, bkar32 invokes bpfis to take a snapshot of individual drives. 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 bpbkar on the client. bpbkar 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 is backing up itself (bptm and bpbkar are on the same host) or the media server is backing up a client that resides on a different host.

   - If the media server is backing up itself, bpbkar stores the snapshot-based image block-by-block in shared memory on the media server.
   - If the media server is backing 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).
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.

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 (nb pem) triggers a backup job when the policy, schedule, and virtual machine are due and the backup window is open. The nb pem process, the Job Manager (nb jm), the Resource Broker (nb rb), and the Enterprise Media Manager (nb emm) 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 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. nb pem uses nb jm 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. nb jm starts an instance of bpbrm through the client service (bpcd) on the media server, and 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 (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 nb jm. nb jm relays the information and status to nb pem.

8. nb pem submits a child job for the backup to nb jm, with a file list derived from the snapshot information. nb jm starts bpbrm to back up the snapshot.

9. bpbrm uses bpcd to start bpbkar on the VMware Backup Host.
The backup and archive manager (bpbkar) starts VxMS and sends the file catalog information to bpbrm, which relays it to the database manager bpdbm on the master server.

bpbrm starts the process bptm (parent) on the media server. The bpbkar process loads VERITAS Mapping Services (VxMS) which implements VMware Disk Development Kit (VDDK) APIs. The APIs are used for reading from vSphere datastore. VxMS maps the stream during runtime and identifies the contents of the VMDK file.

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

- Symantec 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. bpbrm 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 offhost backup with no burden on the production vSphere. Even the source side deduplication occurs in an offhost system.

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 8-1 shows the overall snapshot backup process. PBX (not shown in the diagram) must be running for NetBackup to operate.
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 `nb pem`. `nb pem` 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 bpbkar on the client. bpkar 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 bpkar on the same host), or the media server backs up a client on a different host. If the media server backs up itself, bpkar 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
- bpkar logging
- bpbrm logging
- bpcd logging
- bpcompatd logging
- bpdbm logging
- bjobjd logging
- bprd logging
- bprestore logging
- bptm logging
- daemon logging
- ltid logging
- nbemm logging
- nbjm logging
- nbpem logging
- nbproxy logging
- nbrb 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: /usr/openv/volmgr/debug/acsssi |
| Server where it resides | media |
| How to access | The *acsssi* 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 39. |

See “About backup logging” on page 71.

See “About restore logging” on page 101.

**bpbackup logging**

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

| Log location | Windows: install_path\NetBackup\logs\bpbackup |
| Server where it resides | client |
| How to access | The *bpbackup* 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 39. |
See “About backup logging” on page 71.

**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**

- **Windows**: `install_path\NetBackup\logs\bpbkar`
- **UNIX**: `/usr/openv/netbackup/logs/bpbkar`

**Server where it resides**

- `client`

**How to access**

The bpbkar 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 39.

See “About backup logging” on page 71.

**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**

- **Windows**: `install_path\NetBackup\logs\bpbrm`
- **UNIX**: `/usr/openv/netbackup/logs/bpbrm`

**Server where it resides**

- `media`

**How to access**

The bpbrm 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 39.

See “About backup logging” on page 71.

See “About restore logging” on page 101.
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/openv/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 39.

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/openv/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 39.

bpdbm logging

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

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.

<table>
<thead>
<tr>
<th>Log location</th>
<th>Windows: <code>install_path\NetBackup\logs\bpdbm</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX: <code>/usr/openv/netbackup/logs/bpdbm</code></td>
</tr>
<tr>
<td>Server where it resides</td>
<td>master</td>
</tr>
<tr>
<td>How to access</td>
<td>The <strong>bpdbm</strong> 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 39.</td>
</tr>
</tbody>
</table>

See “About backup logging” on page 71.
See “About restore logging” on page 101.

### bpjobd logging

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

<table>
<thead>
<tr>
<th>Log location</th>
<th>Windows: <code>install_path\NetBackup\logs\bpjobd</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX: <code>/usr/openv/netbackup/logs/bpjobd</code></td>
</tr>
<tr>
<td>Server where it resides</td>
<td>master</td>
</tr>
<tr>
<td>How to access</td>
<td>The <strong>bpjobd</strong> 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 39.</td>
</tr>
</tbody>
</table>

See “About backup logging” on page 71.
See “About restore logging” on page 101.

### bprd logging

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

<table>
<thead>
<tr>
<th>Log location</th>
<th>Windows: <code>install_path\NetBackup\logs\bprd</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX: <code>/usr/openv/netbackup/logs/bprd</code></td>
</tr>
<tr>
<td>Server where it resides</td>
<td>master</td>
</tr>
</tbody>
</table>
The `bprd` 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 39.

See “About backup logging” on page 71.
See “About restore logging” on page 101.

bprestore logging

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

Log location

Windows: `install_path\NetBackup\logs\bprestore`
UNIX: `/usr/openv/netbackup/logs/bprestore`

Server where it resides client

How to access

The `bprestore` 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 39.

See “About restore logging” on page 101.

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: `install_path\NetBackup\logs\bptm`
UNIX: `/usr/openv/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 39.
See “About backup logging” on page 71.
See “About restore logging” on page 101.

daemon logging

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

<table>
<thead>
<tr>
<th>Log location</th>
<th>Windows: <code>install_path\volmgr\debug\daemon</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX: <code>/usr/openv/volmgr/debug/daemon</code></td>
</tr>
<tr>
<td>Server where it resides</td>
<td>master and media</td>
</tr>
</tbody>
</table>

**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 39.

See “About backup logging” on page 71.
See “About restore logging” on page 101.

ltid logging

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

<table>
<thead>
<tr>
<th>Log location</th>
<th>Windows: <code>install_path\volmgr\debug\ltid</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNIX: <code>/usr/openv/volmgr/debug/ltid</code></td>
</tr>
<tr>
<td>Server where it resides</td>
<td>media</td>
</tr>
</tbody>
</table>

**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 39.

See “About backup logging” on page 71.
See “About restore logging” on page 101.
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/openv/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 15.

See “About backup logging” on page 71.
See “About restore logging” on page 101.

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 bpbrm media manager service as needed, and updates the internal job states.

Log location
Windows: install_path\NetBackup\logs\nbjm
UNIX: /usr/openv/logs/nbjm

Server where it resides
master

How to access
The nbjm 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 15.

See “About backup logging” on page 71.
See “About restore logging” on page 101.
**nb pem logging**

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

**Log location**

- **Windows:** install_path\NetBackup\logs\nb pem
- **UNIX:** /usr/openv/logs/nb pem

**Server where it resides**

master

**How to access**

The nb pem 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 15.

See “About backup logging” on page 71.

**nb proxy logging**

The proxy service nb proxy enables nb pem and nb jm to query master server catalogs.

**Log location**

- **Windows:** install_path\NetBackup\logs\nb proxy
- **UNIX:** /usr/openv/netbackup/logs/nb proxy

**Server where it resides**

master

**How to access**

The nb proxy 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 39.

See “About backup logging” on page 71.

**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: \install_path\NetBackup\logs\nbrb
UNIX: /usr/openv/logs/nbrb

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 15.

See “About backup logging” on page 71.
See “About restore logging” on page 101.

PBX logging

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

Log location

Windows: \install_path\VxPBX\log
UNIX: /opt/VRTSpbx/log

Server where it resides

master, media, and client

How to access

The PBX process uses the unified logging method. Use the vxlogview and vxlogmgr 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.

See “About unified logging” on page 15.

See the NetBackup Troubleshooting Guide for more information on how to access PBX logs.
See “About backup logging” on page 71.
See “About restore logging” on page 101.

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: `install_path\volmgr\debug\reqlib`

UNIX: `/usr/openv/volmgr/debug/reqlib`

Server where it resides

master and media

How to access

The `reqlib` 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 39.

See “About backup logging” on page 71.

See “About restore logging” on page 101.

**robots logging**

The `robots` log includes debug information on all robotic daemons, including the `txxd` and `txxcd` daemons.

Log location

Windows: `install_path\volmgr\debug\robots`

UNIX: `/usr/openv/volmgr/debug/robots`

Server where it resides

media

How to access

The `robots` 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 39.

See “txxd and txxcd logging” on page 149.

See “About backup logging” on page 71.

See “About restore logging” on page 101.

**tar logging**

The Tape Archive program (`tar`) writes restore data to the client disk.

Log location

Windows: `install_path\NetBackup\logs\tar`

UNIX: `/usr/openv/netbackup/logs/tar`

Server where it resides

client
The tar 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 39.

See “About restore logging” on page 101.

**ttxd and txxcd logging**

The robotic daemon (ttxd, 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 ttxd and txxcd processes do not have their own log files. Instead, errors are logged in the robots debug log and the system log. The system log is managed by syslog on UNIX and by the Event Viewer on Windows.

See “About UNIX system logs” on page 13.

See “Logging options with the Windows Event Viewer” on page 59.

**How to access**

Debug information is included by adding the word VERBOSE to the vm.conf file.

See “How to control the amount of information written to legacy logging files” on page 49.

On UNIX, debug information is also included by starting the daemon with the -v option (either by itself or through ltid).

See “robots logging” on page 148.

See “About backup logging” on page 71.

See “About restore logging” on page 101.

**vnetd logging**

The NetBackup Legacy Network Service (vnetd) is a communication mechanism used to create firewall-friendly socket connections.
**Log location**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td><code>install_path\NetBackup\logs\vnetd</code></td>
</tr>
<tr>
<td>UNIX</td>
<td><code>/usr/openv/logs/vnetd</code> or <code>/usr/openv/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/openv/netbackup/logs/vnetd</code>.</td>
</tr>
</tbody>
</table>

**Server where it resides**

- master, media, and client

**How to access**

- The `vnetd` 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 39.

- See “About backup logging” on page 71.
- See “About restore logging” on page 101.
A
acssel, description 85
acsssi logging 139
acsssi, description 85
admin log 46
administration interface
  activity logging 64
  errors 62
application server status codes (Java interface) 62
ascd, description 85
avrd, description 86

B
backup
  NetBackup catalogs 109
  process
    multiplexing 71
  snapshot overview 135
  synthetic processes 113
  UNIX clients 71
backup and archive processes 69
backup and restore startup process 69
backup logging 66, 71
backup process 66
barcode operations 81
basic backup procedure 67
besetconfig 53
bin
  Media and Device Management 84
bp
  UNIX client log 41
bp.conf
    file 71
bparchive
    log 41, 43
bpbackup
    log 41, 43
bpbackup log 43
bpbackup logging 139
BPBACKUP_POLICY 71
BPBACKUP_SCHED 71
bpbkar
  log 41, 44
bpbkar log 44
bpbkar logging 140
bpbrm 137
bpbrm log 46
bpbrm logging 140
bpcd
  server log 46
  UNIX client log 41, 44
bpcd logging 141
bpcompatd logging 141
bpdbjobs log 46
bpdbm log 47
bpdbm logging 141
bpdm log 46
bpfis 137
bphdb
  log 41
BPINETD 100
bpinetd log 43
bpinetd.log 43
bpjava-msvc log 47, 65
bpjava-usvc log 65
bpjobd logging 142
bplist
  log 42, 44
bplist log 44
bpmount
  log 42
bpmount log 44
bporaexp log 42
bporaexp64 log 42
bporaimp log 42
bporaimp64 log 42
bprd log 47
bprd logging 142
bprestore
  log 42, 44
bprestore log 44
bprestore logging 143
bpsynth 113
bptm log 47
bptm logging 143

C
catalog backup 109
client
  NetBackup
debug logs. See UNIX clients. See Windows and NetWare clients

D
daemon logging 144
daemons
  robotic 75
  robotic control 75
database backup (see catalog backup) 109
DAYS_TO_KEEP_LOGS vm.conf setting 52
debug level 57
debug logs 64
  NetBackup 84
  vmd 48, 84
debug.properties file 65
directory structure
  Media and Device Management 83
disk space
    for logs files 36
drive_mount_notify script 78
drive_unmount_notify script 78
driver directory 84

E
EMM server 70
enable debug logging 48
Enterprise Media Manager (EMM) 70
Event viewer logging option 59
eventlog 60
  file entries 60
exception errors in Java admin interface 62

F
fibre channel 104
files
  restore process 95
FSM 104
FT Service Manager 104
functional overview
  Media and Device Management
device management 77

functional overview (continued)
  Media and Device Management (continued)
directories and files 83
  volume management 77
NetBackup
  restores 95
  startup 69

G
Global Logging Level 49
Global logging level 56
Global logging levels 54
goodies directory 84

H
help files
  Media and Device Management 84
hostID
  unified logging 19
Hot catalog backup process 110
Hot catalog restore 111

J
Java interface
  debug logging 64
  troubleshooting background 62
job ID search in unified logs 34

K
Keep logs for setting 28

L
legacy logging 40
  client logs 41
  configuring rotation 52
  controlling size of 50
  directories 40
  locations 40
  PC clients 43
  rotation of 51
levels for logging 54
limiting the size of unified and legacy logs 15
Log level
  Windows clients 57
log retention options 13
logging
  changing location of 26
logging (continued)
levels 54
see legacy logging 40
setting level on PC clients 57
synthetic backup 58
logs
debug
enabling detailed 64
event viewer logging option 59
file retention 28
overview 11
PC client activity
bparchive 43
bpbackup 43
bpbkar 44
bpcd 44
bpinetd 43
bplist 44
bpmount 44
bprestore 44
tar 44
user_ops 45
reports
NetBackup 12
server activity
acssi 48
admin 46
bpbrm 46
bpcd 46
bpbdbjobs 46
bpdbm 47
bpdm 46
bpjava-susvc 47
bpdr 47
bpsynt 47
bptm 47
daemon 48
ltid 48
nbatd 20, 47
nbazd 47
nbjm 21
nbpem 21
reqlib 48
robots 48
syslogs 47
tpcmand 49
setting log size retention 15
setting retention period 50
system 13
logs (continued)
UNIX client activity
bp 41
bparchive 41
bpbackup 41
bpbkar 41
bpcd 41
bpbdb 41
bpjava-susvc 47
bplist 42
bpmount 42
bprestore 42
obackup_tape 42
tar 42
user_ops 43
ltid 50
ltid logging 144
ltid, description 86
M
MAX_LOGFILE_SIZE 53
MAX_NUM_LOGFILES 53
MaxLogFileSizeKB 36–37, 39
media and device management 75
misc file 84
mklogdir.bat 40
moving log locations 26
multiplexed backups 71
N
nbatd log 47
nbazd log 47
nbemm 70
nbemm logging 145
nbftclnt 99, 104, 106, 108
nbftsrvr 99, 104, 106, 108
nbjm 21, 70, 113, 137
nbjm logging 145
nbpem 21, 70–71, 113, 137
nbpem logging 146
nbproxy logging 146
nbrb 70
nbrb logging 146
NBWIN 100
NDMP backup logging 116
NDMP restore logging 118
NDMP restore procedure 119
NetBackup
  product ID 19
NetBackup Administration Console
  debug logging 64
  errors 62
NetBackup process descriptions 69
NetBackup Status Collection daemon. See vmscd network daemon (vnetd) 48
NumberOfFiles 36
NumberOfLogFiles 39

O
obackup_tape log 42
operating system errors 63
originator IDs
  list of 20
originatorID
  unified logging 19

P
PBX logging 147
product ID for NetBackup 19
productID
  unified logging 19

Q
query string 30

R
raw partitions
  restore process 95
reports
  NetBackup 12
  reqlib logging 147
  restore logging 101
  restore procedure from disk 94
  restore procedure from tape 93
  restore process 91, 95
    Windows 2000 client 100
retention
  of logs 28
robot drive selection 77
robotic control daemons 76
robotic daemons 76
robots logging 148
robust file logging 37
RolloverMode 39

rotation
  legacy logging 51
  of logs 26
  unified logging 19

S
SAN Client backup procedure 105
SAN Client backup process flow 105
SAN Client Fiber Transport backup 104
SAN Client Fiber Transport restore 107
SAN client restore 97
sending backup logs 72
server
  NetBackup debug logs 40
Shared Storage Option management process 79
snapshot
  backup process overview 136
Snapshot backup 135
Snapshot Client backup 129
Snapshot Client backup procedure 131
startup
  NetBackup 69
  startup process 75
    media and device management 75
Status Collection Daemon 40
stderr 62
stdout 62
Symantec V-Ray 134
synthetic backup
  logs 58
synthetic backups 113
syslogd 13
system logs 13

T
tar
  log 44
    log files 18
tar logging 148
TAR32 100
tl4d, description 86
tl8cd, description 87
tl8d, description 87
tldcd, description 88
tldd, description 88
tlhd, description 88
tlmd, description 89