NetBackup™ Logging Reference Guide

10.0

VERITAS™

Contents

Chapter 1

Using logs

This chapter includes the following topics:

About logging

Note: The **Logging** properties can be configured on the NetBackup web UI. On the left pane, select **Host > Host properties**. Depending on the host to be configured, select the **Primary Server**, **Media Servers**, or **Clients**. From the Actions menu, select the **Edit** option, and then click **Logging**.

The logging settings determine the behavior for NetBackup logging on the primary server, media server, and the clients:

- Overall logging level or global logging level for all NetBackup processes.
- Overrides for the specific processes that use legacy logging.

- Logging levels for the services that use unified logging.
- Logging for critical processes.
- On clients, the logging level for database applications.
- Log retention settings for NetBackup and for NetBackup Vault (if it is installed).

All NetBackup processes use either unified logging or legacy logging. You can set a global or a unique logging level for certain process and services. Retention levels limit the size of the log files or (for the primary server) the number of days the logs are kept. If you use NetBackup Vault, you can select separate logging retention settings for that option.

See "About unified logging" on page 15.

See "About legacy logging" on page 36.

See "Log retention and log size" on page 12.

Table 1-1 Logging dialog box properties

Property	Description
Global logging level	This setting establishes a global logging level for all processes that are set to **Same as global**. The **Global logging level** affects the legacy and unified logging level of all NetBackup and Enterprise Media Manager (EMM) processes on the server or client. This setting does not affect 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 44.
Process specific overrides	These settings let you override the logging level for the specific processes that use legacy logging.
Debug logging levels for NetBackup services	These settings let you manage the logging level for the specific services that use unified logging.

Table 1-1 Logging dialog box properties *(continued)*

Property	Description
Logging for critical processes	The option lets you enable logging for the critical processes: ■ Primary server processes: `bprd` and `bpdbm`. ■ Media server processes: `bpbrm`, `bptm`, and `bpdm`. ■ Client process: `bpfis` Note the following: ■ If you enable **Logging for critical processes**, also enable the option **Keep logs up to GB**. If you disable this option it may adversely affect NetBackup operations. ■ This option sets the log retention to the default log size. ■ Clicking **Defaults** does not modify the **Logging for critical processes** or the **Keep logs up to GB** options. ■ To disable the logging for critical processes, modify the logging levels for those processes.
Keep logs for days	Specifies the length of time NetBackup keeps information from the error catalog, job catalog, and debug logs. Note that NetBackup derives its reports from the error catalog. The logs can consume a large amount of disk space, so do not keep the logs any longer than necessary. The default is 28 days.
Keep logs up to GB	Specifies the size of the NetBackup logs that you want to retain. When the NetBackup log size grows to this value, the older logs are deleted. ■ For primary and media servers, the recommended value is 25 GB or greater. ■ For clients, the recommended value is 5 GB or greater.
Keep Vault logs for	If NetBackup Vault is installed, select the number of days to keep the Vault session directories, or select **Forever**.

Logging levels

You can choose to apply the same logging level for all NetBackup processes. Or, you can select logging levels for specific processes or services.

Table 1-2 Logging level descriptions

Logging level	Description
Same as global	The process uses the same logging level as the **Global logging level**.
No logging	No log is created for the process.

Table 1-2 Logging level descriptions *(continued)*

Logging level	Description
Minimum logging (default)	A small amount of information is logged for the process. Use this setting unless advised otherwise by Veritas Technical Support. Other settings can cause the logs to accumulate large amounts of information.
Levels 1 through 4	Progressively more information is logged at each level for the process.
5 (Maximum)	The maximum amount of information is logged for the process.

Global logging level

This setting controls the logging level for all processes and for those processes that are set to **Same as global**. You can control the logging level for some NetBackup processes individually.

See the section called “Overrides for legacy logging levels” on page 10.

See the section called “Unified logging levels for the primary server” on page 11.

Overrides for legacy logging levels

These logging levels apply to legacy processes logging. The logging levels that are displayed depend on the type of host (primary, media, or client).

Table 1-3 Logging level overrides for legacy processes

Service	Description	Primary server	Media server	Client
BPBRM logging level	The NetBackup backup and restore manager.	X	X	
BPDM logging level	The NetBackup disk manager.	X	X	
BPTM logging level	The NetBackup tape manager.	X	X	
BPJOBD logging level	The NetBackup Jobs Database Management daemon. This setting is only available for the primary server.	X		
BPDBM logging level	The NetBackup database manager.	X		
BPRD logging level	The NetBackup Request Daemon.	X		

Table 1-3 Logging level overrides for legacy processes *(continued)*

Service	Description	Primary server	Media server	Client
Database logging level	The logging level for database agent logs. For details on which logs to create and refer to, see the guide for the specific agent.			X

Unified logging levels for the primary server

These logging levels apply to NetBackup services logging and are only available for the primary server.

Table 1-4 Logging levels for NetBackup services

Service	Description
Policy Execution Manager	The **Policy Execution Manager** (`NBPEM`) creates policy and client tasks and determines when jobs are due to run. If a policy is modified or if an image expires, `NBPEM` is notified and the appropriate policy and client tasks are updated.
Job Manager	The **Job Manager** (`NBJM`) accepts the jobs that the **Policy Execution Manager** submits and acquires the necessary resources.
Resource Broker	The **Resource Broker** (`NBRB`) makes the allocations for storage units, tape drives, client reservations.

Logging values in the registry, bp.conf file, and unified logging

You can also set logging values in the Windows registry, the bp.conf file, or in unified logging.

Table 1-5 Logging levels and their values

Logging level	Legacy logging - Windows registry	Legacy logging - bp.conf	Unified logging
Minimum logging	Hexadecimal value of `0xffffffff`.	`VERBOSE = 0` (global) *`processname`*`_VERBOSE = 0` 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, *`processname`*`_VERBOSE = -1`.	1
No logging	Hexadecimal value of `0xfffffffe`.	`VERBOSE=-2` (global) *`processname`*`_VERBOSE = -2`	0

Log retention and log size

The following options are available to manage how NetBackup recycles and deletes log files.

Table 1-6 Log retention options in NetBackup

Log retention option	Description	Interface
Keep logs up to GB	Limits the size of unified and legacy logs. For a NetBackup server, the recommended value is 25 GB or greater. For clients, the recommended value is 5 GB or greater. See the section called "Log pruning" on page 13.	This option is available in the host property **Logging** settings.
`NumberOfLogFiles`	Limits the number of unified log files that you want to retain for a NetBackup process. See "About recycling unified log files" on page 27.	`vxlogcfg`
`MaxLogFileSizeKB` and other `RolloverMode` options	Prevents the 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. See "About rolling over unified log files" on page 26.	`vxlogcfg`
Keep logs for days	Limits the days for which NetBackup retains for unified and legacy logs. See "Limit the size and retention of legacy logs" on page 45.	This option is available in the host property **Logging** settings.

Table 1-6 Log retention options in NetBackup *(continued)*

Log retention option	Description	Interface
`MAX_LOGFILE_SIZE` and `MAX_NUM_LOGFILES`	Limit the legacy log size and the number of legacy log files that are retained. See "Limit the size and retention of legacy logs" on page 45.	`bpsetconfig`

Log pruning

All logs are retained until the log size reaches the high water mark, that is, 95% of the **Keep logs up to GB** value. NetBackup verifies the log size every 10 minutes. When the log size reaches the high water mark, NetBackup begins to delete older logs. NetBackup stops deleting logs when the log size reaches the low water mark, 85% of the **Keep logs up to GB** value.

If both **Keep logs up to GB** and **Keep logs for days** are selected, the logs are pruned based on the condition that occurs first.

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

install_path`\NetBackup\logs\nbutils`

`/usr/openv/logs/nbutils`

Changing the logging levels

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

See "Set the Media Manager debug logging to a higher level" on page 14.

See " Changing the logging level on Windows clients" on page 14.

Change the global logging level

The global logging level establishes a logging level for all processes that are set to **Same as global**. Changes affect the logging level of both unified logging and legacy logging.

To change the global 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, select the server or client. Then, double-click to view the properties.

4 In the left pane, click **Logging**.

5 In the **Global logging level** list, select the value that you want.

6 Click **OK**.

Set the Media Manager debug logging to a higher level

Setting the debug logging to a higher level can aid in resolving many error conditions. Choose a debug level, then retry the operation and examine the debug logs.

To set debug logging for media manager 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.

Changing the logging level on Windows clients

When Technical Support advises, you can increase the logging level for client processes to perform troubleshooting. Otherwise, use the default level of 0 as higher levels can cause the logs to accumulate large amounts of information.

Note: You can control the logging level for the Bare Metal Restore process (`bmrsavecfg`) with the `vxlogcfg` command.

See "Examples of using vxlogcfg to configure unified logs" on page 35.

To change the logging level on Windows clients

1 On the client, open the **Backup, Archive, and Restore** interface.

2 Select **File** > **NetBackup Client Properties** and click on the **Troubleshooting** tab.

3 For the **Verbose** setting, enter the advised level or 0 if you finished troubleshooting.

About unified logging

Unified logging creates log file names and messages in a format that is standardized across Veritas products. Only the `vxlogview` command can assemble and display the log information correctly. Server processes and client processes use unified logging.

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 **Logging** host properties. You can also manage unified logging with the following commands:

`vxlogcfg`	Modifies the unified logging configuration settings.
`vxlogmgr`	Manages the log files that the products that support unified logging generate.
`vxlogview`	Displays the logs that unified logging generates. See "Examples of using vxlogview to view unified logs" on page 31.

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.

```
# cd /upload
  ls
```

Example output:

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

4 Tar the log files.

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

Types of unified logging messages

The following message types can appear in unified logging files:

Application log messages

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

An example of an application message follows:

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

Diagnostic log messages

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

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

An example of a diagnostic message follows:

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

Debug log messages

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

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

An example of a debug message follows:

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

File name format for unified logging

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

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

Table 1-7 describes each part of the log file name.

Table 1-7 Description of the file name format for unified logging

Example	Description	Details
51216	Product ID	Identifies the product. The NetBackup product ID is 51216. The product ID is also known as the entity ID.
116	Originator ID	Identifies the log writing entity, such as a process, service, script, or other software. The number 116 is the originator ID of the `nbpem` process (the NetBackup policy execution manager).
2201360136	Host ID	Identifies the host that created the log file. Unless the file was moved, this ID is the host where the log resides.
041029	Date	Shows the date when the log was written in YYMMDD format.
0000000000	Rotation	Identifies the numbered instance of a log file for a given originator. The rollover number (rotation) indicates the instance of this log file. By default, log files roll over (rotate) based on file size. If the file reaches maximum size and a new log file is created for this originator, the new file is designated 0000000001. See "About rolling over unified log files" on page 26.

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 19.

Windows	*install_path*\NetBackup\logs
UNIX	/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 can appear in several different log files.

Table 1-8 lists the NetBackup server and NetBackup client processes, services, and libraries that use unified logging.

Table 1-8 Originator IDs for the server entities that use unified logging

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

Table 1-8 Originator IDs for the server entities that use unified logging *(continued)*

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

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. However, do not save logs to a remote file system such as NFS or CIFS. Logs that are stored remotely can grow large and cause critical performance issues.

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 "Log retention and log size" on page 12.

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 1-9.

Table 1-9 vxlogcfg options that control the rollover of the unified log files

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

A complete description of `vxlogcfg` is in the NetBackup Commands Reference Guide.

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

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

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

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

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

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

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

- Logs for the server processes that use unified logging
  See "Originator IDs for the entities that use unified logging" on page 19.
- 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 "Log retention and log size" on page 12.

Limit the number of log files	Specify the maximum number of log files that NetBackup retains. When the number of log files exceeds the maximum, the oldest log files become eligible for deletion during log cleanup. The `NumberOfLogFiles` option for the `vxlogcfg` command defines that number. In the following example, the maximum number of log files that are allowed for each of the unified logging originators in the NetBackup (product ID 51216) is 8000. When the number of log files exceeds 8000 for a particular originator, the oldest log files become eligible for deletion during log cleanup. `# vxlogcfg -a -p 51216 -o ALL -s NumberOfLogFiles=8000` See "Examples of using vxlogcfg to configure unified logs" on page 35.
Specify the number of days the log files are kept	Use the **Keep logs for days** property to specify the maximum number of days logs are kept. When the maximum number of days is reached, the unified logs and legacy logs are automatically deleted. In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Host Properties > Master Servers**. Double-click the server you want to change. A new dialog box appears. In the left pane, click **Logging > Keep logs for days**.
Explicitly delete the log files	To initiate recycling and delete the log files, run the following command: `# vxlogmgr -a -d` If you cannot manually delete or move files with `vxlogmgr`, the **Keep logs for days** property removes the old logs for both unified logging and legacy logging. See "Examples of using vxlogmgr to manage unified logs" on page 32.

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

Only the vxlogview command can assemble and display the unified logging information correctly. The unified logging files are in binary format and some of the information is contained in an associated resource file. These logs are stored in the following directory. You can display vxlogview results faster by restricting the search to the files of a specific process.

UNIX	/usr/openv/logs
Windows	*install_path*\NetBackup\logs

Table 1-10 Fields in vxlogview query strings

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

Table 1-10 Fields in `vxlogview` query strings *(continued)*

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

Table 1-11 Examples of query strings with dates

Example	Description
`(PRODID == 51216) && ((PID == 178964)|| ((STDATE == '2/5/15 09:00:00 AM') && (ENDATE == '2/5/15 12:00:00 PM'))`	Retrieves the log file message for the NetBackup product ID 51216 between 9AM and 12PM on 2015-05-02.
`((prodid = 'NBU') && ((stdate >= '11/18/14 00:00:00 AM') && (endate <= '12/13/14 12:00:00 PM'))) || ((prodid = 'BENT') && ((stdate >= '12/12/14 00:00:00 AM') && (endate <= '12/25/14 12:00:00 PM')))`	Retrieves the log messages for the NetBackup product NBU between 2014-18-11 and 2014-13-12 and the log messages for the NetBackup product BENT between 2014-12-12 and 2014-25-12.

Table 1-11 Examples of query strings with dates *(continued)*

Example	Description
`(STDATE <= '04/05/15 0:0:0 AM')`	Retrieves the log messages that were logged on or before 2015-05-04 for all of the installed Veritas products.

Examples of using vxlogview to view unified logs

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

Table 1-12 Example uses of the vxlogview command

Item	Example
Display all the attributes of the log messages	`vxlogview -p 51216 -d all`
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 116 -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 "05/03/15 06:51:48 AM"`
	`-e "05/03/15 06:52:48 AM"`

Table 1-12 Example uses of the vxlogview command *(continued)*

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

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 1-13 Example uses of the vxlogmgr command

Item	Example

List the log files	List all unified log files for the `nbrb` service:
	
	`# vxlogmgr -s -o nbrb`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050505-00.log`
	`Total 3 file(s)`

Table 1-13 Example uses of the vxlogmgr command *(continued)*

Item	Example

Delete the oldest log files	If the `vxlogcfg NumberOfLogFiles` option is set to 1, the following example deletes the two oldest log files for the `nbrb` service:
	
	`# vxlogcfg -a -p 51216 -o nbrb -s NumberOfLogFiles=1`
	`# vxlogmgr -d -o nbrb -a`
	`Following are the files that were found:`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log`
	`Total 2 file(s)`
	`Are you sure you want to delete the file(s)? (Y/N):`
	`Y`
	`Deleting`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050504-00.log ...`
	`Deleting`
	`/usr/openv/logs/nbrb/51216-118-1342895976-050503-00.log ...`

Delete the newest log files	Delete all the unified log files that NetBackup created in the last 15 days: `# vxlogmgr -d --prodid 51216 -n 15` Make sure that you roll over (rotate) the log files before you recycle them.
Delete the log files for a specific originator	Delete all unified log files for originator `nbrb`: `# vxlogmgr -d -o nbrb` Make sure that you roll over (rotate) the log files before you recycle them.
Delete all the log files	Delete all unified log files for NetBackup: `# vxlogmgr -d -p NB` Make sure that you roll over (rotate) the log files before you recycle them.

Table 1-13 Example uses of the vxlogmgr command *(continued)*

Item	Example

Control the number of log files	You can use the `vxlogmgr` command with the `vxlogcfg` command's `NumberOfLogFiles` option to manually delete log files.
	
	For example, the `NumberOfLogFiles` 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:
	
	`# vxlogmgr -a -d`
	
	The following command keeps the two most recent log files of all PBX originators:
	
	`# vxlogmgr -a -d -p ics`
	
	The following deletes the older log files for the `nbrb` service only:
	
	`# vxlogmgr -a -d -o nbrb`

Control disk space usage	Periodically run the `vxlogmgr -a -d` command (such as through a `cron` job) to delete logs and monitor the disk space that unified logging uses. The disk space that a given originator uses can be calculated as follows: `NumberOfLogFiles` for originator * `MaxLogFileSizeKB` for originator The total disk space that unified logs consume is the sum of the disk space that each originator consumes. If none of the originators override the `NumberOfLogFiles` and `MaxLogFileSizeKB` settings, then the total disk space that unified logging consumes is as follows: Number of originators * default `MaxLogFileSizeKB` * default `NumberOfLogFiles` Use the `vxlogcfg` command to list the current unified logging settings. For example, assume the following: ■ `vxlogmgr -a -d -p NB` is configured as a `cron` job with a frequency of one hour. ■ No originators override default settings for `MaxLogFileSizeKB` or `NumberOfLogFiles`. ■ The number of active NetBackup originators on the host is 10. (Typical of a NetBackup master server that is not running BMR or NDMP.) ■ The default `MaxLogFileSizeKB` is equal to 51200. ■ The default `NumberOfLogFiles` is equal to 3. 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: 10 * 51200 * 3 KB = 1,536,000 KB of additional disk space used each hour.

Examples of using vxlogcfg to configure unified logs

Note the following:

- The `vxlogcfg` command is the only way to turn off diagnostic and debug messages in unified logging.
- Absolute paths must be specified. Do not use relative paths.

Table 1-14 Example uses of the vxlogcfg command

Item	Example

Set the maximum log file size	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 ID 118).
	
	`# vxlogcfg -a -p 51216 -o nbrb -s MaxLogFileSizeKB=1024000`

Set log recycling	The following example sets automatic log file deletion for `nbemm` logs (originator ID 111): `# vxlogcfg -a --prodid 51216 --orgid 111 -s RolloverMode=FileSize MaxLogFileSizeKB=512000 NumberOfLogFiles=999 LogRecycle=TRUE` This example sets the `nbemm` logging 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.
Set debug level and diagnostic level	The following example sets the default debug level and diagnostic level of product ID NetBackup (51216): `# vxlogcfg -a --prodid 51216 --orgid Default -s DebugLevel=1 DiagnosticLevel=6`

Table 1-14 Example uses of the vxlogcfg command *(continued)*

Item	Example

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

About legacy logging

In NetBackup legacy debug logging, a process creates log files of debug activity in its own logging directory. By default, NetBackup creates only a subset of logging directories, in the following locations:

Windows
install_path\NetBackup\logs
install_path\Volmgr\debug

UNIX	`/usr/openv/netbackup/logs` `/usr/openv/volmgr/debug`

To use legacy logging, a log file directory must exist for a process. If the directory is not created by default, you can use the Logging Assistant or the `mklogdir` batch files to create the directories. Or, you can manually create the directories. When logging is enabled for a process, a log file is created when the process begins. Each log file grows to a certain size before the NetBackup process closes it and creates a new log file.

Note: It is recommended to always use the `mklogdir` utility present in Windows and Linux to create the legacy log directories for each platform, in order to have appropriate permissions on them.

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

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

Follow these recommendations when you create and use legacy log folders:

- Do not use symbolic links or hard links inside legacy log folders.
- If any process runs for a non-root or non-admin user and there is no logging that occurs in the legacy log folders, use the `mklogdir` command to create a folder for the required user.
- To run a command line for a non-root or non-admin user (troubleshooting when the NetBackup services are not running), create user folders for the specific command line. Create the folders either with the `mklogdir` command or manually with the non-root or non-admin user privileges.

More information

See the NetBackup Commands Reference Guide for a complete description about the `mklogdir` command.

See “Directory names for legacy debug logs for servers ” on page 42.

See “Directory names for legacy debug logs for media and device management” on page 44.

See “File name format for legacy logging” on page 41.

UNIX client processes that use legacy logging

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

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

Windows	*install_path*\NetBackup\Logs\mklogdir.bat
UNIX	/usr/openv/netbackup/logs/mklogdir

Table 1-15 UNIX client processes that use legacy logging

Directory	Associated process
bp	Menu driven client-user interface program.
bparchive	Archive program. Also useful for debugging bp.
bpbackup	Backup program. Also useful for debugging bp.
bpbkar	Program that is used to generate backup images.
bpcd	NetBackup client daemon or manager.
bpclimagelist	Command-line utility that produces a status report on client NetBackup images or removable media.
bpclntcmd	Command-line utility on the clients that test NetBackup system functionality and enables Fibre Transport services.
bphdb	Program that starts a script to back up a database on a NetBackup database agent client. See the system administrator's guide for the appropriate NetBackup database agent for more information.
bpjava-msvc	The NetBackup Java application server authentication service that inetd starts during the startup of the NetBackup Java interface applications. This program authenticates the user that started the application.
bpjava-usvc	The NetBackup program that bpjava-msvc starts upon successful logon through the logon dialog box that is presented when a NetBackup Java Backup, Archive, and Restore (BAR) interface is started. This program services all requests from the Java user interfaces on the host where bpjava-msvc is running.
bplist	Program that lists backed up and archived files. Also useful to debug bp.
bpmount	Program that determines the local mount points and wildcard expansion for multiple data streams.

Table 1-15 UNIX client processes that use legacy logging *(continued)*

Directory	Associated process
bporaexp	Command-line program on clients to export Oracle data in XML format. Communicates with bprd on the server.
bporaexp64	64-bit command-line program on clients to export Oracle data in XML format. Communicates with bprd on the server.
bporaimp	Command-line program on clients to import Oracle data in XML format. Communicates with bprd on the server.
bporaimp64	64-bit command-line program on clients to import Oracle data in XML format. Communicates with bprd on the server.
bprestore	Restore program. Also useful for debugging bp.
bptestnetconn	Tests and analyzes DNS and connectivity problems with any specified list of hosts, including the server list in the NetBackup configuration.
db_log	For more information on these logs, see the NetBackup guide for the database-extension product that you use.
mtfrd	These logs have information about the mtfrd process that is used for phase 2 imports and restores of the Backup Exec media.
tar	nbtar processing during restores.
user_ops	The user_ops directory is created during the install of NetBackup on all servers and clients. The NetBackup Java interface programs use this directory for temporary files and for job and progress log files that the **Backup, Archive, and Restore** program (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 of the files that are in the user_ops directory hierarchy are removed according to the setting of the KEEP_LOGS_DAYS configuration option.

PC client processes that use legacy logging

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

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

Table 1-16 PC client processes that use legacy logging

Directory	NetBackup client	Description
bpinetd	All Windows clients	Client service logs. These logs have information on the bpinetd32 process.
bparchive	All Windows clients	Archive program that is run from the command line.
bpbackup	All Windows clients	The backup program that is run from the command line.
bpbkar	All Windows clients	Backup and archive manager. These logs have information on the bpbkar32 process.
bpcd	All Windows clients	NetBackup client daemon or manager. These logs have information on communications between the server and client.
bpjava-msvc		NetBackup Java application server authentication service that the Client Services service starts during startup of the NetBackup Java interface applications. This program authenticates the user who started the application. (On all Windows platforms.)
bpjava-usvc		NetBackup program that bpjava-msvc starts upon successful logon through the logon dialog box that is presented when a NetBackup Java **Backup, Archive, and Restore** (BAR) interface is started. This program services all requests from the Java user interfaces on the NetBackup host where bpjava-msvc is running. (On all Windows platforms.)
bplist	All Windows clients	List program that is run from the command line.
bpmount	All Windows clients	The program that is used to collect drive names on the client for multistreaming clients.
bprestore	All Windows clients	The restore program that is run from the command line.
bptestnetconn	All Windows clients	The program that performs several tasks that help you test and analyze DNS and connectivity problems with any specified list of hosts, including the server list in the NetBackup configuration.
tar	All Windows clients	tar processing. These logs have information about the tar32 process.

Table 1-16 PC client processes that use legacy logging *(continued)*

Directory	NetBackup client	Description
`user_ops`	All Windows clients	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. `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 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.

File name format for legacy logging

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

user_name`.`*mmddyy_nnnnn*`.log`

The file names include the following 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.

nnnnn The counter or the rotation number for the log file. When the counter exceeds the setting for number of log files, the oldest log file is deleted.

The `MAX_NUM_LOGFILES` configuration parameter sets the maximum number of a legacy log file per process.

The new folder structure for non-root or non-admin invoked process logs is created under process log directory name.

For example,

```
/usr/openv/netbackup/logs/tar/root.031020_00001.log
```

```
/usr/openv/netbackup/log/tar/usr1/usr1.031020_00001.log
```

Here, for non-root user usr1, a non-root username directory is created under the respective NetBackup processes.

Directory names for legacy debug logs for servers

NetBackup creates certain directories for legacy logging for servers. Each directory corresponds to a process. Unless it is noted, each directory should be created under the following directory.

Windows	*install_path*\NetBackup\logs
UNIX	/usr/openv/netbackup/logs

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

Table 1-17 describes the directories you need to create to support legacy debug logs for servers.

Table 1-17 Directory names for legacy debug logs

Directory	Associated process
admin	Administrative commands
bpbrm	NetBackup backup and restore manager
bpcd	NetBackup client daemon or manager. The NetBackup Client service starts this process.
bpjobd	NetBackup jobs database manager program
bpdm	NetBackup disk manager
bpdbm	NetBackup Database Manager. This process runs only on master servers. On Windows systems, it is the NetBackup Database Manager service.
bpjava-msvc	The NetBackup Java application server authentication service that is started when the NetBackup interface applications start. On UNIX servers, inetd starts it. On Windows servers, the Client Services service starts it. This program authenticates the user that started the application.

Table 1-17 Directory names for legacy debug logs *(continued)*

Directory	Associated process
bpjava-susvc	The NetBackup program that bpjava-msvc starts upon successful logon through the logon dialog box that is presented when a NetBackup interface starts. This program services all requests from the Java user interfaces on the NetBackup master or media server host where the bpjava-msvc program runs (all Windows platforms).
bprd	NetBackup Request Daemon. On Windows systems, this process is called the NetBackup Request Manager service.
bpsynth	The NetBackup process for synthetic backup. nbjm starts bpsynth. bpsynth runs on the master server.
bptm	NetBackup tape management process
nbatd	Authentication daemon (UNIX and Linux) or service (Windows). nbatd authenticates access to interfaces of NetBackup services or daemons.
nbazd	Authorization daemon (UNIX and Linux) or service (Windows). nbazd authorizes access to interfaces of NetBackup services or daemons.
syslogs	System log You must enable system logging to troubleshoot ltid or robotic software. See the syslogd man page.
user_ops	The user_ops directory is created during the install of NetBackup on all servers and clients. NetBackup interface programs use it for the following: temporary files and for job and progress log files that the **Backup, Archive, and Restore** program (jbpSA) generates. This directory must exist for successful operation of any of the Java programs and must have public read, write, and execute permissions. user_ops contains a directory for every user that uses the Java programs. The NetBackup Java interface log files are written in the nbjlogs subdirectory. All files in the user_ops directory hierarchy are removed according to the setting of the KEEP_LOGS_DAYS configuration option.
vnetd	The Veritas network daemon, used to create firewall-friendly socket connections. Started by the inetd(1M) process. **Note:** Logging occurs in either the /usr/openv/logs directory or the /usr/openv/netbackup/logs if the vnetd directory exists there. If the vnetd directory exists in both locations, logging occurs only in /usr/openv/netbackup/logs/vnetd.

Directory names for legacy debug logs for media and device management

The following directories enable legacy logging for the media management processes and device management processes. NetBackup creates 1 log per day in each of the debug directories. Each directory corresponds to a process. Unless it is noted, each directory should be created under the following directory.

Windows	*install_path*\Volmgr\debug
UNIX	/usr/openv/volmgr/debug

Table 1-18 Media and device management legacy debug logs

Directory	Associated process
acsssi	UNIX only. Debug information on transactions between NetBackup and the StorageTek ACSLS server.
daemon	Debug information for vmd (NetBackup Volume Manager service, Windows) and its associated processes (oprd and rdevmi). Stop and restart vmd after creating the directory.
ltid	Debug information on ltid, the Media Manager device daemon (UNIX), or on the NetBackup Device Manager service (Windows), and on avrd. Stop and restart ltid after creating the directory.
reqlib	Debug information on the processes that request media management services from vmd or EMM. Stop and restart vmd after creating the directory.
robots	Debug information on all robotic daemons, which includes tldcd, tl8cd, and tl4d daemons. Stop and restart robotic daemons.
tpcommand	Debug information for device configuration, including the tpconfig and the tpautoconf commands and the **NetBackup Administration Console**.
vmscd	Debug information for the NetBackup Status Collection daemon. Stop and restart vmscd after creating the directory.

Disable media and device management logs

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

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**, which also affect 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.
  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.

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. The `vm.conf` file is located in the following directory:

Windows	*install_path*\Volmgr\
UNIX	/usr/openv/volmgr/

Limit the size and retention of legacy 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.

Keep logs for days

Limits the time that NetBackup retains NetBackup processes logs (except media and device management logs). The default is 28 days.

DAYS_TO_KEEP_LOGS in vm.conf

Controls log file rotation for media and device management legacy logs. The default is 30 days. The vm.conf file is located in *install_path*\Volmgr\ or /usr/openv/volmgr/.

MAX_LOGFILE_SIZE and MAX_NUM_LOGFILES

With legacy logging, NetBackup uses the configuration file (the Windows registry or the bp.conf file on UNIX) to set the maximum size of a log file. Use the bpsetconfig command to configure the following bp.conf parameters:

- The MAX_LOGFILE_SIZE parameter indicates the maximum size of a log file. When the log file size in NetBackup matches the MAX_LOGFILE_SIZE setting, the next logs are stored in a new log file. The default is 500 MB.
- The MAX_NUM_LOGFILES parameter indicates the maximum number of log files that can be created in NetBackup. When the number of log files matches theMAX_NUM_LOGFILES setting, the older log files are purged. The default is 0 (infinite).

Accessibility of the legacy logs

NetBackup 10.0 sets the permissions on the legacy log directories to a more restrictive but configurable level. This change is designed to prevent unauthorized access to the NetBackup logs, which may contain sensitive information.

You can control the accessibility of the logs by configuring the value of the parameter ALLOW_WORLD_READABLE_LOGS using the nbsetconfig command.

Here are the configurable values:

- If ALLOW_WORLD_READABLE_LOGS=YES, the debug logs have world-readable permissions.
- If ALLOW_WORLD_READABLE_LOGS=NO, which is the default state, the debug logs do not have world-readable permissions.

Note: user_ops (except user_ops/nbjlogs) and dbagents logs are world-readable and non-world-writable.

See the *NetBackup Commands Reference Guide* for details of the nbsetconfig command.

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 clients

1. In the **NetBackup Administration Console**, expand **Host Properties > Clients**.
2. Double-click the client you want to modify.
3. Select the applicable node, either **UNIX Client** or **Windows Client**.
4. Locate the field **Keep status of user-directed backups, archives, and restores for**.
5. Enter the number of days you want to retain the log files and click **OK**.

UNIX logging with syslogd

On UNIX, NetBackup uses `syslogd` to record robotic errors, network errors, and state changes for robotically controlled drives. On HP-UX, the `sysdiag` tool may provide additional information on hardware errors.

To enable this additional logging, use one of the following methods:

- Use the `ltid` command with the `-v` option to start the device management processes. This option starts robotic daemons and `vmd` in verbose mode.
- Use a command and the `-v` option to start a specific daemon (for example, `acsd -v`).

Errors are logged with `LOG_ERR`, warnings with `LOG_WARNING`, and debug information with `LOG_NOTICE`. The facility type is daemon.

Logging options with the Windows Event Viewer

You can configure a NetBackup Windows master server to also write logging application and diagnostic messages to the Windows **Event Viewer** Application Event log.

For details on `vxlogcfg`, see the NetBackup Commands Reference Guide.

To write unified logging messages to the Windows Event Viewer for an originator

1. Use the `vxlogcfg` command to set the `LogToOslog` value to true for the originator.

   For example:

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

2. Restart the NetBackup services.

To write legacy logging messages to the Windows Event Viewer

1. Create the `eventlog` file on the NetBackup master server.

   install_path`\NetBackup\db\config\eventlog`

2. Optionally, add an entry to the `eventlog` file. For example:

   ```
   56 255
   ```

   “56” produces a log with the messages that have a severity of warning, error, and critical (56 = 8 + 16 + 32). “255” produces a log with messages for all types (255 = 1 + 2 + 4 + 8 + 16 + 32 + 64 +128).

3. Restart the NetBackup services.

Event log parameters

The parameters in the `eventlog` represent severity and type. Both parameters are specified as decimal numbers and equate to a bitmap for the values below.

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.

1 = Unknown

2 = Debug

4 = Info

8 = Warning

16 = Error

32 = Critical

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).

1 = Unknown

2 = General

4 = Backup

8 = Archive

16 = Retrieve

32 = Security

64 = Backup Status

128 = Media Device

In the logs, the messages are formatted as follows:

```
<Severity>  <Job type>  <Job ID>  <Job group ID>  <Server>  <Client>  <Process>  <Text>
```

For example:

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

Chapter 2

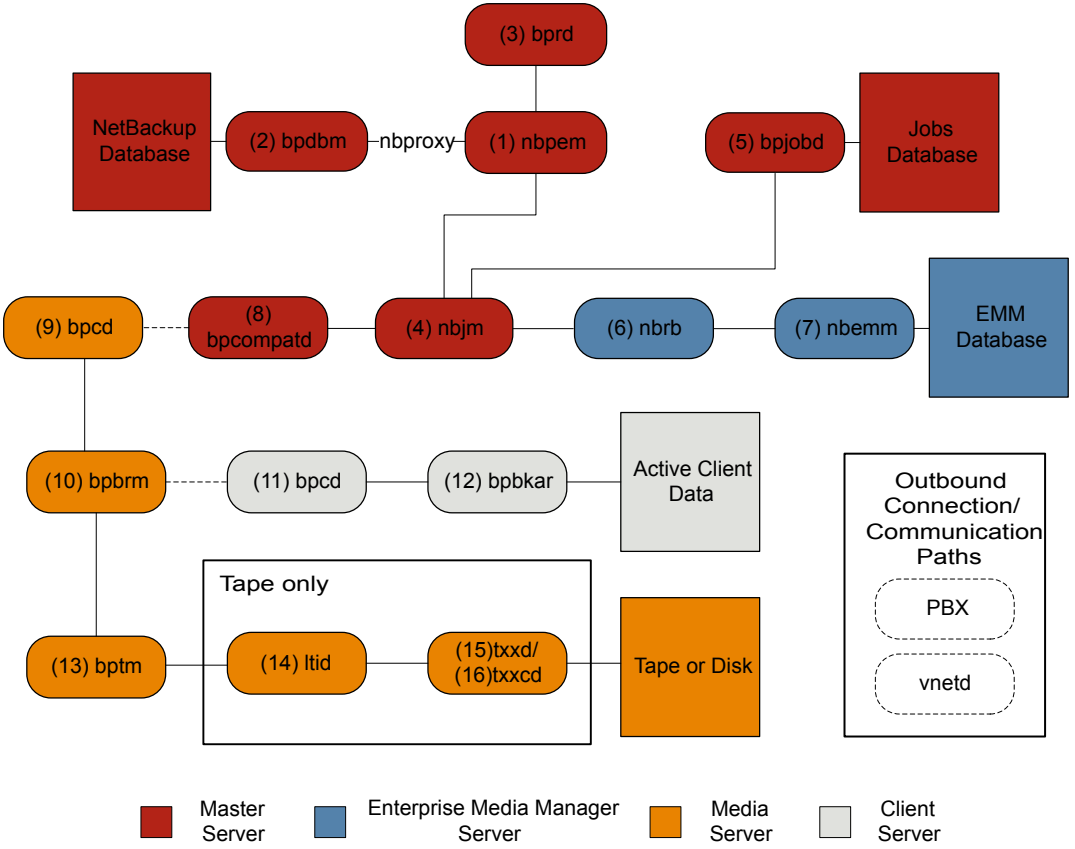
Backup process and logging

This chapter includes the following topics:

Backup process

Figure 2-1 illustrates the backup procedure and the process flow during a scheduled backup.

Figure 2-1 Basic backup process flow



Basic backup procedure

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

   In the case of a user-initiated backup, the backup is started when `nbpem` receives a request from the (3) NetBackup Request Daemon (`bprd`).

2. When the job is due, `nbpem` issues a request to the (4) NetBackup Job Manager (`nbjm`) to submit the backup and get a `jobid`.

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

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

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

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

NetBackup process descriptions

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

See "Backup and restore startup process" on page 53.

See "Backup and archive processes" on page 54.

See "Backups and archives - UNIX clients" on page 55.

See "Multiplexed backup process" on page 55.

Backup and restore startup process

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

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

For more information about SAN client and Fibre Transport startup processes, see the NetBackup SAN Client and Fibre Transport Guide.

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

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

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

Backup and archive processes

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

The job scheduler processes consist of the following:

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

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

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

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

Backups and archives - UNIX clients

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

Each type of backup is started differently as follows:

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

Multiplexed backup process

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

About backup logging

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

See "nbpem logging" on page 155.

See “nbproxy logging” on page 155.

See “bpdbm logging” on page 151.

See “bprd logging” on page 152.

See “nbjm logging” on page 154.

See “bpjobd logging” on page 151.

See “nbrb logging” on page 155.

See “nbemm logging” on page 154.

See “bpcompatd logging” on page 151.

See “PBX logging” on page 157.

See “vnetd logging” on page 159.

See “bpcd logging” on page 151.

See “bpbrm logging” on page 150.

See “bpbkar logging” on page 150.

See “bptm logging” on page 153.

See “ltid logging” on page 153.

See “txxd and txxcd logging” on page 159.

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

See “acsssi logging” on page 149.

See “bpbackup logging” on page 150.

See “daemon logging” on page 153.

See “reqlib logging” on page 158.

See “Robots logging” on page 158.

Sending backup logs to Technical Support

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

See “About backup logging” on page 55.

See “Logs to accompany problem reports for synthetic backups” on page 94.

Note: It is recommended that the diagnostic level for unified logging be set at the default level of 6.

Table 2-1 Logs to gather for specific backup issues

Type of problem	Logs to gather
Problems with backup scheduling	■ The `nbpem` log at debug level 5 ■ The `nbjm` log at debug level 5 ■ The `nbproxy` log at verbose 4 ■ The `bpdbm` log at verbose 2 ■ The `bprd` log at verbose 5 **Note:** The `bprd` log is only needed for problems with manual or user-initiated backups.
Problems with the queued backup jobs that do not go active	■ The `nbpem` log at debug level 3 ■ The `nbjm` log at debug level 5 ■ The `nbrb` log at debug level 4 ■ The `nbproxy` log at verbose 4 ■ The `bpdbm` log at verbose 2 ■ The `nbemm` logs at the default levels ■ The `mds` log at debug level 2 **Note:** The `mds` log writes to the `nbemm` log.
Problems with the active backup jobs that do not write	■ The `nbjm` log at debug level 5 ■ The `nbrb` log at debug level 4 ■ The `bpdbm` log at verbose 2 ■ The `bpbrm` log at verbose 5 ■ The `bptm` log at verbose 5 ■ The `bpcd` log at verbose 5 If the problem is a tape load or unload issue, Support may also need the following logs: ■ The `ltid` log ■ The `reqlib` log ■ The `daemon` log ■ The `robots` log ■ The `acsssi` log (UNIX only)

Chapter 3

Media and device processes and logging

This chapter includes the following topics:

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.

See Figure 3-1 on page 59.

For robotic daemons like `tl8d` and `tlhd`, the associated robot must also be configured for the daemon to run. Additional ways to start and stop daemons are available. You must know the hosts that are involved to start all the daemons for a robot.

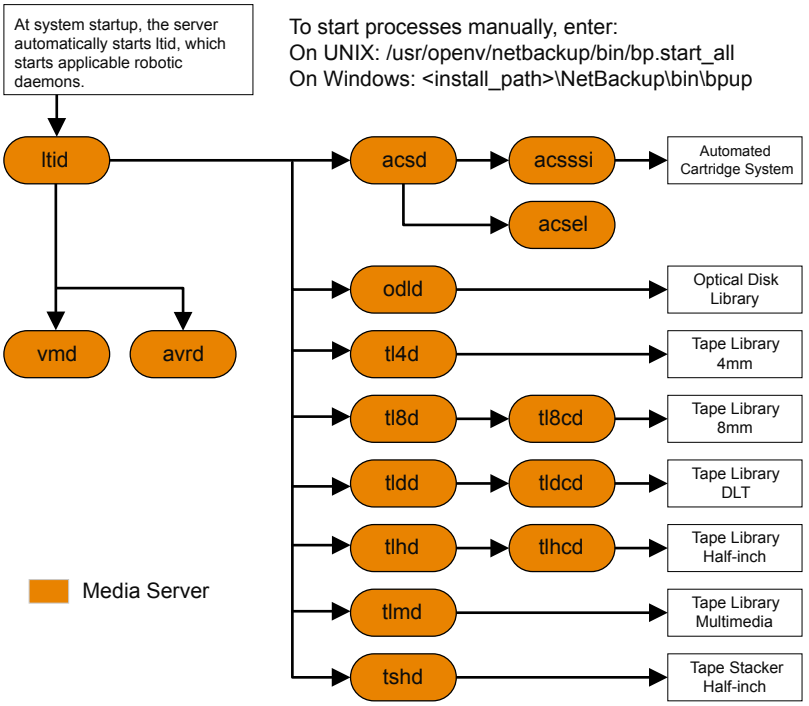
See Table 3-1 on page 66.

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

robotic
: Each host with a robotic drive attached must have a robotic daemon. These daemons provide the interface between `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
: 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.

Figure 3-1 Starting media and device management



Media and device management process

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

See "Backup and archive processes" on page 54.

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 that is shared by itself and `ltid`. Drive busy status also appears in the Device Monitor.

See Figure 3-2 on page 61.

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

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

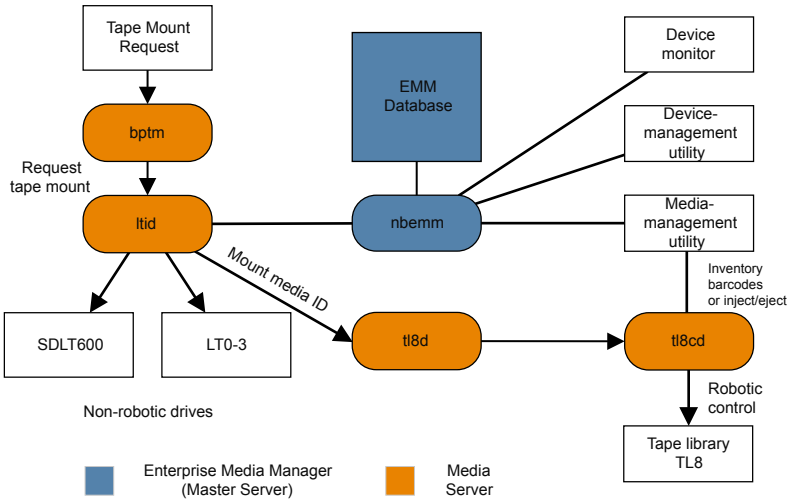
For more information about NetBackup media selection for nonrobotic drives, see the NetBackup Administrator's Guide, Volume II.

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

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

Figure 3-2 shows an example of the media and device management process.

Figure 3-2 Media and device management example process



Shared Storage Option management process

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

For more information about the Shared Storage Option, see the NetBackup Administrator's Guide, Volume II.

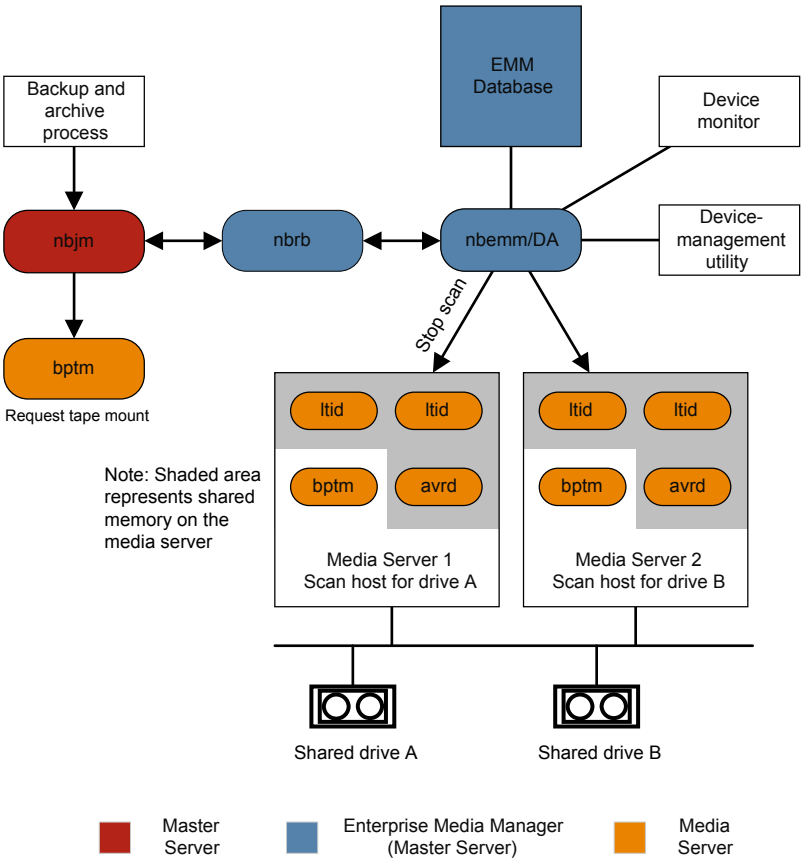
The following shows the Shared Storage Option management process in the order presented:

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

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



Barcode operations

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

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

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

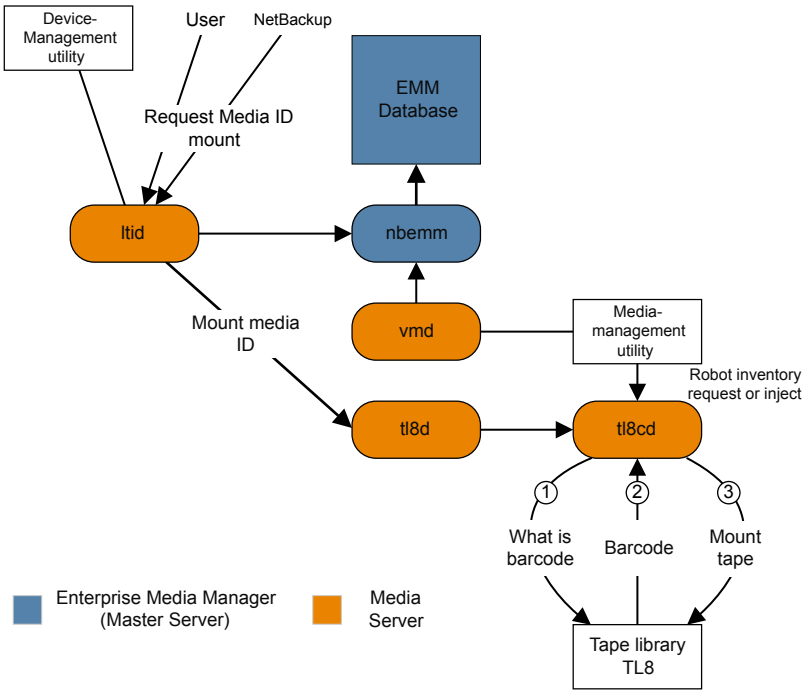
See Figure 3-4 on page 65.

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

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

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

Figure 3-4 Barcode request



Media and device management components

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

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

Figure 3-5 Media and device management directories and files

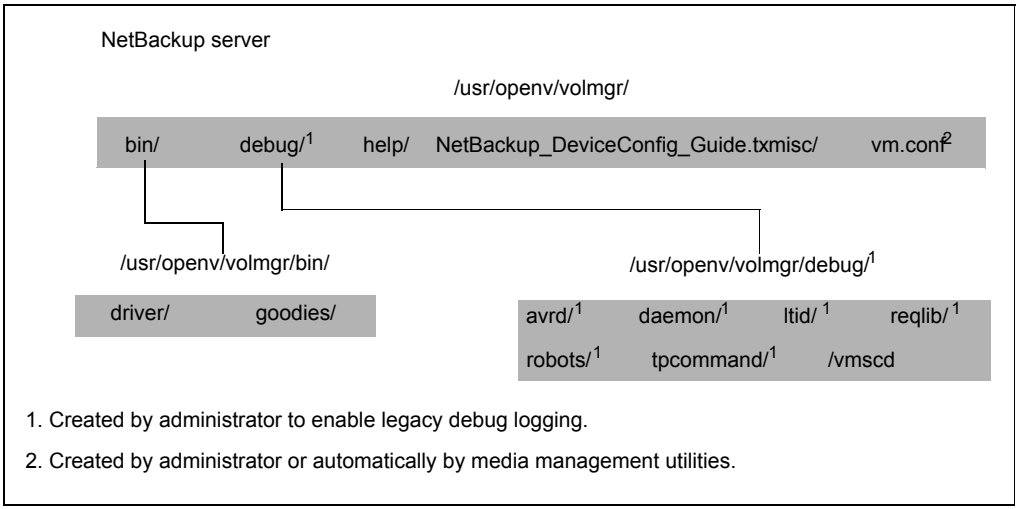


Table 3-1 Media and device management directories and files

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

Table 3-2 describes the media management and device management programs and daemons. The components are located in the following directory:

`/usr/openv/volmgr/bin`

install_path\volmgr\bin.

Note: On UNIX, `syslog` manages the system log (the facility is daemon). On Windows, the Event Viewer manages the system log (the log type is Application).

Table 3-2 Media and device management daemons and programs

Program or daemon	Description
acsd	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). For UNIX, see the acsssi and the 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.
acsssi	Available only on UNIX. See the NetBackup Device Configuration Guide.
avrd	The automatic-volume-recognition daemon controls the automatic volume assignment and label scanning. This daemon lets NetBackup read labeled tape volumes and automatically assigns the associated removable media to the requesting processes. **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.

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

Program or daemon	Description
ltid	The device daemon (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 the `Stop/Restart Device Manager Service` command in the **Media and Device Management** window on Windows. **Stopped By**: `/usr/openv/volmgr/bin/stopltid` command on UNIX or the `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 the `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`).
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 can 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 controls the robot (it can 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 the 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)*

Program or daemon	Description
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/tl8cd` command). **Stopped By**: Stopping `ltid` or by using the `tl8cd -t` command. **Debug Log:** Errors are logged in the system log and the 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`).
tldd	The Tape Library DLT daemon works 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 way 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`).
tldcd	The tape library DLT control daemon provides robotic control for a TLD robot in the same way as explained previously for tl8cd. **Started By**: Starting `ltid` (or on UNIX, independently by using the `/usr/openv/volmgr/bin/tldcd` command). **Stopped By**: Stopping `ltid` or by using the `tldcd -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)*

Program or daemon	Description
tlhd	The Tape Library Half-inch daemon works with tlhcd to handle requests to the TLH robots that are in an IBM Automated Tape Library (ATL). tlhd provides the interface between the local ltid and the robotic control (tlhcd) in the same way as explained previously for tl8d. **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`).
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 the same way as 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 starting independently by using the `/usr/openv/volmgr/bin/tlmd` command. **Stopped By**: Stopping `ltid` or stopping 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.

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

Program or daemon	Description
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/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.
vmd	The Volume Manager daemon (NetBackup Volume Manager service on Windows) allows the remote administration and control of Media and Device Management. **Started By**: Starting `ltid`. **Stopped By**: Terminating the 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 the drives that are attached to the 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)

Chapter 4

Restore process and logging

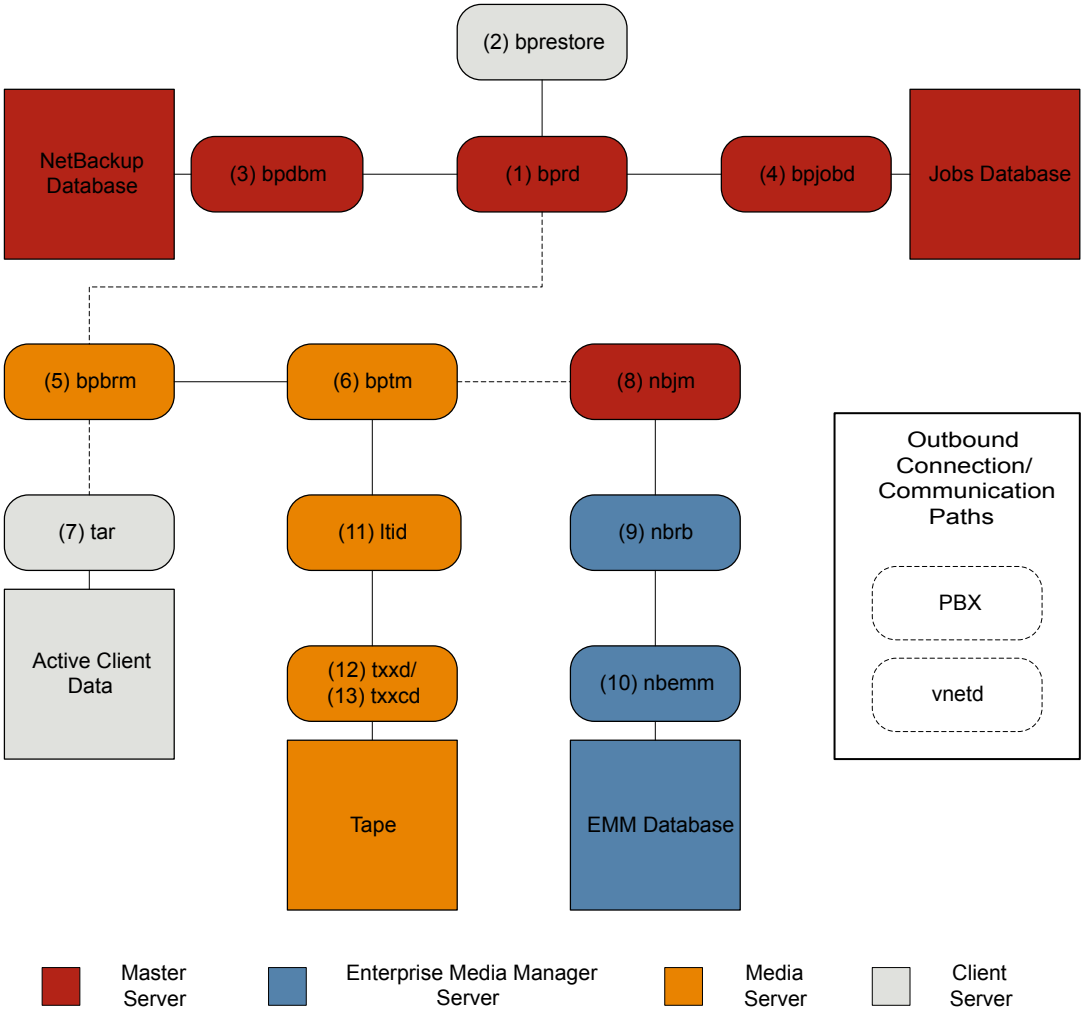
This chapter includes the following topics:

Restore process

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

Figure 4-1 illustrates a restore from tape.

Figure 4-1 Restore from tape process flow



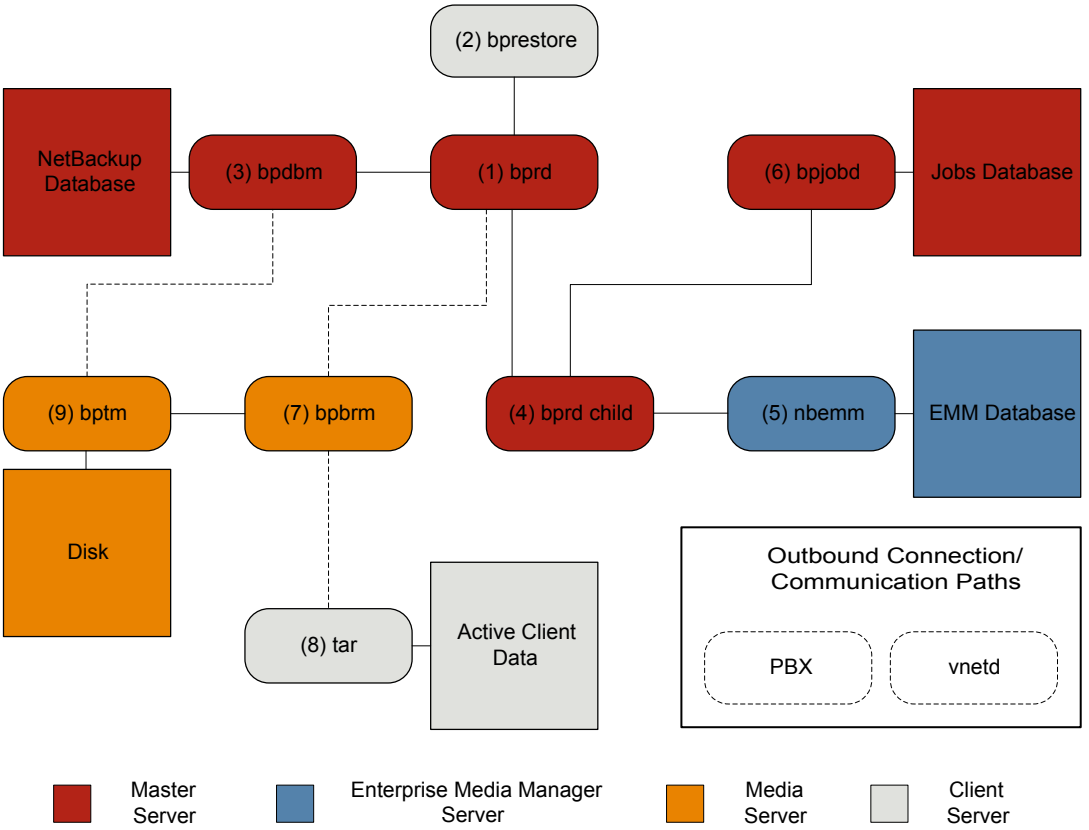
Restore procedure from tape

1. The (1) NetBackup Request Daemon (bprd) receives a restore request. This request can be initiated from the Backup, Archive, and Restore user interface or from the (2) command line (bprestore).
2. The bprd process launches two child processes: MAIN bprd and MPX-MAIN-bprd. The MAIN bprd process is used to identify images and media, while the MPX-MAIN-bprd process manages the restore operation. For simplicity's sake, these three processes are all referred to here as bprd.
3. The bprd service communicates with the (3) NetBackup Database Manager program (bpdbm) to get the information that is required to restore the files that have been requested.
4. Once it has the information it needs, bprd communicates with (4) bpjobd, and the job is added to the job list in the jobs database. The job is now visible in the Activity Monitor. It may show as "Active" even before resources are acquired.
5. The bprd service goes through Private Branch Exchange (PBX) and the NetBackup Legacy Network (vnetd) to start the (5) NetBackup backup and restore manager (bpbrm).
6. The bpbrm service starts the (6) tape management process (bptm) and provides the media information that is required for the restore. It also starts the (7) Tape Archive program (tar) on the client (through PBX and vnetd) and creates a connection between tar and bptm.
7. The bptm process sends a resource request to the (8) NetBackup Job Manager (nbjm) through PBX and vnetd.
8. The nbjm process sends the resource request to the (9) NetBackup Resource Broker (nbrb), which queries the (10) Enterprise Media Manager (nbemm). Once the resources have been allocated, nbrb notifies nbjm, which notifies bptm.
9. The bptm process makes a mount request to the (11) logical tape interface daemon (ltid). The ltid service calls on the (12) robotic drive daemon (t*xx*d, where *xx* varies based on the type of robot being used). The t*xx*d daemon communicates the mount request to the (13) robotic control daemon (t*xx*cd), which mounts the media.
10. The bptm process reads the data to be restored from the media and delivers it to tar.
11. The tar process writes the data to the client disk.
12. When the restore is completed, bptm unmounts the media and notifies nbjm. The job now appears as "Done" in the Activity Monitor.

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

Figure 4-2 illustrates a restore from disk.

Figure 4-2 Restore from disk process flow



Restore procedure from disk

1. The (1) NetBackup Request Daemon (`bprd`) receives a restore request. This request can be initiated from the Backup, Archive, and Restore user interface or from the (2) command line (`bprestore`).
2. The `bprd` 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.

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.

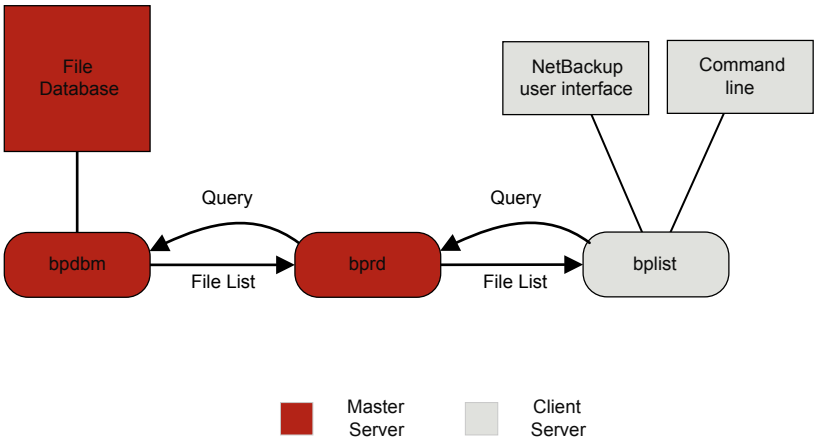
See "About restore logging" on page 79.

UNIX client restore

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

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

Figure 4-3 List operation - UNIX client



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

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

  Note: To restore Backup Exec images, `bpbrm` initiates `mtfrd` instead of `nbtar` 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 `nbtar` 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), nbtar reads the data directly from shared memory. If the server restores a client that resides on a different host, it creates a child bptm process which transmits the data to nbtar on the client.

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

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

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

Windows client restore

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

The following are the Windows processes involved in restore operations:

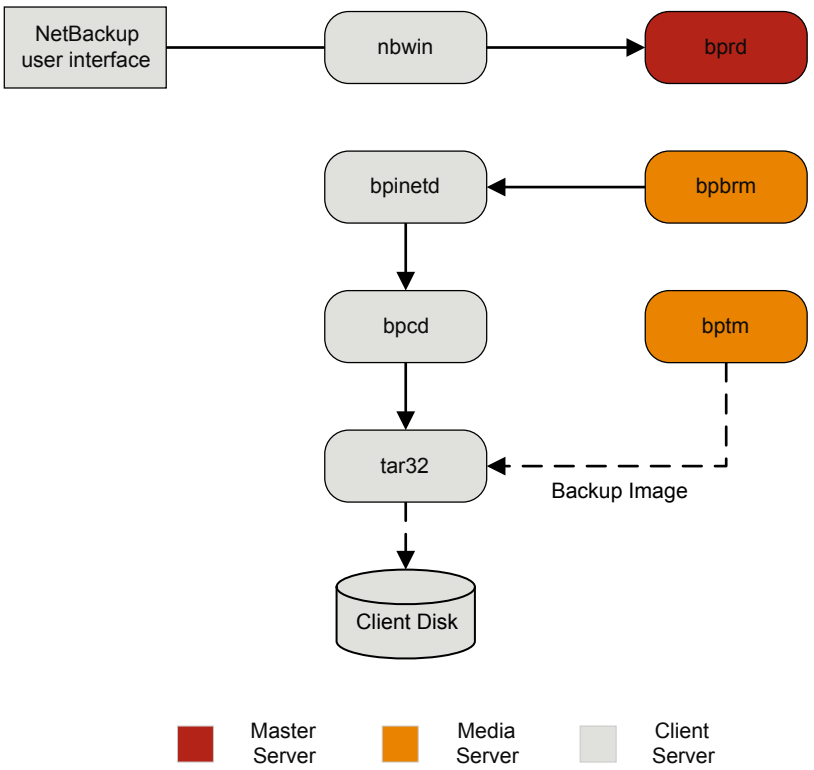
- NBWIN is the user interface program on the client. The bpbackup function and the bparchive function are merged into NBWIN.
- BPINETD serves the same purpose as inetd on UNIX clients.
- The NetBackup client daemon is called BPCD.
- TAR32 is part of NetBackup for Windows and serves the same purpose as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

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

The server processes are the same as described for UNIX.

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

Figure 4-4 Restore - Windows client



About restore logging

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

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

See “Sending restore logs to Technical Support” on page 80.

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

See “bprd logging” on page 152.

See “bprestore logging” on page 152.

See “PBX logging” on page 157.

See “vnetd logging” on page 159.

See “bpdbm logging” on page 151.

See “bpjobd logging” on page 151.

See “bpbrm logging” on page 150.

See “bptm logging” on page 153.

See “tar logging” on page 159.

See “nbjm logging” on page 154.

See “nbrb logging” on page 155.

See “nbemm logging” on page 154.

See “ltid logging” on page 153.

See “reqlib logging” on page 158.

See “Robots logging” on page 158.

See “acsssi logging” on page 149.

Sending restore logs to Technical Support

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

See “Logs to accompany problem reports for synthetic backups” on page 94.

Note: It is recommended that the diagnostic level for unified logging be set at the default level of 6.

Table 4-1 Log to gather for specific restore issues

Type of problem	Log to gather
Problems with restore jobs from tape	■ The `nbjm` log at debug level 5 ■ The `nbemm` log at debug level 1 ■ The `nbrb` log at debug level 4 ■ The `bpdbm` log at verbose 1 ■ The `bprd` log at verbose 5 ■ The `bpbrm` log at verbose 5 ■ The `tar` log at verbose 5 ■ The `bpcd` log at verbose 5 If the problem is a media or a drive issue, Support may also need the following logs: ■ The `reqlib` log ■ The `daemon` log ■ The `robots` log ■ The `acsssi` log (UNIX only)
Problems with restore jobs from disk	■ The `bpdbm` log at verbose 1 ■ The `bprd` log at verbose 5 ■ The `bpbrm` log at verbose 5 ■ The `bptm` log at verbose 5 ■ The `bpdm` log at verbose 5 ■ The `tar` log at verbose 5 ■ The `bpcd` log at verbose 5

Chapter 5

Advanced backup and restore features

This chapter includes the following topics:

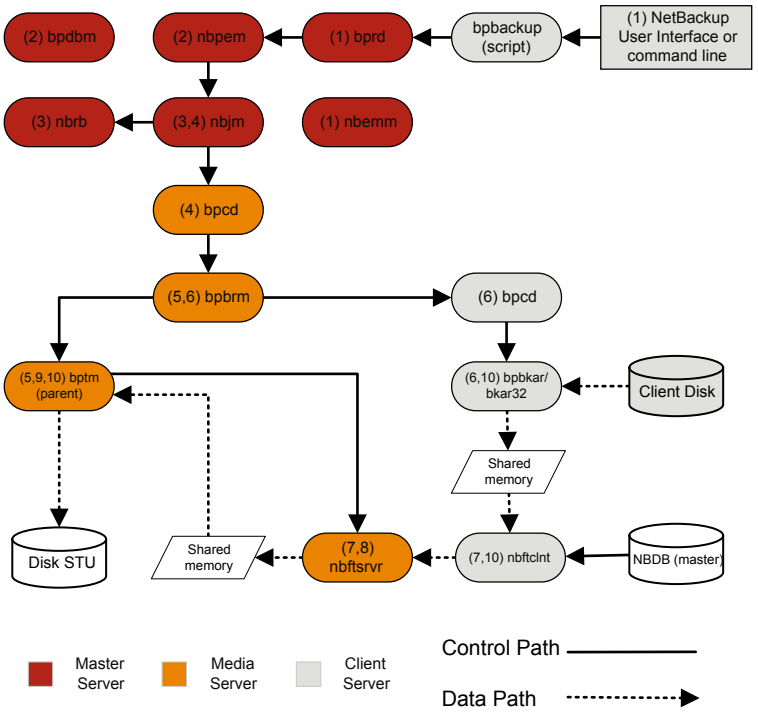
SAN Client Fiber Transport backup

The following shows a SAN client backup process.

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

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

Figure 5-1 SAN client backup process flow



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

SAN client backup procedure

1. The NetBackup master server or primary client initiates the backup. The NetBackup Request Daemon (`bprd`) submits a backup request to the NetBackup 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.

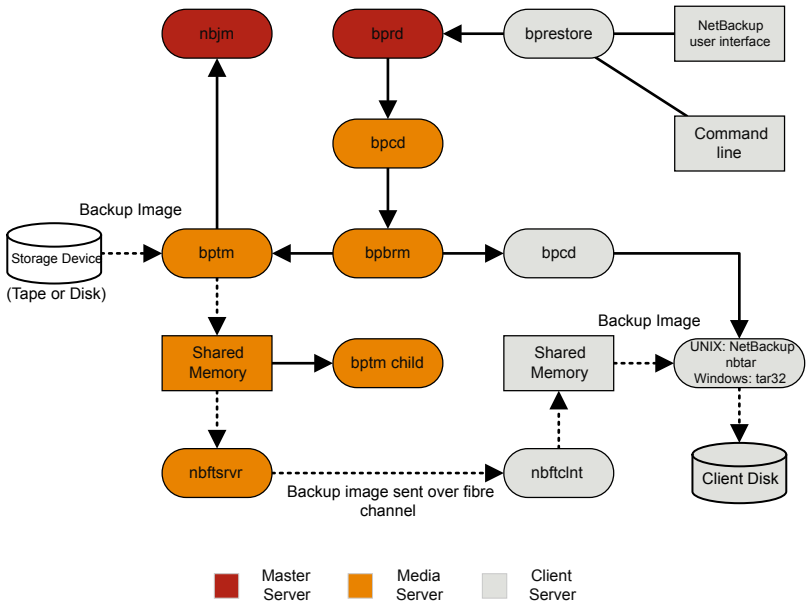
- Sorts the work list in order of due time.
- Submits to nbjm all jobs that are currently due.
- Sets a wake-up timer for the next due job.
- When the job finishes, it recomputes the due time of the next job and submits to nbjm all of the jobs that are currently due.

3. The Job Manager service (nbjm) requests backup resources from the Resource Broker (nbrb), that returns information on the use of shared memory for the SAN client.
4. The nbjm service starts the backup by means of the client daemon bpcd, which starts the backup and restore manager bpbrm.
5. The bpbrm service starts bptm, which does the following:
   - Requests the SAN client information from nbjm.
   - Sends a backup request to the FT server process (nbftsrvr).
   - Sends a backup request to the FT client process on the client (nbftclnt), that does the following: Opens a Fibre Channel connection to nbftsrvr on the media server, allocates the shared memory, and writes the shared memory information to the backup ID file.
6. The bpbrm service uses bpcd to start bpbkar, that does the following:
   - Reads the shared memory information from the BID file (waits for the file to exist and become valid).
   - Sends the information about files in the image to bpbrm.
   - Writes the file data to bpbkar, optionally compresses it, then writes the data to the shared buffer.
   - Sets the buffer flag when the buffer is full or the job is done.
7. The FT client process (nbftclnt) waits for the shared memory buffer flag to be set. It then transfers the image data to the FT Server (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

Figure 5-2 SAN client restore with Fibre Transport



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

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

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

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

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

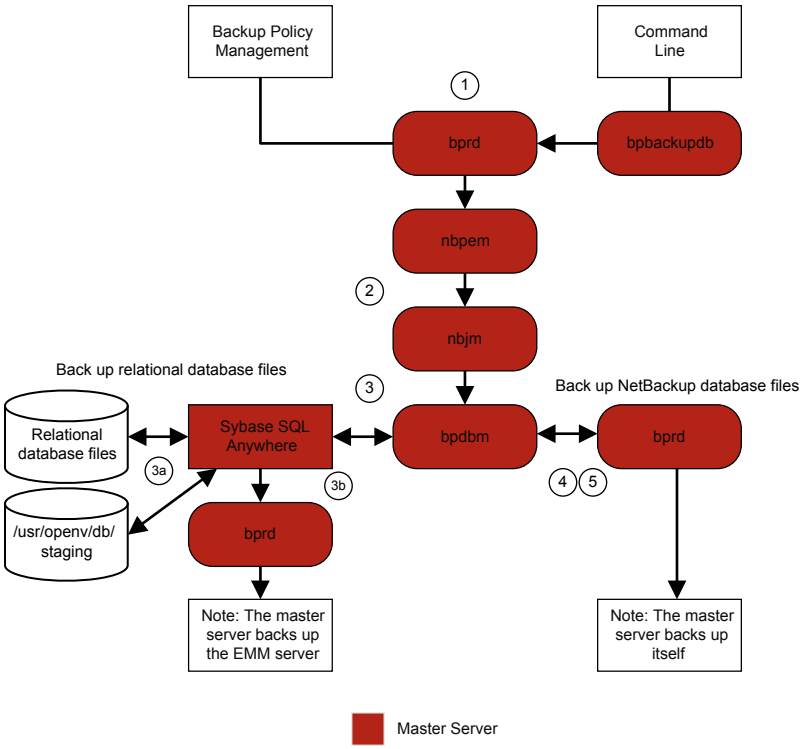
Hot catalog backup

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

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

Figure 5-3 shows the hot catalog backup process.

Figure 5-3 Hot catalog backup process



NetBackup initiates the following hot catalog backup jobs:

- A parent job that is started manually by the administrator or by a catalog backup policy schedule.
- A child job that creates the `.drpkg` file for use when recovering the identity of the master server. After successful creation of the `.drpkg` file and before staging, the same child job will run an online backup of the SQL Anywhere database files to the staging directory located at:
  UNIX: `/usr/openv/db/staging`
  Windows: *`install_path`*`\Veritas\NetBackupDB\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`, `bpbrm`, `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`

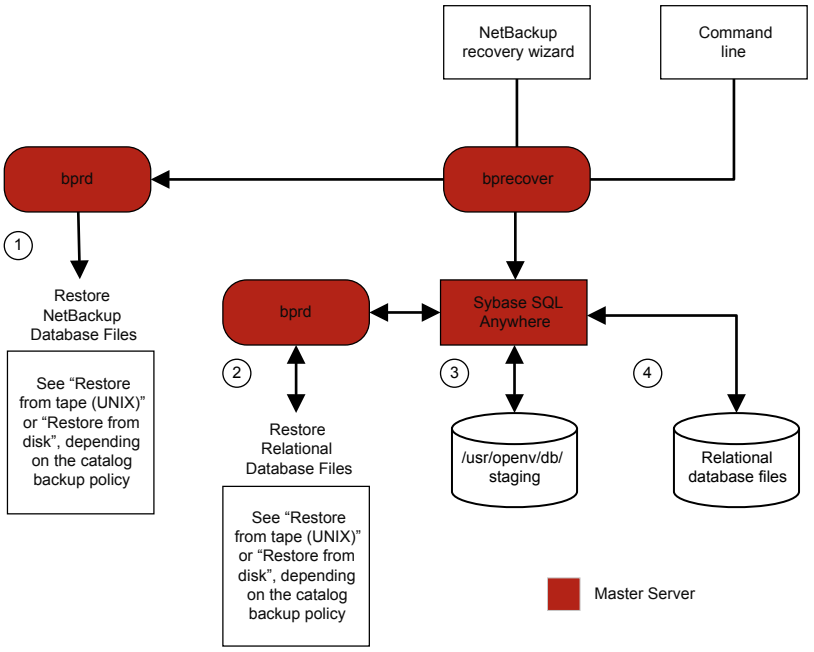
Hot catalog restore

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

Note: Before you run a hot catalog restore in a disaster recovery situation, the identity of the master server should be recovered either by the disaster recovery installation or the `nbhostidentity -import -infile` *`drpkg.path`* command. Once the identity is recovered, the hot catalog recovery can be completed as usual.

Figure 5-4 illustrates the catalog restore and recovery process.

Figure 5-4 Catalog restore and recovery



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

- The NetBackup catalog image and configuration files are restored.
- The NBDB files are restored. The database files are restored to `/usr/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 previously created backup images called component images.

Note: Synthetic archives do not exist.

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

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

`bpsynth` makes a synthetic image in several phases:

Table 5-1

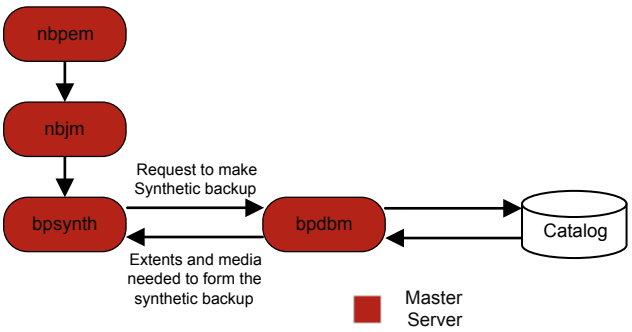
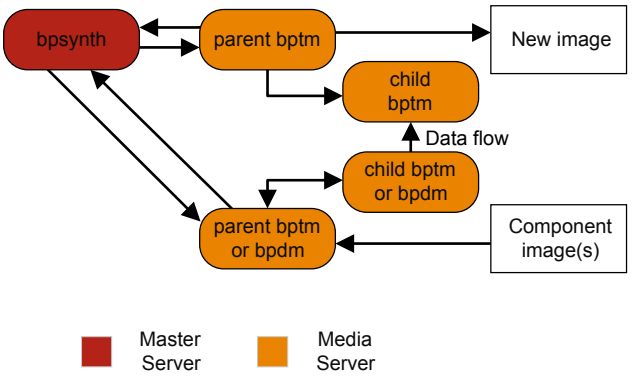
Phase	Description
1 - Prepare catalog information and extents	In phase 1, `bpsynth` makes a synthetic backup request to the database manager, `bpdbm`. It uses the entries and the TIR information from the catalogs of the component images to build the catalog for the new synthetic image. It also builds 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: 
2 - Obtain resources	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.

Table 5-1 *(continued)*

Phase	Description
3 - Copy data	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:  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. The `bpsynth` process sends the extents (starting block and count) for each component image to the corresponding child `bptm` or `bpdm` reader process. 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. 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.
4 - Validate the image	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. 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.

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 the following log types to Veritas Technical Support.

- Log files that unified logging creates
  See "Gathering unified logs for NetBackup" on page 15.
- Log files that legacy logging creates
  See "Creating legacy log directories to accompany problem reports for synthetic backup" on page 94.
- 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`

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.

See "Logs to accompany problem reports for synthetic backups" on page 94.

Table 5-2 Creating legacy log directories

Step	Action	Description
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
Step 2	Create directories on the media server.	Create the following directories: *install_path*/netbackup/logs/bpcd *install_path*/netbackup/logs/bptm
Step 3	Change the **Global logging level**.	In **Host Properties**, select a master server and set the **Global logging level** to 5. See the NetBackup Troubleshooting Guide for more information on how to use the host properties to access configuration settings.
Step 4	Rerun the job.	Rerun the job and gather the logs from the directories that you created. The `bptm` logs are required only if the images are read from or written to a tape device or disk. The `bpdm` logs are needed only if the images are read from disk. If the images are read from multiple media servers, the debug logs for `bptm` or `bpdm` must be collected from each media server.

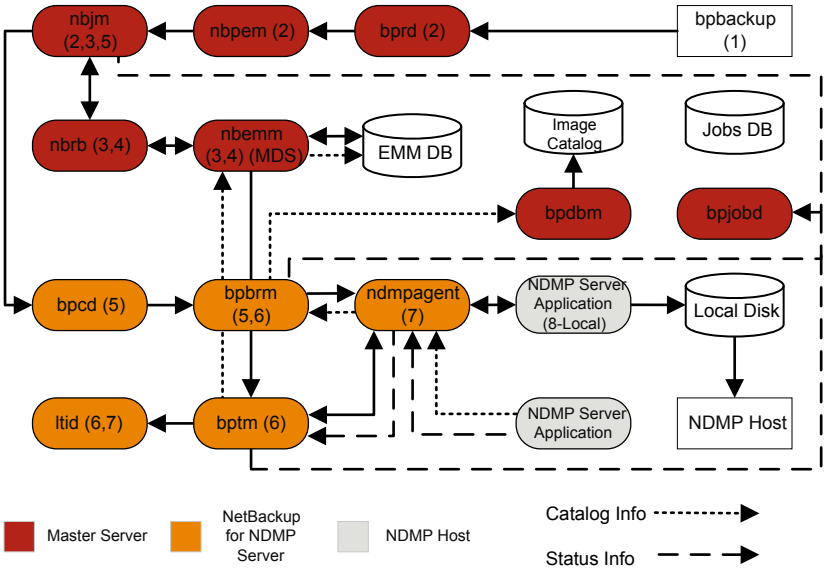
Chapter 6

Storage logging

This chapter includes the following topics:

NDMP backup logging

Figure 6-1 NDMP backup process



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

NDMP backup procedure

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

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

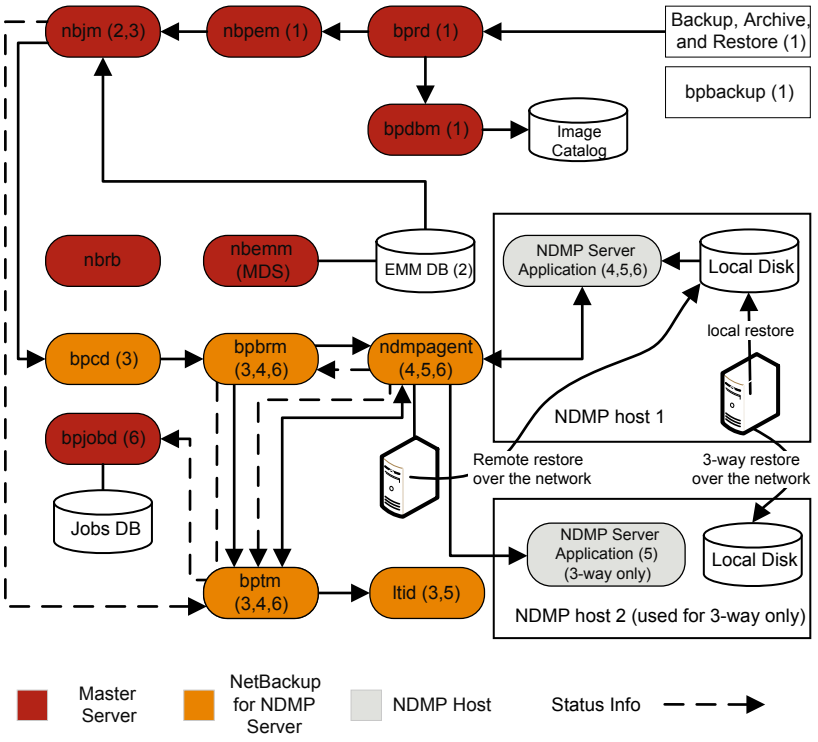
- Remote backup (not shown in the process flow diagram). The device that is used to write the backup is associated with a NetBackup storage unit. `bptm` on the NetBackup media server mounts a tape on a tape drive. NetBackup sends the NDMP commands to the NDMP server to initiate the backup to the non-NDMP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

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

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

NDMP restore logging

Figure 6-2 NDMP restore process



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

NDMP restore procedure

1. An administrator at the NetBackup Administration Console on a NetBackup master server or media server initiates a restore job by browsing the images catalog and by selecting the files to be restored from the NDMP images. This process is similar to selecting files to be restored from standard backup images. The NetBackup master server identifies the specific media that is required to perform the restore. In this diagram, the media is a tape volume.
2. After the master server identifies the data to be restored and the media required, it submits a restore job. The Job Manager (`nbjm`) then requests the required resources. This resource request causes the allocation of the media that contains the data to be restored. In this example, a tape drive is used during the restore operation.
3. The master server contacts the media server that participates in the restore job, and starts the Restore Manager (`bpbrm`) process to manage the restore job. `bpbrm` starts the Tape Manager process (`bptm`), that queries `nbjm` for the tape volume. Then, `bptm` requests that the logical tape interface daemon (`ltid`) mounts the tape.
4. On the NetBackup for NDMP server, the NDMP agent (`ndmpagent`) connects to the filer and issues NDMP commands to mount the tape that is directly attached, and `ltid` sends NDMP commands to mount the requested tape on the storage device. Or, the media server itself issues tape mount requests much like a regular media manager storage unit.
5. One of the following occurs, depending on the type of NDMP restore operation:
   - Local restore. NetBackup sends the NDMP commands to the NDMP server to initiate the restore operation from a tape drive to a local disk. The restore data travels from a tape drive to a local disk on the NDMP host without traversing the LAN.
   - Three-way restore. The NetBackup media server establishes NDMP communications with both of the NDMP servers that are involved in the restore. To initiate the restore of data from tape on one NDMP server to disk storage on the other NDMP server, the media server sends NDMP commands to both NDMP servers. The restore data travels over the network between the NDMP hosts.
   - Remote restore. NetBackup sends the NDMP commands to the NDMP server to prepare the server for the restore. `bptm` on the media server reads

the restore data from tape and sends it over the network to the NDMP host where the data is written to disk storage.

6 The NDMP server sends status information about the restore operation to the NetBackup for NDMP server. Various NetBackup processes (`nbjm`, `bpbrm`, `bptm`, and others) send job status information to the master server. The Jobs Database Manager (`bpjobd`) process on the master server updates the restore job status in the jobs database. You can view this status in the Activity Monitor.

Chapter 7

NetBackup Deduplication logging

This chapter includes the following topics:

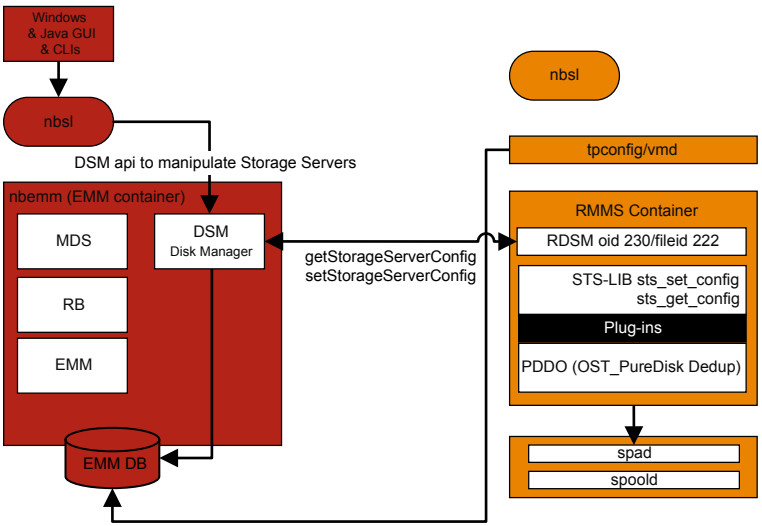
Deduplication backup process to the Media Server Deduplication Pool (MSDP)

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

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

- `spoold` writes `tlogs` to the `.tlog` files in the queue (*`dedupe_path`*`\queue`) directory and to the processed directory. The `tlog` data from the queue directory is processed into the `crdb` later when the next content router queue processing job runs. Beginning with NetBackup 7.6, `.tlog` files do not contain additions to the database.

Figure 7-1 Deduplication configuration for MSDP



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

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

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

```
BPTM_VERBOSE = 5
```

- Edit the `pd.conf` configuration file that is located at:
  Windows:
  install_path`\NetBackup\bin\ost-plugins\pd.conf`
  UNIX/Linux:
  `/usr/openv/lib/ost-plugins/pd.conf`
  and uncomment and/or modify the following line:

  ```
  LOGLEVEL = 10
  ```

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

2. Enable verbose `spad`/`spoold` logging (optional).
   - Edit the *dedup_path*`\etc\puredisk\spa.cfg` and *dedup_path*`\etc\puredisk\contentrouter.cfg` files so that the following line:
     `Logging=long,thread` is changed to `Logging=full,thread`
   - Ensure that you are on the correct media server and restart the MSDP storage server services.

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

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

- Windows command line:

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

- UNIX/Linux command line:

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

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

   Windows:

dedup_path\log\spoold*mediasvr_IP_or_hostname*\bptm\Receive*MMDDYY*.log
dedup_path\log\spoold*mediasvr_IP_or_hostname*\bptm\Store*MMDDYY*.log

   UNIX/Linux:

dedup_path/log/spoold/*mediasvr_IP_or_hostname*/bptm/Receive/*MMDDYY*.log
dedup_path/log/spoold/*mediasvr_IP_or_hostname*/bptm/Store/*MMDDYY*.log

Client deduplication logging

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

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

  install_path\NetBackup\logs\nbostpxy

  UNIX/Linux:

  /usr/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 Administration Console for Windows wizard logging:

1. `wingui` (OID: 263):

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

2. On the applicable MSDP storage pool, edit *`install_path`*`\etc\puredisk\spa.cfg` and *`install_path`*`\etc\puredisk\contentrouter.cfg`. Specify **Logging=full,thread** and then restart the spad and spoold services for the changes to take effect.

   - `nbsl` (OID: 132):

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

   - `dsm` (OID: 178):

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

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

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

4. Remote Monitoring & Management Service:

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

5. `tpcommand` (`...\volmgr\debug\tpcommand`)
6. *`storage_directory`*`\log\msdp-config.log`

Command-line configuration logging:

- Administration log for `nbdevquery` (add *`storage_server`*)
- `tpcommand` log for `tpconfig` (add credentials) (`...\volmgr\debug\tpcommand`)
- *`storage_directory`*`\log\pdde-config.log`
- Storage service (turn on STS logging, to log the msdp/pdplugin responses to NetBackup):

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

- Remote Monitoring and Management Service:

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

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

NetBackup Administration Console logging:

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

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

The logs are located under `C:\Program Files\VERITAS\NetBackup\logs\user_ops\nbjlogs` (Windows) or `/opt/openv/netbackup/logs/user_ops/nbjlogs` (UNIX/Linux). Ensure that you look at the most recent log.

- Storage service (turn on STS logging, to log the `msdp/pdplugin` responses to NetBackup):
  ```
  # vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6
  ```
- Remote Monitoring and Management Service:
  ```
  # vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6
  ```
- `tpcommand` (`...\volmgr\debug\tpcommand`)
- *storage_directory*\log\msdp-config.log

Universal share logs

The following are the universal share configuration logs.

On the storage server:

- `/var/log/vpfs/ia_byo_precheck.log`
  Instant access build-your-own (BYO) pre-condition checking results
- `/var/log/vpfs/vpfs-config.log`
  Velocity Provisioning File System (VPFS) configuration log
- `/var/log/vpfs/spws/spws.log`
  Storage Platform Web Service (spws) log
- `/var/log/vpfs/spws_backend/spws_backend.log`
  Storage Platform Web Service (spws) spws_backend log

On the master server:

- `/usr/openv/logs/nbwebservice/`
  NetBackup Web Services (nbwmc) log

Media server deduplication/pdplugin logging

This topic describes the media server deduplication/pdplugin logging.

- Unless you are troubleshooting the Private Branch Exchange (PBX) communication between the client direct and its media server, reduce the unnecessary CORBA/TAO to zero (0) for deduplication logging by using the following command:
  `# vxlogcfg -a -p NB -o 156 -s DebugLevel=0 -s DiagnosticLevel=0`

For backups:

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

For duplications or replications:

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

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

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

Disk monitoring logging

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

- Storage service (turn on the STS logging to show the response that NetBackup receives when it runs the MSDP plug-in):
  `# vxlogcfg -a -p 51216 -o 202 -s DebugLevel=6 -s DiagnosticLevel=6`
- Remote Monitoring and Management Service:`# vxlogcfg -a -p 51216 -o 222 -s DebugLevel=6 -s DiagnosticLevel=6`

Logging keywords

Support uses the following keywords when it reviews the logs.

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

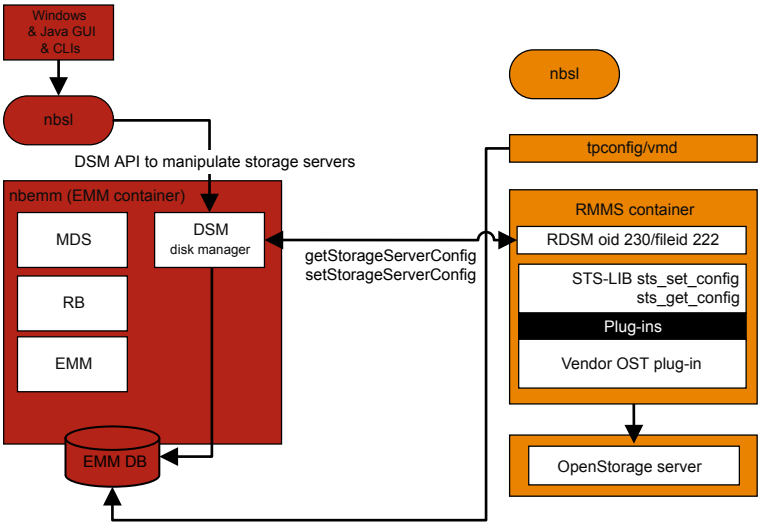
Chapter 8

OpenStorage Technology (OST) logging

This chapter includes the following topics:

OpenStorage Technology (OST) backup logging

Figure 8-1 OST configuration



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

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

1. In the registry or `bp.conf` file, set `VERBOSE = 5`.
2. Ensure that the following directories exist under `/usr/openv/netbackup/logs` (for Windows, use *`install_path`*`\NetBackup\logs`):
   - `bptm`
   - `bpbrm`
   - `bpstsinfo`
3. Create the `volmgr/debug/tpcommand` directory.
4. Put VERBOSE in the `vm.conf` file.

   See "How to control the amount of information written to legacy logging files" on page 45.

5. Set `DebugLevel=6` and `DiagnosticLevel=6` for the following processes:
   - OID 178 (Disk Manager Service, dsm)
   - OID 202 (Storage service, stssvc)
   - OID 220 (Disk Polling Service, dps)
   - OID 221 (Media Performance Monitor Service)
   - OID 222 (Remote Monitoring & Management Service)
   - OID 230 (Remote Disk Manager Service, rdsm)
   - OID 395 (STS Event Manager, stsem)

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

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

   - UNIX/Linux command line:

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

10. Gather the vendor specific plug-in logs that cover the dates from when the backup started and when it failed.

OpenStorage Technology (OST) configuration and management

The OpenStorage Technology (OST) technology uses a plug-in architecture, similar to a software driver, that lets the third-party vendors direct the NetBackup data streams and metadata into their devices. The plug-in is developed and created by the OST partner and it resides on the media server for use by NetBackup. NetBackup depends on the OST plug-in for a path to the storage server.

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

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

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

The basic steps include the following:

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

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

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

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

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

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

- UNIX/Linux: `/usr/openv/netbackup/bin/admincmd/bpstsinfo -li -storage_server` *`storage server name`* `-stype` *`OST_TYPE`*
- Windows: *`install dir`*`\netbackup\bin\admincmd\bpstsinfo -li -storage_server` *`storage server name`* `-stype` *`OST_TYPE`*

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

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

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

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

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

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

Review the flags in the diskpool information, for example:

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

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

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

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

- `nbdevconfig -changests -storage_server` *`storage server name`* `-stype` *`OST_TYPE`* `-setattribute OptimizedImage`

- `nbdevconfig -changedp -stype` *`OST_TYPE`* `-dp` *`diskpool name`* `-setattribute OptimizedImage`

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

- `OptimizedImage` - supports accelerator

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

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

Chapter 9

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

This chapter includes the following topics:

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

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

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

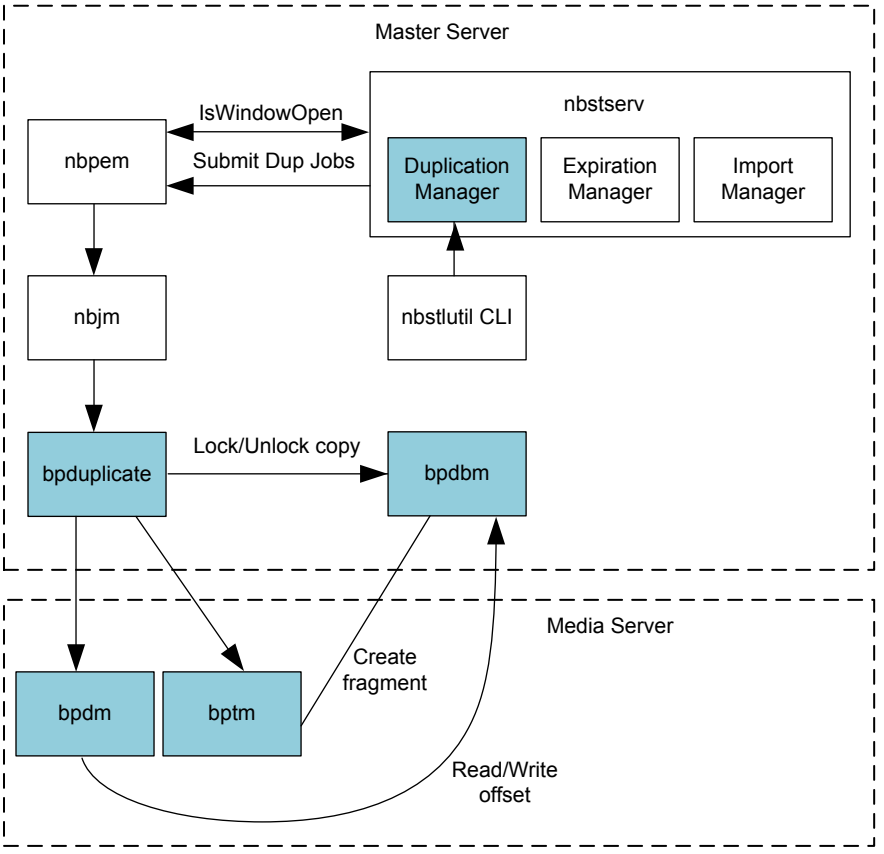
Understanding the storage lifecycle policy (SLP) operations (for example, backup, duplication, replication, import, and snapshot) can help determine which logs can be used to troubleshoot an issue. This topic primarily focuses on the Automatic Image Replication (A.I.R.) and duplication process flows.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

See the NetBackup Administrator's Guide, Volume I for more information about SLPs and A.I.R.

Storage lifecycle policy (SLP) duplication process flow

The following figure describes the SLP duplication process flow.

Figure 9-1 SLP duplication process flow



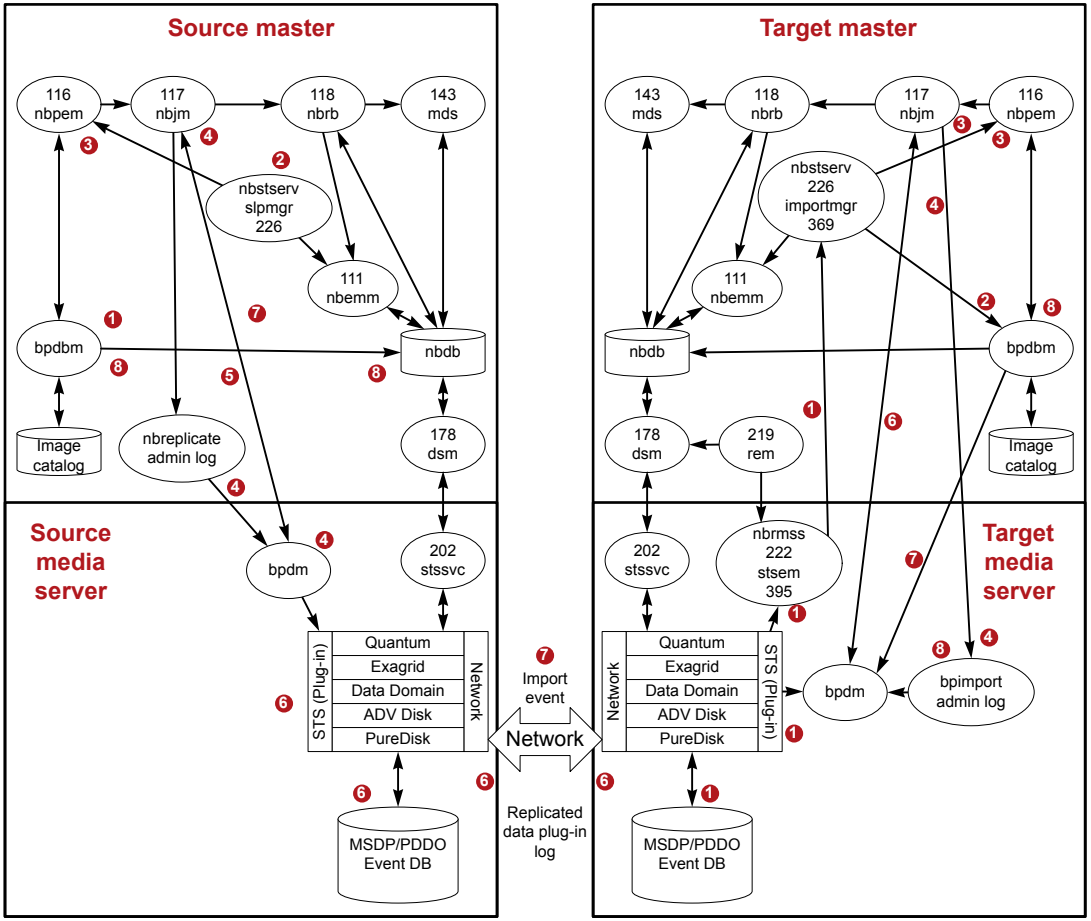
The SLP duplication process flow is as follows:

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

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

The following figure shows the Automatic Image Replication (A.I.R.) process flow.

Figure 9-2 Automatic Image Replication (A.I.R.) process flow



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

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

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

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

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

Import process flow

The import process flow is as follows:

1. The media server that is responsible for monitoring the disk storage polls the storage for the A.I.R. import events. It is the `nbrmms` process that does the polling. The image associated with the import event is sent to the import manager (running within `nbstserv`) on the master server.
2. The import manager (OID 369) inserts the image records into the NBDB database.

3. On a regular interval, nbstserv looks for images that need to be imported. It batches up the images to be imported and sends the request to nbpem. nbpem passes the job to nbjm and then checks for resources from nbrb and nbemm.
4. nbjm starts bpimport. For replicated images, a fast import is run since most of the information that NetBackup needs for the image was brought in when the import event was received.
5. bpimport (admin log) starts bpdm on the media server.
6. bpdm obtains the physical resources needed from nbjm.
7. bpdm reads the image information and sends it to bpdbm on the master server.
8. The import of the image completes and is validated by bpdbm.

SLP and A.I.R. logging

nbstserv (master server):

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

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

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

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

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

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

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

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

```
bpdm (verbose 5)
```

```
bptm (verbose 5)
```

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

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

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

SLP configuration and management

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

```
nbstl –L –all_versions
```

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

```
nbstlutil list –image_incomplete
```

To display the SLP backlog, use the following command:

```
nbstlutil report
```

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

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

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

```
nbstlutil repllist
```

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

```
nbstlutil pendimplist
```

Chapter 10

NetBackup secure communication logging

This chapter includes the following topics:

About NetBackup secure communication logging

NetBackup logs information for secure communication of control-type functions between NetBackup 8.1 and later hosts. These functions include command execution

and the starting processes that are required to initiate a backup or restore. Currently, these processes do not include the `bpbkar` or `tar` data transfer. The hosts must have a Certificate Authority (CA) certificate and a host ID-based certificate for successful communication. NetBackup uses the Transport Layer Security (TLS) protocol for host communication where each host needs to present its security certificate and validate the peer host's certificate against the Certificate Authority (CA) certificate.

The master server acts as the CA. The master server depends on the correct installation and configuration of services, such as `pbx`, `nbatd` and `nbwmc`, to deploy the certificates.

In NetBackup 8.1, certificates are deployed to all the media servers and the clients when they are upgraded. If certificate deployment fails, backups and restores cannot occur. Deployment fails if the following occurs:

- The `pbx`, `nbatd`, or `nbwmc` processes are not running on the master server.
- A host cannot retrieve both the CA certificate and the host ID-based certificate from the master server during the installation or upgrade.

When you diagnose issues with secure communication and certificates, the services or processes that run on the master server are typically involved. After verifying that the services are running and are at the expected NetBackup version, the log files can help determine the issue.

For more details about NetBackup secure communications, see the *Read This First for Secure Communications* document at the following URL:

https://www.veritas.com/docs/DOC5332

Note: If you have NetBackup 8.0 hosts in your environment, you can enable insecure communication with the **Enable insecure communication with NetBackup 8.0 and earlier hosts** option.

Tomcat logging

The Tomcat log files are as follows (on the master server only):

UNIX: `/usr/openv/wmc/webserver/logs`

Windows: *`install path`*`\netbackup\wmc\webserver\logs`

You cannot adjust the verbosity for Tomcat log files.

The Tomcat directories contain log files such as `catalina.log`, `nbwmc.log`, and other logs that are critical to troubleshoot Tomcat issues. This directory can also contain Tomcat Java heap dumps that end with `.hprof` or Java dumps that have

file names that start with `hs_err`. If these files are seen in conjunction with issues with the startup or crashes of Tomcat or `nbwmc`, the files from the affected time frame should also be collected.

NetBackup web services logging

The NetBackup web services logs are as follows (on the master server only):

UNIX: `/usr/openv/logs/nbwebservice`

Windows: *`install path`*`\netbackup\logs\nbwebservice`

This log directory contains the web services originator log files. They include, but are not limited to, the following log files:

Table 10-1 Web services OIDs and log files

Originator ID	Log file	Description
439	`nbwebservice\nbwebservice`	NetBackup Web Service
466	`nbwebservice\security`	NetBackup Security Service (security web app)
482	`nbwebservice\hosts`	NetBackup Hosts Webservice (hosts web app)
483	`nbwebservice\nbconfigmgmt`	NetBackup Configuration Management Service (web app)
484	`nbwebservice\nbgateway`	NetBackup Gateway Service (web app)
485	`nbwebservice\nbwss`	NetBackup WebSocket Service (NBWSS) (web app)
487	`nbwebservice\nbcatalogws`	NetBackup Catalog Web Service (web app)
488	`nbwebservice\nbrbac`	NetBackup Role-based Access Control (RBAC) Web Service (web app)
489	`nbwebservice\nbadminws`	NetBackup Admin Web Service (web app)

The logging for processes with originator IDs (OIDs) can be increased and decreased using the `vxlogcfg` command located in `NetBackup\bin`. This command can be used to add and remove logging for each of the previous processes. See the following examples that use OID 439:

To add logging, use the following command with the `-a` (add) option:

```
vxlogcfg -a -p NB -o 439 -s DebugLevel=6
```

To remove logging, use the following command with the `-r` (remove) option:

```
vxlogcfg -r -p NB -o 439 -s DebugLevel=6
```

If an issue can be easily or quickly reproduced, it can be easier to configure the default log file setting to 6, and then decrease it to the out-of-the-box setting of 1. See the following examples:

To increase logging, use the following command:

```
vxlogcfg -a -p NB -o Default -s DebugLevel=6
```

To decrease logging, use the following command:

```
vxlogcfg -a -p NB -o Default -s DebugLevel=1
```

Note: In the previous examples, the `-a` option was added to both commands because we do not want to remove the default logging, but only change the debug level to the out-of-the-box default level.

Caution: Always wait at least 1 full minute after changing the log file logging levels as it may take a minute for the changes to be implemented.

Do not leave a high level of logging in place for a long period of time as it can cause the file systems to fill up with logs.

If the OIDs are set to 0 by default, they are not affected when the default logging levels are changed. These OIDs are as follows:

- 156 – NetBackup ACE/TAO; this logs to any process that needs to utilize an ACE/TAO call
- 486 – NetBackup proxy helper; this logs to the unified `nbpxyhelper` log file. See "NetBackup proxy helper logging" on page 131.

Command-line logging

The command-line logs are as follows (on any master, media, or client server):

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

Windows: *`install path`*`\netbackup\logs\nbcert`

The `nbcert` log files log any `nbcertcmd` commands that run either manually or automatically from the application, such as during the automatic certificate renewal.

When issues occur that can be reproduced using `nbcertcmd`, the `bp.conf` file or registry `VERBOSE` setting should be increased to 5 to troubleshoot the issue. To increase the logging level, use the following command:

```
echo VERBOSE = 5 | nbsetconfig
```

NetBackup cURL logging

Any process or daemon that calls cURL will log the cURL messages on any master, media, or client server. The NetBackup cURL logging should be increased when you need to see the cURL messages in the daemons and processes that utilize the cURL calls.

The cURL logging is disabled by default, but it can be enabled by using the following command:

```
echo ENABLE_NBCURL_VERBOSE=1 | nbsetconfig
```

Note: NetBackup cURL logging is either on or off and it can be enabled on all of the NetBackup clients and servers that experience issues related to secure communication.

Java logging

Java logging can occur on any master, media, or client server on which Java is executed. Many issues with `nbwmc` and secure communication are revealed when you cannot log in to the Java console. If this occurs, it is helpful to collect the log files for the appropriate location on which you are starting the console, such as a PC or directly on the master server. See “Configuring and gathering logs when troubleshooting NetBackup Administration Console issues” on page 168.

Embeddable Authentication Client (EAT) logging

The Embeddable Authentication Client (EAT) logging occurs only on the master server. Any process or daemon that makes Authentication Services (AT) calls will log these messages. In NetBackup 8.1, the authentication (`nbatd`) log content can be added to any NetBackup processes that interacts with `nbatd` when the AT logging is enabled. To enable AT logging, use the following command:

```
echo EAT_VERBOSE=5 | nbsetconfig
```

Valid log levels are 0 through 5.

To disable EAT logging, use the following command:

```
echo EAT_VERBOSE=0 | nbsetconfig
```

Authentication Services (AT) logging

The Authentication Services (AT) log files are located as follows (on the master server only):

UNIX: `/usr/openv/logs/nbatd`

Windows: *install path*`\netbackup\logs\nbatd OID 18`

To increase logging, use the following command:

```
vxlogcfg -a -p NB -o 18 -s DebugLevel=6
```

To remove logging, use the following command:

```
vxlogcfg -r -p NB -o 18 -s DebugLevel=6
```

vssat logging

The `vssat` log files are located wherever they are specified. To enable `vssat` logging on UNIX, use the following command:

```
/usr/openv/netbackup/sec/at/bin/vssat setloglevel -l 4 -f /usr/openv/logs/nbatd/vssat.log
```

To enable `vssat` logging on Windows, use the following command:

```
install_path\Veritas\NetBackup\sec\at\bin\vssat setloglevel -l 4
  -f C:\Program Files\Veritas\NetBackup\logs\nbatd\vssat.log
```

To disable `vssat` logging on UNIX, use the following command:

```
/usr/openv/netbackup/sec/at/bin/vssat setloglevel -l 0
```

To disable `vssat` logging on Windows, use the following command:

```
install_path\Veritas\NetBackup\sec\at\bin\vssat setloglevel -l 0
```

Use `-F, --enable_fips` option to run the `vssat` command in the FIPS mode. By default, the FIPS mode is disabled.

To disable `vssat` logging in FIPS mode on UNIX, use the following command:

```
/usr/openv/netbackup/sec/at/bin/vssat setloglevel -l 0 -F
```

To disable `vssat` logging in FIPS mode on Windows, use the following command:

```
install_path\Veritas\NetBackup\sec\at\bin\vssat setloglevel -l 0 -F
```

NetBackup proxy helper logging

The locations of the NetBackup proxy helper log files are as follows on any master, media or client server:

UNIX: `/usr/openv/logs/nbpxyhelper`

For UNIX startup and shutdown issues: `/usr/openv/netbackup/logs/vnetd`

Windows: *install path*`\netbackup\logs\nbpxyhelper`

For Windows startup and shutdown issues: *install path*`\netbackup\logs\vnetd`

Originator ID 486

The NetBackup proxy helper log files are useful when there are issues with communication due to SSL/TSL errors or other secure communication issues. You can start the processes by using the `vnetd -standalone` command. If there are startup and shutdown issues, examine the `vnetd` log file.

The following are examples of the expected minimum number of `vnetd` processes:

```
/usr/openv/netbackup/bin/vnetd -proxy inbound_proxy -number 0

/usr/openv/netbackup/bin/vnetd -proxy outbound_proxy -number 0

/usr/openv/netbackup/bin/vnetd -standalone
```

The inbound and outbound proxy processes send logs to the `nbpxyhelper` log files. The communication between them can be followed through the job details; it locates the `:INBOUND` or `:OUTBOUND` connection ID and searches for them in the `nbpxyhelper` log files. The `:INBOUND` and `:OUTBOUND` connections are only displayed if there is an error. See the following example:

```
Aug 5, 2018 5:13:14 PM - Info nbjm (pid=3442) starting backup job (jobid=268) for
  client nbclient1, policy ANY_nbclient1, schedule Full-EXPIRE_IMMEDIATELY
Aug 5, 2018 5:13:14 PM - Info nbjm (pid=3442) requesting STANDARD_RESOURCE resources from RB
  for backup job (jobid=268, request id:{5DD92BD0-98F4-11E8-AEE4-55B66A58DDB2})
```

```
Aug 5, 2018 5:13:14 PM - requesting resource  __ANY__
Aug 5, 2018 5:13:14 PM - requesting resource  nbmaster2.NBU_CLIENT.MAXJOBS.nbclient1
Aug 5, 2018 5:13:14 PM - requesting resource  nbmaster2.NBU_POLICY.MAXJOBS.ANY_nbclient1
Aug 5, 2018 5:13:15 PM - Error bpbrm (pid=21177) [PROXY] Connecting host: nbmaster2
Aug 5, 2018 5:13:15 PM - Error bpbrm (pid=21177) [PROXY] ConnectionId:
  {5E0FBBD2-98F4-11E8-804A-EC7198374CC6}:OUTBOUND
```

By default, OID 486 is set to `DebugLevel=0` due to the potential to create many log files. Do not leave the logging enabled for long periods of time at `DebugLevel=6`.

The logging level can be changed by using the `vxlogcfg` command. See the following examples:

To add logging, use the following command:

```
vxlogcfg -a -p NB -o 486 -s DebugLevel=6
```

To remove logging, use the following command:

```
vxlogcfg -a -p NB -o 486 -s DebugLevel=0
```

Note: In this case, the logging level is being explicitly set to 0 after the troubleshooting is finished.

NetBackup proxy tunnel logging

The NetBackup proxy tunnel logs are at the following location (on any media server):

UNIX: `/usr/openv/logs/nbpxytnl`

Windows: `install path\netbackup\logs\nbpxytnl`

Originator ID 490

In NetBackup 8.1, the media servers can be used as a proxy tunnel for clients that cannot connect directly with the master server.

If there are issues between the clients and media servers that act as a proxy, the `nbpxytnl` logging should be increased. The logging level can be changed using the `vxlogcfg` command. See the following examples:

To add logging, use the following command:

```
vxlogcfg -a -p NB -o 490 -s DebugLevel=6
```

To remove logging, use the following command:

```
vxlogcfg -r -p NB -o 490 -s DebugLevel=6
```

PBX logging

The Private Branch Exchange (PBX) logs are located at the following location on any master, media or client server:

UNIX: `/opt/VRTSpbx/log`

Windows: `C:\Program Files (x86)\VERITAS\VxPBX\log`

The PBX log files can be critical when you troubleshoot secure communication issues. In these cases, you may have to increase the size and the number of log files as the defaults are to retain 5 log files at 1MG each. To increase the size of the log files to 50000 KB and number of log files to 20, use the `vxlogcfg` command, as follows:

Windows (in `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Veritas\VxICS\logcfg\103`):

```
C:\Program Files (x86)\VERITAS\VxPBX\bin\vxlogcfg –a –p 50936 –s "MaxLogFileSizeKB=50000"
  –o 103
C:\Program Files (x86)\VERITAS\VxPBX\bin\vxlogcfg –a –p 50936 –s "NumberOfLogFiles=20"
  –o 103
```

This lets you capture 20 log files at 50 mg each.

UNIX:

1. Use the `vxlogcfg` command to increase the size of the log files to 51200 KB and the number of log files to 10 as follows:

   UNIX (in `/etc/vx/VxICS/icsul.conf`):

```
/opt/VRTSpbx/bin/vxlogcfg –a –p 50936 –s "MaxLogFileSizeKB=51200" –o 103
/opt/VRTSpbx/bin/vxlogcfg –a –p 50936 –s "NumberOfLogFiles=10" –o 103
```

2. Change directories using the `cd /opt/VRTSpbx/log` command. Verify that the log files are increasing in size to more than 1 mg (it should read the change within 1 minute).
3. To verify the PBX log settings, the configuration file can be viewed as follows:
   - Change your directory to `/etc/vx/VxICS`.
   - Use the `cat icsul.conf` command and verify that the changes were made.

   See the following example:

```
cat icsul.conf
##############################################
# Caution! Do not update/modify file by hand.
# Use vxlogcfg tool to update/modify this file
##############################################
103.DebugLevel=6
103.AppMsgLogging=ON
103.LogToOslog=false
103.LogDirectory=/var/log/VRTSpbx/
103.L10nResourceDir=/opt/VRTSpbx/resources
103.L10nLib=/optVRTSpbx/lib/libvxexticu.so.3
103.L10nResource=VxPBX
103.MaxLogFileSizeKB=51200
103.RolloverMode=FileSize
103.NumberOfLogFiles=10
103.LogRecycle=true
```

Sending secure communication logs to Veritas Technical Support

If you encounter a issue that is due to certificate deployment or secure communication, you can send a problem report and the relevant logs to Veritas Technical Support for assistance. Table 10-2 provides a list of logs and the recommended logging levels that Veritas Technical Support may need to diagnose certain secure communication issues.

Table 10-2 Logs to gather for secure communication issues

Type of issue	Logs to gather

Java console login failure	From the master server:
	■ The `bpjava-msvc` log at verbose 5
	■ The `bpjava-susvc` log at verbose 5
	■ The `nbsl` log at debug level 6
	  ■ Enable the OIDs 156 (ACE\TAO) and 137 (NB libraries) at debug level 6. They will write to the calling process.
	  ■ Enable the Embeddable Authentication Client (EAT) logging
	■ The `nbatd` log (from the AT logging section)
	■ `vssat.log` (see the `vssat` log)
	■ The PBX log files
	■ The NetBackup support utility `nbsu`
	From the entity on which the console was started:
	■ `nbj` or `jbp` log file

Starting or unexpected termination of `nbwmc` on the master server	All of the logs are located on the master: ■ The `nbwebservice` logs (see the web services logs) ■ The `nbwmc` logs, the `webserver` logs, and all of the files from the data of the issue, including any `hs_err.*` or `*.hprof` files ■ Installation log files ■ Gather an `nbsu` as it automatically collects the installation logs ■ The `nbcert` log ■ The PBX log files
Certificate deployment during installation	From the server on which the certificates cannot be deployed: ■ The `nbcert` log ■ Installation logs ■ Gather an `nbsu` as it automatically collects the installation logs ■ The `nbpxyhelper` log files at debug level 4 or greater From the master server: ■ The `nbcert` log ■ The `nbpxyhelper` log files at debug level 4 or greater

Table 10-2 Logs to gather for secure communication issues *(continued)*

Type of issue	Logs to gather
Backup failures due to failure to establish a secure connection	From the problem client or media server: ■ The `nbcert` log ■ The `nbpxyhelper` log files at debug level 4 or greater From the master or media server rejecting the connection: ■ The `nbpxyhelper` log files at debug level 4 or greater ■ Any process logs that are involved in the communication, such as `bprd`, `bptm`, and `bpbrm`. From the master: ■ The job details **Note:** Ensure that cURL logging is enabled.
Disaster recovery (DR) package creation during catalog backup	From the master server: ■ The `bpdbm` log at verbose 5
Web server tunnel (media) or web server router (client)	From the media server acting as the tunnel: ■ The `nbpxytnl` log files at debug level 6 ■ The `nbpxyhelper` log files at debug level 4 or greater From the client/router: ■ The `nbcert` log file ■ The `bpcd` log file ■ The `nbproxyhelper` log files at debug level 4 or greater From the master server: ■ The `nbproxyhelper` log files at debug level 4 or greater. **Note:** Ensure that cURL logging is enabled.

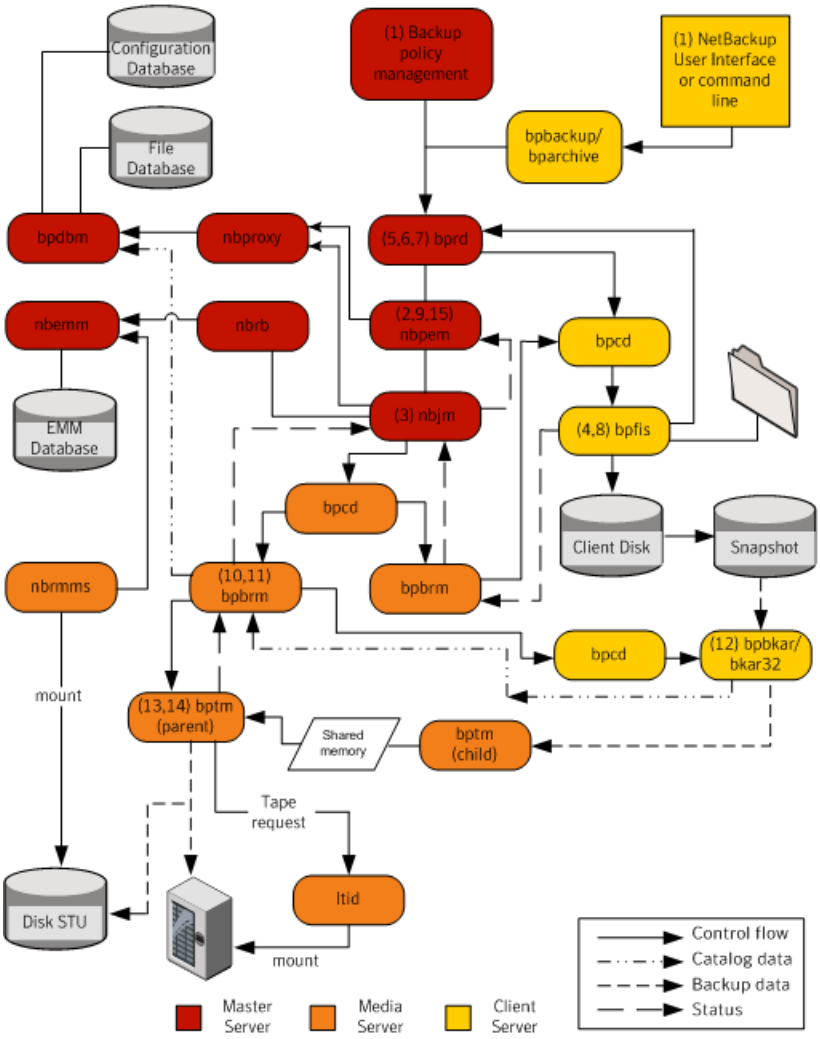
Chapter 11

Snapshot technologies

This chapter includes the following topics:

Snapshot Client backup

The following shows a typical snapshot backup process. In this scenario, the snapshot is created on the client and is then backed up to a storage unit (disk or tape) from that client. With the exception of Windows open file backups that do not use multiple data streams, all snapshots are created by a separate parent job, followed by a child job that backs up the snapshot. For non-multistreamed Windows Open File Backups, `bpbrm` using `bpcd` invokes `bpfis` to take a snapshot of individual drives. If you use System State or Shadow Copy Component backups, `bpbkar32` creates the snapshot using Volume Shadow Copy Service (VSS). Windows Open File Backups do not require a Snapshot Client license, although they do use Snapshot Client components, such as `bpfis`.



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

Snapshot Client backup procedure

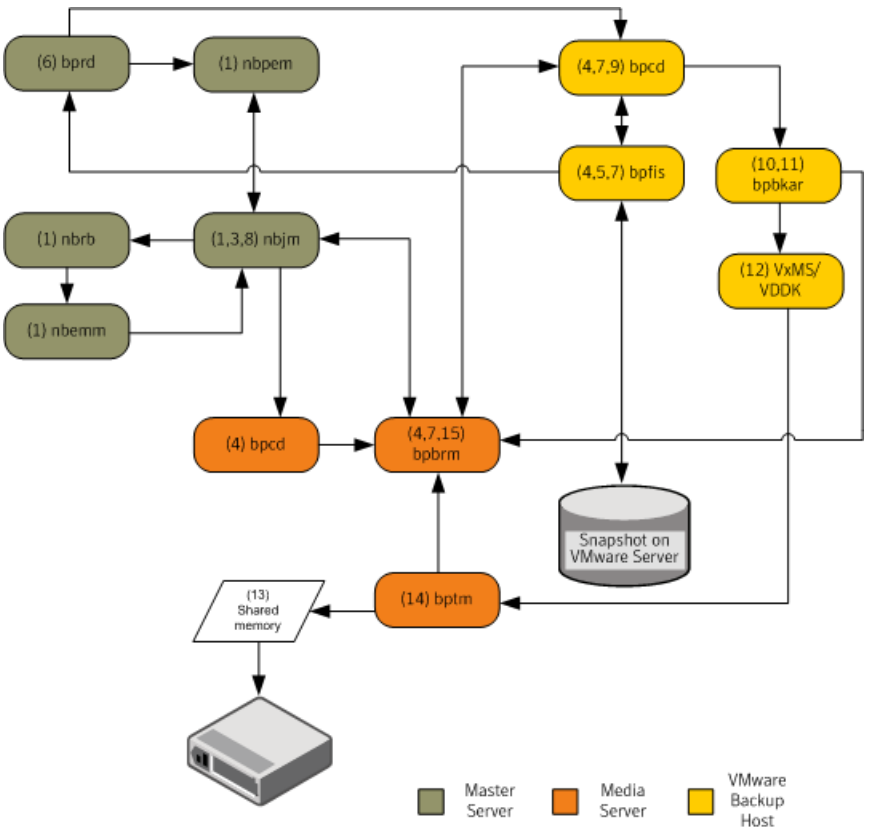
1. The NetBackup master server or primary client initiates the backup, which causes the NetBackup Request Daemon (bprd) to submit a backup request to the NetBackup Policy Execution Manager (nbpem). nbpem processes the policy configurations.
2. nbpem uses nbjm to start a parent job to create the snapshot. 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 backs up itself (bptm and bpbkar are on the same host) or the media server backs up a client that resides on a different host.
    - If the media server backs up itself, bpbkar stores the snapshot-based image block-by-block in shared memory on the media server.
    - If the media server backs up a client that resides on a different host, the bptm process on the server creates a child process of itself. The child receives the snapshot-based image from the client by means of socket communications and then stores the image block-by-block in shared memory.
13. The original bptm process takes the backup image from shared memory and sends it to the storage device (disk or tape).

14 bptm sends the backup completion status to bpbrm, which passes it to nbjm.

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

VMware backup

The following shows a VMware backup process.



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

VMware backup procedure

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

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

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

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

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

- Veritas V-Ray within VxMS generates the catalog of all the files inside the VMDK from both Windows and Linux VMs. The operation occurs while backup data is being streamed. 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 off-host backup with no burden on the production vSphere. Even the source side deduplication occurs in an off-host system.

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

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

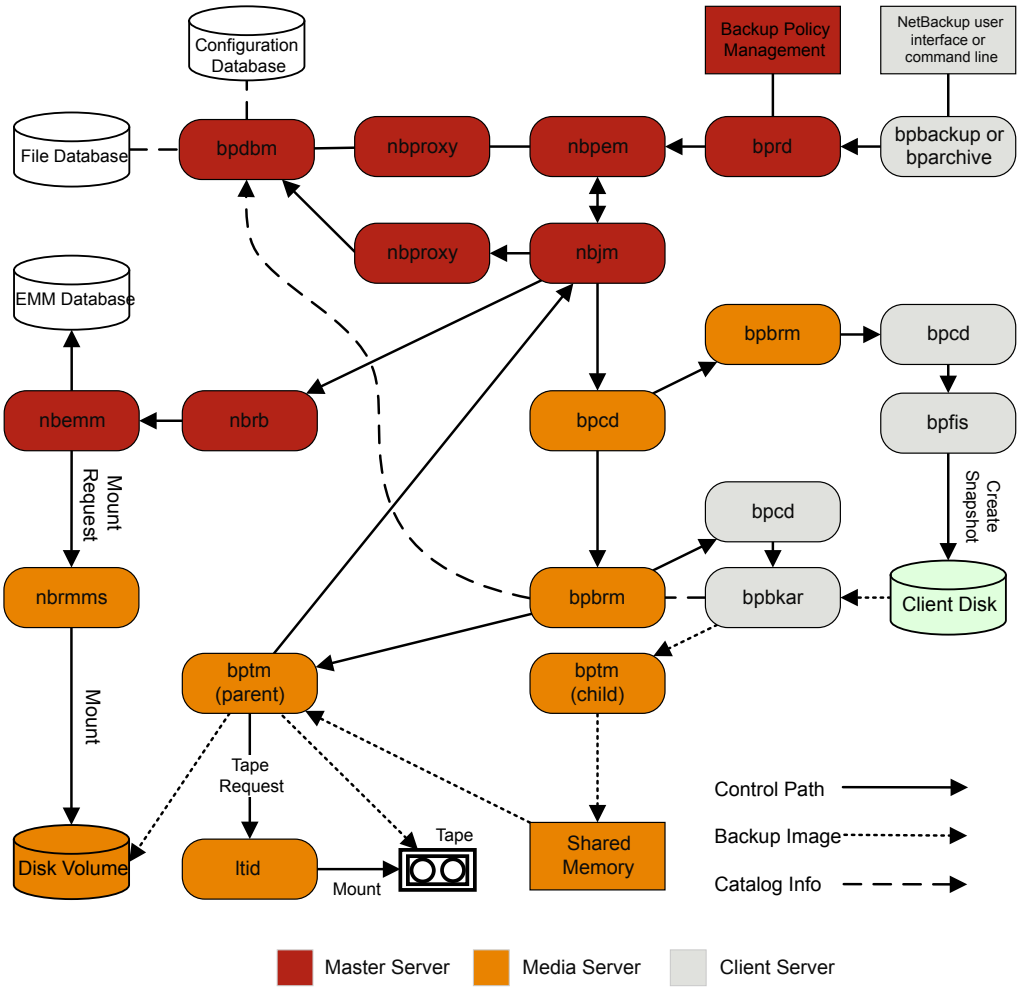
14 `bptm` sends backup completion status to `bpbrm`, which passes it to `nbjm` and `nbpem`.

15 `nbpem` tells `nbjm` to delete the snapshot. `nbjm` starts a new instance of `bpbrm` on the media server, and `bpbrm` starts a new instance of `bpfis` on the VMware Backup Host. `bpfis` deletes the snapshot on the vSphere environment. `bpfis` and `bpbrm` report their status and exit.

Snapshot backup and Windows open file backups

Figure 11-1 shows the overall snapshot backup process. PBX (not shown in the diagram) must be running for NetBackup to operate.

Figure 11-1 Snapshot backup and Windows open file backup using multiple data streams



Notes:
* For details on these components, see the Media and Device Management Functional Description later in this chapter.
** If the media server is backing up itself (server and client on same host), there is no bptm child: bpbkar sends the data directly to shared memory.

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 NetBackup Policy Execution Manager `nbpem`. `nbpem` processes the policy configurations.
- `nbpem` (through `nbjm`) starts a parent job to create the snapshot. This job is separate from the job that backs up the snapshot.
- `nbjm` starts an instance of `bpbrm` through `bpcd` on the media server, and `bpbrm` starts `bpfis` through `bpcd` on the client.
- `bpfis` creates a snapshot of the client's data by means of a snapshot method.
- When `bpfis` is finished, it sends snapshot information and completion status to `bpbrm` and exits. `bpbrm`, in turn, reports the snapshot information and status to `nbjm` and exits. `nbjm` relays the information and status to `nbpem`.
- `nbpem` submits a child job for the backup to `nbjm`, with a file list derived from the snapshot information. `nbjm` starts `bpbrm` to back up the snapshot.
- `bpbrm` starts `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.
- `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 `bpbkar` on the same host), or the media server backs up a client on a different host. If the media server backs up itself, `bpbkar` 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.

Chapter 12

Locating logs

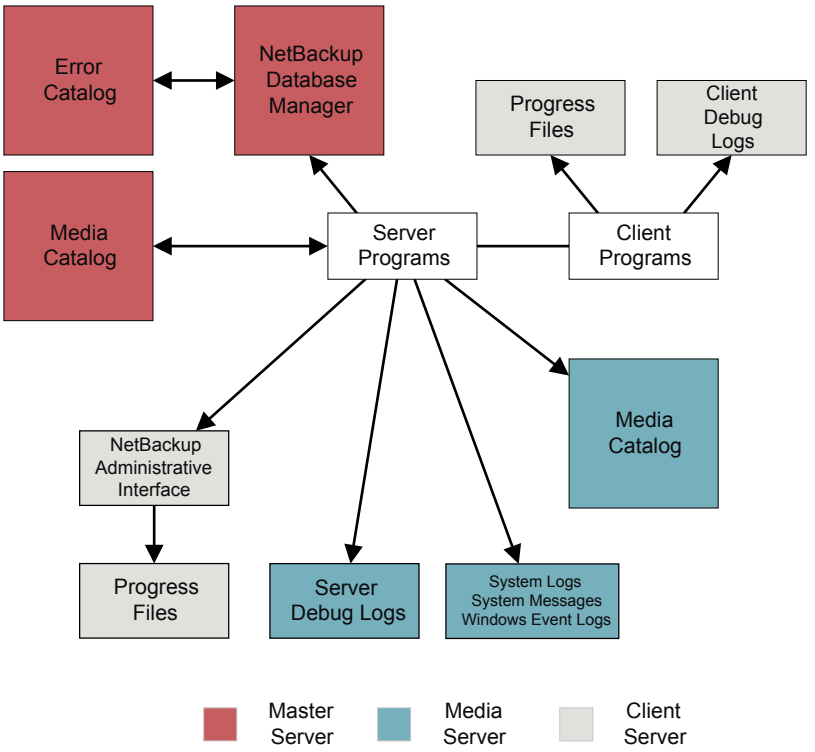
This chapter includes the following topics:

- nbproxy logging
- nbrb logging
- NetBackup Vault logging
- NetBackup web services logging
- NetBackup web server certificate logging
- PBX logging
- reqlib logging
- Robots logging
- tar logging
- txxd and txxcd logging
- vnetd logging

Overview of NetBackup log locations and processes

Figure 12-1 shows the location of the log and report information on the client and the server and the processes that make the information available.

Figure 12-1 Logs in the NetBackup enterprise system



For information on reports, see the following topic.

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.

acsssi logging

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

Log location	`/usr/openv/volmgr/debug/acsssi`
Server where it resides	media
Logging method	Legacy

bpbackup logging

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

Log location	*`install_path`*`\NetBackup\logs\bpbackup` `/usr/openv/netbackup/logs/bpbackup`
Server where it resides	client
Logging method	Legacy

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	*`install_path`*`\NetBackup\logs\bpbkar` `/usr/openv/netbackup/logs/bpbkar`
Server where it resides	client
Logging method	Legacy

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	*`install_path`*`\NetBackup\logs\bpbrm` `/usr/openv/netbackup/logs/bpbrm`
Server where it resides	media
Logging method	Legacy

bpcd logging

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

Log location	*install_path*\NetBackup\logs\bpcd /usr/openv/netbackup/logs/bpcd
Server where it resides	media and client
Logging method	Legacy

bpcompatd logging

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

Log location	*install_path*\NetBackup\logs\bpcompatd /usr/openv/netbackup/logs/bpcompatd
Server where it resides	master
Logging method	Legacy

bpdbm logging

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

Log location	*install_path*\NetBackup\logs\bpdbm /usr/openv/netbackup/logs/bpdbm
Server where it resides	master
Logging method	Legacy

bpjobd logging

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

Log location	*install_path*\NetBackup\logs\bpjobd /usr/openv/netbackup/logs/bpjobd
Server where it resides	master
Logging method	Legacy

bprd logging

The NetBackup Request Daemon (bprd) responds to client and administrative requests for backups, restores, and archives.

Log location	*install_path*\NetBackup\logs\bprd /usr/openv/netbackup/logs/bprd
Server where it resides	master
Logging method	Legacy

bprestore logging

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

Log location	*install_path*\NetBackup\logs\bprestore /usr/openv/netbackup/logs/bprestore
Server where it resides	client
Logging method	Legacy

bptestnetconn logging

The bptestnetconn command performs several tasks that help you analyze DNS and connectivity problems with any specified list of hosts, including the server list in the NetBackup configuration.

To help troubleshoot connectivity problems between the services that use CORBA communications, bptestnetconn can perform and report on CORBA connections to named services. The command can also perform and report the responsiveness of the NetBackup Web Service. The command shows the connection direction,

whether the communication was encrypted by a connection to the secure proxy process or not.

Log location	*install_path*\Veritas\NetBackup\logs\nbutils /usr/openv/logs/nbutils
Server where it resides	master, client, and media
Logging method	Unified

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	*install_path*\NetBackup\logs\bptm /usr/openv/netbackup/logs/bptm
Server where it resides	media
Logging method	Legacy

daemon logging

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

Log location	*install_path*\volmgr\debug\daemon /usr/openv/volmgr/debug/daemon
Server where it resides	master and media
Logging method	Legacy

ltid logging

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

Log location	*install_path*\volmgr\debug\ltid /usr/openv/volmgr/debug/ltid
Server where it resides	media
Logging method	Legacy

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.

Create the following directory before you start nbemm:

Windows: install_path\Volmgr\debug\vmscd\

UNIX: /usr/openv/volmgr/debug/vmscd

Log location	*install_path*\NetBackup\logs\nbemm /usr/openv/logs/nbemm
Server where it resides	master
Logging method	Unified

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	*install_path*\NetBackup\logs\nbjm /usr/openv/logs/nbjm
Server where it resides	master
Logging method	Unified

nbpem logging

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

Log location	*install_path*\NetBackup\logs\nbpem /usr/openv/logs/nbpem
Server where it resides	master
Logging method	Unified

nbproxy logging

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

Log location	*install_path*\NetBackup\logs\nbproxy /usr/openv/netbackup/logs/nbproxy
Server where it resides	master
Logging method	Legacy

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	*install_path*\NetBackup\logs\nbrb /usr/openv/logs/nbrb
Server where it resides	master
Logging method	Unified

NetBackup Vault logging

Vault Session directories are found in the following location:

install_path\NetBackup\vault\sessions\vaultname*session_x*

Where *session_x* is the session number. This directory contains vault log files, temporary working files, and report files.

See the NetBackup Administrator's Guide, Volume II for instructions about how to use this entry.

NetBackup web services logging

This topic describes the logs for the NetBackup web services.

Log location	*Web server logs* *install_path*\NetBackup\wmc\webserver\logs /usr/openv/wmc/webserver/logs *Web applications logs* *install_path*\NetBackup\logs\nbwebservice /usr/openv/logs/nbwebservice
Server where it resides	master
Logging method	Unified The NetBackup web server framework does not use the standard VxUL format. For more information on the format of these logs and how they are created, see the documentation for Apache Tomcat at http://tomcat.apache.org.

See the NetBackup Troubleshooting Guide for more information on how to access the web services logs.

NetBackup web server certificate logging

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

Log location	*install_path*\NetBackup\logs\nbatd *install_path*\NetBackup\logs\nbcert C:\ProgramData\Veritas\NetBackup\InstallLogs\ WMC_configureCerts_*yyyymmdd_timestamp*.txt /usr/openv/logs/nbatd /usr/openv/netbackup/logs/nbcert /usr/openv/wmc/webserver/logs/configureCerts.log
Server where it resides	master
Logging method	The nbatd log uses unified logging. The configureCerts.log uses a simple logging style and not VxUL. The nbcert log uses the legacy logging method.

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

Log location	*install_path*\NetBackup\logs\nbatd *install path*\NetBackup\logs\nbwebservice C:\ProgramData\Veritas\NetBackup\InstallLogs\ WMC_configureCerts_*yyyymmdd_timestamp*.txt /usr/openv/logs/nbatd /usr/openv/logs/nbwebservice /usr/openv/wmc/webserver/logs/configureCerts.log
Server where it resides	master
How to access	The nbwebservice (OID 466 and 484) and nbatd (OID 18) logs use unified logging. The configureCerts.log uses a simple logging style and not VxUL.

See the NetBackup Troubleshooting Guide for more information on how to access the web services logs.

PBX logging

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

Log location	*install_path*\VxPBX\log /opt/VRTSpbx/log
Server where it resides	master, media, and client
Logging method	Unified To view logs for PBX, you must use the PBX product ID, which is 50936. You also must have root or administrator privileges.

See the NetBackup Troubleshooting Guide for more information on how to access PBX logs.

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	*install_path*\volmgr\debug\reqlib /usr/openv/volmgr/debug/reqlib
Server where it resides	master and media
Logging method	Legacy

Robots logging

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

Log location	*install_path*\volmgr\debug\robots /usr/openv/volmgr/debug/robots
Server where it resides	media
Logging method	Legacy

See "txxd and txxcd logging" on page 159.

tar logging

The Tape Archive program (`tar`) writes restore data to the client disk. On Windows clients, the binary name is `tar32.exe` and on UNIX clients the binary name is `nbtar`.

Log location	*install_path*\NetBackup\logs\tar /usr/openv/netbackup/logs/tar
Server where it resides	client
Logging method	Legacy

See "About restore logging" on page 79.

txxd and txxcd logging

The robotic daemon (`txxd`, where *xx* varies based on the type of robot being used) provides the interface between `ltid` and the tape library. The robotic control daemon (`txxcd`) provides the robotic control for the robot and communicates mount and unmount requests.

Log location	The `txxd` 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 "UNIX logging with syslogd" on page 47. See "Logging options with the Windows Event Viewer" on page 47. See "Robots logging" on page 158.
Logging method	The 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 45. On UNIX, debug information is also included by starting the daemon with the `-v` option (either by itself or through `ltid`).

vnetd logging

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

Log location	*install_path*\NetBackup\logs\vnetd /usr/openv/logs/vnetd or /usr/openv/netbackup/logs/vnetd if the vnetd directory exists there. If the vnetd directory exists in both locations, logging occurs only in /usr/openv/netbackup/logs/vnetd.
Server where it resides	master, media, and client
Logging method	Legacy

Chapter 13

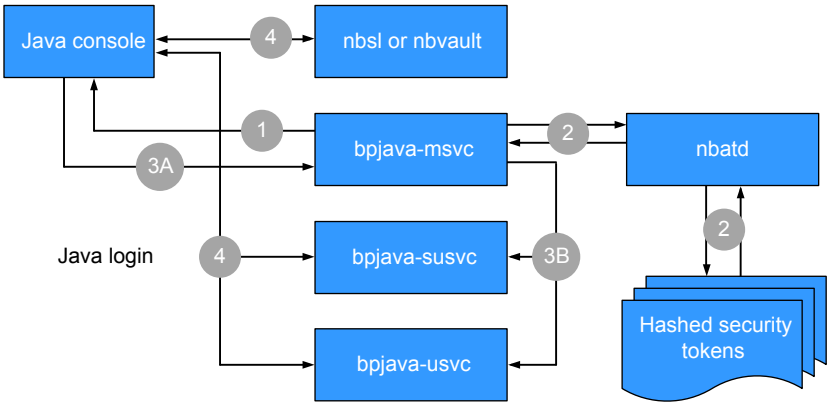
NetBackup Administration Console logging

This chapter includes the following topics:

NetBackup Administration Console logging process flow

The console can run directly on a supported Java-capable UNIX computer or on a Windows computer where the **NetBackup Administration Console** is installed.

The NetBackup Administration Console logging process flow is as follows:



The following steps describe the NetBackup Administration Console login process:

1. The user initiates a login request to the NetBackup Administration Console. The credentials are sent to `bpjava-msvc` over the Secure Sockets Layer (SSL) using the Server Security Certificate.
2. The `bpjava-msvc` process authenticates the token through `nbatd`, which reads the hashed security tokens on the server.
3. The following steps describe the process with the session certificate:
   - The `bpjava-msvc` process sends a response to the console login with a session token and a fingerprint of the session certificate.
   - The `bpjava-msvc` process initiates the appropriate `bpjava-*usvc` process and the session certificate and token are passed to one of the following processes:
     - `bpjava-susvc` for the **NetBackup Administration Console**
     - `bjava-usvc` for the Backup, Archive, and Restore (BAR) interface
4. Various calls are made between the NetBackup Administration Console and `nbsl`, `bpjava-*usvc`, and `nbvault` (if configured) to populate the interface with the appropriate contents.

Enabling detailed debug logging for the NetBackup Administration Console

The **NetBackup Administration Console** is a distributed application that allows the administration of remote NetBackup servers. All administration is accomplished through the application server of the console, which has an authentication service

and a user service. The logon request is sent to the authentication service. If the user name and password are valid, 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.

Table 13-1 describes how to create detailed debug logging for the **NetBackup Administration Console**.

Table 13-1 Enabling detailed debug logging

Step	Description
Step 1	On the NetBackup client or server, create the following directories: ■ `bpjava-msvc` (authentication service) ■ `bpjava-susvc` (the user service on the server) ■ `bpjava-usvc` (the user service on the client) Create the directories in the following locations: ■ *`install_path`*`\NetBackup\logs` (Windows) ■ `/usr/openv/netbackup/logs` (UNIX)

Step 2	Add the following line to the `Debug.properties` file:
	`debugMask=0x00040000`
	On UNIX, change the file on the UNIX machine where you run the `jnbSA` or `jbpSA` commands.
	If you use the **NetBackup Remote Administration Console**, change the file in the following locations:
	`/usr/openv/java`
	`install_path``\VERITAS\java`

Step 3	If you use the **Remote Administration Console**, edit the `nbjava.bat` file to redirect output to a file: *`install_path`*`\VERITAS\java\nbjava.bat`

Setting up a secure channel between the NetBackup Administration Console and bpjava-*

The following steps describe the process flow to set up a secure channel between the NetBackup Administration Console and `bpjava-*`:

Note: The following processes are used: `bpjava-msvc`, which controls the login and authentication; `bpjava-susvc`, which is the administration console process; and `bpjava-usvc`, which is the client Backup, Archive, and Restore (BAR) interface.

1. The user initiates a login to the console. The credentials are sent to `bpjava-msvc` over the SSL (using the Server Security Certificate).
2. The `bpjava-msvc` process authenticates the user who uses the user credentials that were received in step 1.
3. After the user is authenticated, the `bpjava-msvc` process performs the following:
   - Generates the entities that are called the self-signed session certificate, the key, and the session token.
   - Launches the daemon `bpjava-*usvc` to gather more requests from the NetBackup Administration Console.
   - Passes the self-signed session certificate and the session token to `bpjava-*usvc`.

     Note: The `bpjava-*usvc` process uses a session certificate as a Server Security Certificate for the SSL channel. It uses the session token to authenticate the NetBackup Administration Console. The console does not use credentials while it connects to the `bpjava-*usvc` process. The NetBackup Administration Console uses the session token for authentication.

   - Sends the session token and the fingerprint of the session certificate to the NetBackup Administration Console.
   - Persists session token and user information to a secure directory (*`install_path`*`/var`; for example, `/usr/openv/var`) in a file on the NetBackup host. This directory is accessible only to the root/administrator. The file name format is as follows:

     `hash(session token)_bpjava-*usvc_pid`

     Note: `msvc` saves this information so it can be used by `nbsl` or `nbvault` to authenticate the NetBackup Administration Console.

   - The `msvc` process stops the execution and exits.
4. `bpjava-*usvc` uses the session certificate to start the secure channel with the NetBackup Administration Console. This secure channel is a one-way

authenticated SSL channel. (Only the server certificate is present and there is no peer certificate. There is no certificate from the NetBackup Administration Console side.)

5. The NetBackup Administration Console receives the session certificate as a part of the initial SSL handshake. It verifies the authenticity of the session certificate by using the pre-existing fingerprint of the session certificate (see step 3). The NetBackup Administration Console calculates the fingerprint of the session certificate that was received from `bpjava-*usvc` due to the SSL handshake. It compares the new fingerprint with the fingerprint sent by `msvc`.
6. Once the authenticity of the certificate is verified, the NetBackup Administration Console sends the session token (received in step 3) to `bpjava-*usvc`.
7. `bpjava-*usvc` verifies the received session token with the pre-existing one (see step 3).
8. The success of the session token validation creates trust between `bpjava-*usvc` and the NetBackup Administration Console.
9. All further communication occurs between `bpjava-*usvc` and the NetBackup Administration Console on this trusted secure channel.

Setting up a secure channel between the NetBackup Administration Console and either nbsl or nbvault

The following steps describe the process flow to set up a secure channel between the NetBackup Administration Console and either `nbsl` or `nbvault`:

1. Trust is already set up between the NetBackup Administration Console and `bpjava-*`. The user information and session token already exist in a designated location with a name similar to the following:

   `hash(session token)_susvc_pid`

   See "Setting up a secure channel between the NetBackup Administration Console and bpjava-*" on page 163.
2. The NetBackup Administration Console sends a request to `nbsl/nbvault` for a secure connection.
3. `nbsl/nbvault` accepts the request and initiates a secure channel using the security certificate on the host. These daemons run with root/administrator privileges and can access the security certificate.

4. This secure channel is a one-way authenticated SSL channel where only the server certificate is present and there is no peer certificate. There is no certificate from the NetBackup Administration Console side.
5. The trust options for the security certificate are as follows:
   - The NetBackup Administration Console accepts the security certificate (or gives approval for this secure channel) if it trusts the NetBackup Certificate Authority (CA) who signed the security certificate.
   - If the NetBackup Administration Console does not trust the CA who signed the security certificate, it displays a pop-up dialog box. This dialog box asks if the user trusts the CA who has signed the certificate (This is a one-time activity. After the user gives consent to trust the CA, the dialog box does not display again.).
6. The NetBackup Administration Console sends a session token to `nbsl/nbvault`. See "Setting up a secure channel between the NetBackup Administration Console and bpjava-*" on page 163.
7. `nbsl/nbvault` verifies this session token by performing the following procedure:
   - Generates a hash of the session token that was received
   - Searches for the file with the name that starts with this hash at the designated location
   - If the file is found, it extracts the PID from it (see step 1)
   - Checks to see if the PID is valid
8. The success of the verification creates a trust between `nbsl/nbvault` and the NetBackup Administration Console.
9. All further communication occurs between `nbsl/nbvault` and the NetBackup Administration Console on this trusted secure channel.

NetBackup Administration Console logging configuration on NetBackup servers and clients

Java console logging is automatically set up on systems on which the NetBackup client or server software is installed along with the Java GUI option. The Java logs are located in the following pre-existing log directories:

The Java GUI logs are located in the following log directories for root and administrator users:

- UNIX: `/usr/openv/netbackup/logs/user_ops/nbjlogs/`

- Windows: *install_directory*\netbackup\logs\user_ops\nbjlogs\

The Java GUI logs are located in the following log directories for non-root and non-administrator users:

- UNIX: /usr/openv/netbackup/logs/user_ops/nbjlogs/<*non-root-username*>
- Windows: *install_directory*\netbackup\logs\user_ops\nbjlogs\<*non-admin-username*>

The administrator needs to create the non-root username directories under the nbjlogs directory using mklogdir -user *username* -group *groupname* command which is present in the NetBackup legacy log folder. If these username directories are not created with appropriate write permission for that user, then the users home directory is used for logging. The nbjlogs folder is created in the user's home directory first and all logs appear in this folder. If the home directory is not accessible, the logs are redirected to console. The administrator can also use the mklogdir command to create a specific log directory for a specific user. For example, use the mklogdir -create user_ops/nbjlogs -user *username* -group *groupname* command to create this directory.

Logging Java operations for the NetBackup Remote Administration Console

To log Java operations for a host that uses the NetBackup Remote Administration Console, you must update the setconf.bat file.

1. Create the following directory:

   C:\Program Files\Veritas\NetBackup\logs\user_ops\nbjlogs

2. Edit the following file:

   install_path\Veritas\Java\setconf.bat

3. Locate the following line and remove the remark:

   REM SET NB_INSTALL_PATH=C:\\Program Files\\Veritas\NetBackup

4. Save the file.
5. The next time that you open the Console, the following log is created:

   C:\Program Files\Veritas\NetBackup\logs\user_ops\nbjlogs

Configuring and gathering logs when troubleshooting NetBackup Administration Console issues

After the NetBackup Administration Console is installed, the log levels are configured to gather a detailed set of logs.

The NetBackup Administration Console uses the `Debug.properties` file to determine which logging level to use:

`/usr/openv/java/Debug.properties`

`install_dir``\VERITAS\Java\Debug.properties`

The following settings are tuned to enable additional logging:

```
printcmds=true
debugMask=0x00040000
```

To increase the verbosity to max (which is recommended for troubleshooting), set `debugMask` to `debugMask=0x00160000`.

1. Gather the following NetBackup Administration Console logs from the following pre-existing log directories on the system from which the console was started:

   The Java GUI logs are located in the following log directories for root and administrator users:

   - UNIX: `/usr/openv/netbackup/logs/user_ops/nbjlogs/`
   - Windows: *`install_directory`*`\netbackup\logs\user_ops\nbjlogs\`

   The Java GUI logs are located in the following log directories for non-root and non-administrator users, the Java GUI logs are located in the following log directories:

   - UNIX: `/usr/openv/netbackup/logs/user_ops/nbjlogs/<`*`non-root-username`*`>`
   - Windows: *`install_directory`*`\netbackup\logs\user_ops\nbjlogs\<`*`non-admin-username`*`>`

   The administrator needs to create the non-root username directories under the `nbjlogs` directory using `mklogdir -user` *`username`* `-group` *`groupname`* command which is present in the NetBackup legacy log folder. If these username directories are not created with appropriate write permission for that user, then the users home directory is used for logging. The `nbjlogs` folder is created in the user's home directory first and all logs appear in this folder. If the home directory is not accessible, the logs are redirected to console.

2. On the master server, log on to the NetBackup Administration Console to create the `admin`, `bpjava-msvc`, `bpjava-susvc`, and `bpjava-usvc` log directories and enable VERBOSE 5 logging. You do not have to restart the NetBackup daemons for the logging level changes to take effect.

   For UNIX systems, create the following directories:

   - `/usr/openv/netbackup/logs/admin`
   - `/usr/openv/netbackup/logs/bpjava-msvc`
   - `/usr/openv/netbackup/logs/bpjava-susvc`
   - `/usr/openv/netbackup/logs/bpjava-usvc`

3. In the `/usr/openv/netbackup/bp.conf` file add the following lines:

   ```
   ADMIN_VERBOSE = 5
   BPJAVA-MSVC_VERBOSE = 5
   BPJAVA-SUSVC_VERBOSE = 5
   BPJAVA-USVC_VERBOSE = 5
   ```

4. For Windows systems, create the following directories:

   - *install_dir*`\VERITAS\NetBackup\logs\admin`
   - *install_dir*`\VERITAS\NetBackup\logs\bpjava-msvc`
   - *install_dir*`\VERITAS\NetBackup\logs\bpjava-susvc`
   - *install_dir*`\VERITAS\NetBackup\logs\bpjava-usvc`

5. Update the Windows registry at **HKEY_LOCAL_MACHINE > SOFTWARE > Veritas > NetBackup > CurrentVersion > Config** and add the following entries of type `DWORD`:

   ```
   ADMIN_VERBOSE = 5
   BPJAVA-MSVC_VERBOSE = 5
   BPJAVA-SUSVC_VERBOSE = 5
   BPJAVA-USVC_VERBOSE = 5
   ```

6. Run the following commands to set up detailed logging for `nbatd` (OID 18) and `nbsl` (OID 132). OID 137 (NetBackup libraries) and OID 156 (CORBA/ACE) write to the caller that requires access to either the libraries or CORBA/ACE, as follows:

   ```
   vxlogcfg -a -p NB -o 18 -s DebugLevel=6
   vxlogcfg -a -p NB -o 132 -s DebugLevel=6
   ```

```
vxlogcfg -a -p NB -o 137 -s DebugLevel=6
vxlogcfg -a -p NB -o 156 -s DebugLevel=6
```

7. Gather the `nbatd` and `nbsl` logs that are located in the following directory paths:

   For UNIX:

   - `/usr/openv/logs/nbsl`
   - `/usr/openv/logs/nbatd`

   For Windows:

   - *`install_dir`*`\VERITAS\NetBackup\logs\nbsl`
   - *`install_dir`*`\VERITAS\NetBackup\logs\nbatd`

8. Finally, gather the PBX logs, as follows:

   - For UNIX: `/opt/VRTSpbx/log` (gather any logs that cover the current date and time)
   - For Windows: *`install_dir`*`\VERITAS\pbx\log`

Undo logging

Ensure that you undo the logging after you have gathered the logs that relate to your troubleshooting issue.

To remove the log configuration settings, use the following commands:

```
vxlogcfg -r -p NB -o 18 -s DebugLevel=6
    vxlogcfg -r -p NB -o 132 -s DebugLevel=6
    vxlogcfg -r -p NB -o 137 -s DebugLevel=6
    vxlogcfg -r -p NB -o 156 -s DebugLevel=6
```

On the master server, comment out the following Java `VERBOSE` entries in the `bp.conf` file (UNIX) or in the registry (Windows):

- `ADMIN_VERBOSE`
- `BPJAVA-MSVC_VERBOSE`
- `BPJAVA-SUSVC_VERBOSE`
- `BPJAVA-USVC_VERBOSE`

Chapter 14

Using the Logging Assistant

This chapter includes the following topics:

About the Logging Assistant

The Logging Assistant is a helpful tool that can shorten the time that is required to set up and collect debug logs and other information. Because Logging Assistant automatically performs a number of functions, you can avoid the problems that are associated with manually logging into NetBackup hosts, creating log directories, and changing logging levels.

Note: Use the Logging Assistant under the guidance of Veritas Support.

The Logging Assistant uses a series of wizards to help quickly troubleshoot a problem. Depending on the category of NetBackup problem, the tool suggests the

possible hosts that may be involved in the problem and the logs that should be enabled on those hosts.

Note: While collecting NetBackup logs using Logging Assistant, the total NetBackup log size may increase. If you have enabled the **Keep logs up to GB** property on the **Host Properties > Logging** dialog box and the total NetBackup log size reaches its high water mark, logs are deleted. Logs that you want to retain may also be deleted. To avoid the deletion of logs that you want to retain, you need to disable the **Keep logs up to GB** property while you collect logs using the Logging Assistant. Alternatively, you can set the **Keep logs up to GB** property to a value higher than the current value, so that the important logs are not deleted before the log collection is complete.

No special licensing is required. However, you must have root permissions for UNIX and administrator privileges for Windows to use the Logging Assistant.

Table 14-1 shows a summary of the main Logging Assistant operations.

Table 14-1 Logging Assistant operations

Operation	Description
Add a new Logging Assistant record.	Add a Logging Assistant record that you use throughout the process of troubleshooting a NetBackup problem. Typically, you associate the record that you create with a failed job that appears in the Activity Monitor. A list of records appears when you select the Logging Assistant node in the right pane of the **NetBackup Administration Console**. See "Adding or deleting a Logging Assistant record" on page 175.
View details.	View the details for the selected record.
Delete a Logging Assistant record.	After the failed job runs successfully, you can delete the Logging Assistant record. See "Adding or deleting a Logging Assistant record" on page 175.
Enable debug logging.	Use the Setup Debug Logging Wizard to enable selected NetBackup debug logs and other processes that Technical Support uses to troubleshoot NetBackup problems. Logging Assistant automatically creates the necessary log folders and sets the log levels of the debug logs to the highest verbosity. (The highest log level of many of the debug logs is 5). See "Setting up debug logging" on page 177.

Table 14-1 Logging Assistant operations *(continued)*

Operation	Description
Set minimum debug logging.	Use the Set Minimum Debug Logging Wizard to set the specified process log levels (verbosity) to a minimum value. This setting still allows info, warning, error, and critical messages (legacy logging), and application logs, diagnostic logs, and some debug logs (unified logging). You may want to set the debug logging to minimum values after you resolve the job failure problem for the selected Logging Assistant record. See "Set minimum debug logging" on page 178.
Disable debug logging.	Use the Disable Debug Logging Wizard to disable the debug logging after you resolve the job failure problem for the selected Logging Assistant record. The only logs that NetBackup continues to generate are application logs and diagnostic logs. See "Disabling debug logging" on page 179.
Collect nbsu output.	Use the Collect nbsu Output Wizard to gather the `nbsu` diagnostic information onto the master server.
Cancel operation.	After you complete collection, you can cancel the operation if it is currently in progress (for example, if the data is too large and the operation is time-consuming). Check that the progress field of this record displays **In progress**, then right-click the Logging Assistant record and select **Cancel Operation**.

Logging Assistant sequence of operation

Table 14-2 shows a typical procedure that uses the Logging Assistant to help troubleshoot and solve a NetBackup problem.

Table 14-2 Steps for troubleshooting NetBackup problems

Step	Action	Description
Step 1	Create a Logging Assistant record.	Create a Logging Assistant record to troubleshoot a NetBackup problem. You normally associate the record with a failed job that appears in the Activity Monitor. See "Adding or deleting a Logging Assistant record" on page 175.
Step 2	Enable debug logging.	Use the Setup Debug Logging Wizard to enable selected NetBackup debug logging that Technical Support uses to troubleshoot NetBackup problems. See "Setting up debug logging" on page 177.
Step 3	Rerun the failed job.	Go to the Activity Monitor and rerun the failed job. NetBackup generates the debug logs that you have enabled.

Table 14-2 Steps for troubleshooting NetBackup problems *(continued)*

Step	Action	Description
Step 4	Collect data.	You can collect debug logs, `nbsu` diagnostics, and additional evidence. Technical Support may want you to run the NetBackup Support Utility (`nbsu`) first to get diagnostic information. You can also collect the evidence that supports the debug logs and `nbsu` diagnostic information.
Step 5	Troubleshoot the problem.	Work with Technical Support to read and analyze the debug logs, `nbsu` diagnostics, and evidence, and correct any problems you detect.
Step 6	Rerun the failed job.	After you take corrective actions, retry the operation. If the job still fails, work with Technical Support to identify and enable additional debug logs using the same Logging Assistant record. Repeat steps 2-6 until the job runs successfully.
Step 7	Disable or set minimum debug logging.	Disabling debug logging automatically removes the log directories and disables all debug logging messages. Minimum debug logging disables all messages except application logs and diagnostic logs. See "Disabling debug logging" on page 179. See "Set minimum debug logging" on page 178.
Step 8	Delete the record.	Remove the Logging Assistant record from the list of records. See "Adding or deleting a Logging Assistant record" on page 175.

Viewing the Logging Assistant records

The **Logging Assistant** node contains all the Logging Assistant records that you create. To view the details of a record, select the record and then **Actions > View Details**. Click **Associated Host and Logs** to display the hosts for the current record. These hosts include the master server, any media servers, any associated clients, and a list of logs that are enabled on each host.

The record details in the logging assistant include the following information:

Logging Assistant Record ID	The record name that you entered.
Description	The record description that you entered.
Debug logs currently enabled	**Yes**: Debug logs are set up or enabled. **No**: Debug logs are not set up or have been disabled or reset.

Record status	**Record created**: You added this record to the Logging Assistant with **Add a New Logging Assistant Record**. **Debug logging set up**: You completed the Setup Debug Logging Wizard for this record. **Debug Logging partially set up**: You set up debug logging for all the selected hosts and processes with the Setup Debug Logging Wizard, but not all the hosts and processes have been verified. **Debug logging set to minimum**: You completed the Set Minimum Debug Logging Wizard for this record. **Debug logging disabled**: You completed the Disable Debug Logging Wizard for this record. **Debug logs collected**: You completed the Collect Debug Logs Wizard. **nbsu output collected**: You have completed the Collect nbsu Output Wizard.
Progress	The progress of the current activity being performed for the Logging Assistant record.
Staging Directory	The output location where the debug logs and other troubleshooting information is collected.
Record creation time	The date and time you created this record.
Record last modified	The date and time that you performed the most recent activity on the record.
Debug logging set up time	The date and time when you completed the Setup Debug Logging Wizard.

Adding or deleting a Logging Assistant record

Create a Logging Assistant record that you can use throughout the process of troubleshooting a NetBackup problem.

To create a new Logging Assistant record

1. In the **NetBackup Administration Console**, in the left pane, select the **Logging Assistant** node.
2. In the **Actions** menu, select **New Logging Assistant Record**.

3 In the **Logging Assistant Record ID** field, enter a unique descriptive name for the new record. Use numeric, alphabetic, plus, minus, underscore, and period characters. Do not use the minus (-) symbol as the first character or leave spaces between characters.

   Veritas recommends that you use the Support Case ID as the **Logging Assistant Record ID** to more easily track the record through the troubleshooting process.

4 In the optional **Description** field, enter a problem summary and the Job ID of the failed job (if applicable).

5 Under **Actions**, select an action:

   - **Collect nbsu diagnostic information**: Select this option to immediately display the Collect nbsu Output Wizard when you leave this dialog. This is often the first action to take after you create the Logging Assistant Record.
   - **Setup debug logging**: Select this option to immediately display the Setup Debug Logging Wizard after you leave this dialog.
     See "Setting up debug logging" on page 177.
   - **No action, only create a record**: Select this option to return to the **NetBackup Administration Console** after you click **OK**. The new record appears in the console.

6 Click **OK** to create a Logging Assistant record in its database and add it to the list of records in the Logging Assistant Records pane.

Deleting a Logging Assistant record

You can delete the Logging Assistant record if you no longer require it.

Warning: Before deleting the record, ensure that logging is disabled or set to minimum.

To delete a Logging Assistant record

1 In the **NetBackup Administration Console**, in the left pane, select the **Logging Assistant** node.

2 The right pane of the **NetBackup Administration Console** displays the list of Logging Assistant records. Select the record that you want to delete.

3 In the **Actions** menu, select **Delete Record**. The **Delete Logging Assistant Record** dialog box displays.

4 Click **Yes** to delete the selected Logging Assistant record.

Setting up debug logging

Use the Setup Debug Logging Wizard to enable selected NetBackup debug logging that Technical Support uses to troubleshoot NetBackup problems. The Logging Assistant automatically creates the necessary log folders and sets the verbosity of the pertinent logs to their highest level.

Note: Ensure that each of the selected hosts contains enough available space for the selected debug logs.

To set up debug logging

1. In the **NetBackup Administration Console**, select **Logging Assistant**.
2. Select the Logging Assistant record for which you want to set up debug logging.
3. Select **Actions > Setup Debug Logging**.

   Click **Next**.
4. Select one of the following:
   - Enable the **Analyze job to identify relevant hosts and debug logging** parameter, and then enter the **Job ID**. The Logging Assistant identifies the hosts and the pertinent debug logs related to the job.

     The Logging Assistant identifies and selects debug logs for hosts and the processes that enable the most effective troubleshooting of this problem.
   - For the following cases, skip this step that sets up the logs based on a Job ID, and click **Next**:
     - The problem does not involve a particular NetBackup job.
     - You already know the hosts and debug logs that you want to enable.
     - You want to enable the logs that you previously set up for this record. Those log selections appear automatically in the panels that follow.
5. Do one or more of the following:
   - Click **Setup debug logging on Master Server** to set up debug logging on the master server. The **Master Server** text window displays the master server.
   - Click **Setup debug logging on Media Server(s)** to select the media servers. If the master server is also a media server, you may still want to select it again as a media server.
   - Click **Setup debug logging on Client(s)** to set up debug logging on clients. In the text window under this parameter, enter the names of clients for which

you want to set up debug logging. Separate the client names by commas. Do not use any spaces between the client names.

6 Follow the prompts in the wizard.

The Logging Assistant validates the specified hosts to establish connectivity and validates the NetBackup versions that are installed on the hosts.

7 If you selected **Setup debug logging on Master Server**, a list displays of the problem categories on the master server and the process names that are associated with each category of problem. Click the problem category or categories that apply to the failed job.

For example, for problems with synthetic backups on the failed job, select **Backup - Synthetic**. The debug logs for `bpcd`, `bpdm`, and `bptm` are enabled.

Click **If required, specify additional components to set up debug logging** to enter components. For example, if you have Java interface problems, but only want the debug logs for `jnbSA`, instead of all four of the processes lists, enter `jnbSA` in the text window.

Technical Support publishes a list of valid components that you can specify in this text box.

http://www.veritas.com/docs/TECH204256

Click **Next** to continue.

8 If applicable, perform the same actions for the media server and the clients.

Click **Next** to continue.

9 Review the summary for the hosts that you selected for logging.

Click **Next**.

10 Follow the prompts to complete the wizard.

Set minimum debug logging

Use the Set Minimum Debug Logging Wizard to set the minimum logging levels for an existing Logging Assistant record.

Information about the logging levels that this wizard configures is available in the following document:

http://www.veritas.com/docs/TECH204256

To set minimum debug logging

1. In the **NetBackup Administration Console**, select the **Logging Assistant** node.
2. Select the record whose debug logging you want to set to minimum.
3. Select **Actions > Set Minimum Debug Log**.

   Click **Next** to continue.
4. Confirm the host problem categories and processes for all debug logs that were enabled. The hosts can be the master server, one or more media servers, and one or more clients.

   Click **Next** to continue.
5. Follow the prompts to complete the wizard.

Disabling debug logging

Use the Disable Debug Logging Wizard to disable logging for an existing Logging Assistant record.

To disable debug logging

1. In the **NetBackup Administration Console**, select the **Logging Assistant** node.
2. Select the record whose debug logging you want to disable.
3. Select **Actions > Disable Debug Logs**.
4. Following the prompts in the wizard.