Open Systems SnapVault® (OSSV) Best Practices Guide

Darrin Chapman, Jeremy Merrill, Network Appliance, Inc.
May 2006 | TR-3466

Abstract

The following document is intended to serve as a deployment guide for successfully architecting and deploying Open Systems SnapVault in a customer environment. This OSSV deployment guide will describe backing up and restoring data that resides on systems other than NetApp (open systems) to a NetApp NearStore® system or secondary system utilizing NetApp SnapVault technology. As always, please refer to the latest release notes available on the NOW™ site for updates and the latest requirements, issues, and limitations. This document is intended for field personnel requiring assistance deploying and architecting an OSSV solution.

Table of Contents

Table of Figures

1) Introduction

The Open Systems SnapVault (OSSV) primary agent has extended the reach of Network Appliance™ SnapVault technology to the open systems server. OSSV facilitates block-level incremental transfers from the open systems platform directly to a secondary storage system. Various data sets can now be maintained remotely on a common NearStore system or secondary platform.

The following information describes the OSSV theory of operation, major features and enhancements, management options, typical deployments, and finally best practices for deploying OSSV in an enterprise environment.

Various appendices describing tools and scenarios to consider when deploying an OSSV solution are included.

Requirements and Assumptions

For the methods and procedures described in this document to be useful to the reader, several assumptions are made:

- The reader has at least basic knowledge of backup and recovery in a tape and/or disk environment.
- The reader has at least basic UNIX® and Windows® administration skills, has access to the administrative login for the server, and has administrative access to the server console.
- The reader has at least basic Network Appliance administration skills and has administrative access to the storage system or NearStore system via the command-line interface.
- The secondary system has the licenses necessary to perform the activities outlined in this document. Specifically, the storage system or NearStore system will need the SnapVault secondary and SnapVault primary licenses installed.
- The NetApp secondary system has the required block-level storage or network protocol interconnects to perform the activities outlined in this document.

In the examples in this report, all administrative commands are performed at the server, storage system, or NearStore console for clarity. Web-based management tools (DFM or NetVault) can also be used but are not demonstrated throughout the document.

Intended Audience

The information in this document is intended for field personnel responsible for architecting and deploying successful Open Systems SnapVault solutions. A brief overview of OSSV basics is presented in order to establish baseline knowledge before migrating toward the specific features, best practices, and finally the actual installation and configuration.

2) Overview

Open Systems SnapVault is a heterogeneous disk-to-disk data protection solution ideal for use with NearStore nearline storage systems. An OSSV primary system corresponds to a backup client in the traditional backup architecture. The SnapVault secondary is always a data storage system running Data ONTAP®, such as a NearStore system. OSSV software protects data residing on a primary,

which can be a storage system from a server running an operating system from leading server vendors such as Solaris™, HP-UX, AIX, Windows, IRIX, and Linux®.

Three main components are installed within the OSSV environment:

- The primary system
- The OSSV agent residing on the primary system
- The secondary system

A predetermined directory or file system is chosen to be backed up to nearline storage. The data set is mapped to a secondary system qtree on a NetApp NearStore system or secondary storage system. Once the data to be protected is identified and a destination volume/qtree is chosen as the secondary, the agent can be installed on the primary system and the basic parameters configured. During installation, the agent installs various subdirectories on the open systems server. Various components installed on the primary system include:

- OSSV primary database
- Set of OSSV executables
- OSSV log file
- OSSV exclude list files

Theory of Operation

There are two phases to the OSSV backup mechanism:

- Phase I: file system scan on primary and directory structure built on secondary
- Phase II: actual data set transfer

After a successful baseline transfer, OSSV operates by examining files for changes via two methods: modification time and block checksums. The modification time is a coarse estimation of the true amount of changed data, due to the fact that the modification time is updated when at least one block of the file is written. By using 4kB block checksums, OSSV is able to back up only the portions of the file that have changed. This is referred to as block-level incrementals, or BLI. OSSV can back up whole files or changed blocks, depending on user requirements. In all cases, only changed data blocks are sent to the secondary system.

In a block-level incremental deployment, OSSV can significantly reduce the amount of network traffic over traditional backup strategies by sending only incremental changes in increments of 4kB data blocks. Once the initial baseline transfer is complete, OSSV will send only changed blocks, effectively resulting in an “incremental forever” strategy. Remote office backups, especially those over slower wide area networks, are now easier to achieve with the introduction of OSSV agents at the remote office. These remote offices can be backed up to a central location such as a data center.

However, environments with faster network connections and/or high change rates in many large files may benefit more with the block-level incremental option turned *off,* resulting in entire file transfers.

In addition to block-level incremental backups, OSSV introduces other critical features into the backup environment. These include open file backups, health checks, checkpoint restarts, exclude lists, and system state backup and restore.

OSSV deployment consists of various steps, including schedule determination, change rate determination, retention policy, performing a baseline or level-0 backup, volume creation and sizing, and relationship creation. The actual relationship is created during the baseline transfer.

Relationship Creation and Baseline Transfer

In response to command-line or NDMP-based management interface input, the SnapVault secondary storage system (storage system or NearStore system) requests initial baseline (entire file system requiring backup) image transfers of directories specified for backup from an open systems platform. These transfers establish SnapVault relationships between the open systems platform *directories* and the SnapVault secondary *qtrees*.

The open systems platform, when prompted by the secondary storage system, transfers initial base images of specified directories to a qtree location on the secondary storage system. Once the baseline transfer has completed, the secondary system will create a Snapshot™ copy (baseline) of the volume containing the destination qtree. If multiple transfers are occurring, faster transfers will be in a "quiescing" state until ALL transfers have completed.

A new Snapshot copy is created each time a baseline is performed, and up to 250 Snapshot copies can be maintained according to a schedule configured by the backup administrator. Each Snapshot copy consumes an amount of disk space equal to the differences between it and the previous Snapshot copy.

Scheduling and Retention Policy Considerations

In a typical legacy backup environment, incremental backups were usually performed once per day, with full backups once a week. The fastest restore could take hours and required intervention of a backup operator or a system administrator. In the OSSV or SnapVault configuration, incremental backups are performed as often as once per hour (with daily and weekly options); each incremental backup is usable as if it were a full backup, and most restores can be performed in minutes or less by end users, without the need for backup operator intervention or use of a backup server.

Two scheduling options are available today: command-line scheduling in Data ONTAP and an NDMP-based scheduling mechanism utilizing DataFabric® Manager or another supported NDMP management tool.

To determine the proper scheduling and retention policies for a particular environment, it is important to understand the backup and restore requirements of an organization. Several factors must be considered, including restore granularity, media costs, data change rates, types of data to be protected, and risk.

An excellent resource for determining schedules and retention policies is the SnapVault Deployment and Configuration Guide.

3) OSSV 2.x Feature Review

In this section, we will review several of the features included with OSSV (but not all of them).

Block-Level Incrementals

Open Systems SnapVault BLI backup is designed to minimize the backup of data that has not changed since a previous backup operation. OSSV uses checksums to identify portions of a file that have changed between a previous and the current backup.

A BLI backup recognizes that a file has changed based on a time stamp and checksum algorithm. Exactly which blocks have changed is determined, and only those blocks are sent to the secondary storage system.

Typically, incremental backups are more frequent, reduce the amount of time required to back up data, and minimize the resources required to perform backups when compared to baseline or full backups. BLI significantly reduces the amount of data that needs to be transferred to backup storage as well as the amount of data that must be stored on backup storage disk.

Changed blocks are recognized based on checksum values calculated and preserved for each block by the OSSV agent. Checksums are calculated on 4kB blocks of file data stored on an internal database. These checksum database files are stored in the OSSV internal DB directory. Each relationship has its own checksum file directories. *Approximately 2%* of the baseline ends up being your checksum file database size.

First, time stamps of files are compared to the time of last successful backup operation. After being identified, a checksum is performed on that file. By default, every block of every file has a checksum operation performed against it during baseline operations. This is referred to as “high” BLI and results in typically longer transfer times and more CPU and disk consumption on the primary system. You can configure block incremental processing to trade off efficiencies among four variables: primary system CPU utilization, disk consumption, network bandwidth utilization, and Open Systems SnapVault transfer time. Enabling block-level incremental updates normally causes a checksum value to be calculated for every block of every file during the initial Open Systems SnapVault baseline transfer. As a result, baseline transfer execution time, CPU utilization, disk consumption, and network bandwidth utilization are increased compared to incremental transfers that are not block related. In this case, it is possible that significant resources can be consumed by calculating checksum values for static files that never change. The checksum levels can be configured as high, low, or off, using the `svconfigurator` utility.

- **High.** Always computes checksums, on baseline transfers and incremental updates.
  Primary impact: High CPU utilization on the baseline transfer
  Network impact: Lower amount of data transferred on updates (incremental)
  Secondary impact: Same
  When to implement: If all files are subject to small changes
- **Low.** Compute checksums on changed files and only on updates; no checksum performed during baseline.
  Primary impact: More CPU utilization during updates, faster baseline transfer
  Network impact: Large amount of data transferred during baseline, lower amounts of data transferred during updates
  Secondary impact: Same
  When to implement: If a small subset of files is likely to change
- **Off**. No checksums are calculated at any time. Similar to older versions of OSSV. Full files are transferred once identified as being changed files.

Primary impact: Fast baseline transfer, less impact on CPU during file system scan
Network impact: Large amount of data transferred during baseline, potentially large amounts of data (large files) during updates as well
Secondary impact: Same
When to implement: If a small subset of files is likely to change or files are changing completely

Name-Based BLI

In some cases, applications modify files by:

1. Creating a temporary copy of the original file

2. Making the necessary changes to that temporary file

3. Deleting the original file

4. Saving the temporary file under the same name as the original file

OSSV can detect this condition and treat the new instance of the renamed temporary file as the updated original file without having to transfer the entire file.

In other cases, applications make changes to files by:

1. Inserting data into or removing data from the middle of the file

2. Rewriting all subsequent data blocks in the file to new positions in the file

Microsoft® Word, Excel, and PowerPoint are some of the applications that are known to exhibit this behavior when saving changes to files. For files that are modified in this manner, OSSV backs up all blocks in the file that have different positions or different checksum values.

Open File Backup

In a Microsoft Windows environment, there are currently two options for ensuring open files are backed up successfully: Open File Manager (OFM) and Microsoft Virtual Shadow Copy Service (VSS).

The Open File Manager (OFM) utility allows Windows NT® and Windows 2000 files that are open and in use to be backed up with only a very short disruption to users or their current applications. OFM is automatically installed in your system at the same time the proper agent is installed. It will not be enabled unless the Windows 2000 server is rebooted and the OFM component has been licensed on the secondary system.
NOTE: This is a separate license on the secondary system, and it must be installed in order for OFM to operate properly.

Windows 2003 provides a native snapshot mechanism as part of the VSS. VSS snapshot functionality (called shadow copy) is integrated with the OSSV agent as a standard feature. No VSS installation is required; although there are configurable parameters, there aren't any required configuration steps for OSSV and VSS integration. All Windows 2003 Open Systems SnapVault agent backups use VSS unless specifically disabled from the secondary using the `back_up_open_files=off` option. This option can be used with OFM as well.

Both open file backup components can be tuned using the `svconfigurator` tool built into the OSSV primary system.

OFM Configuration Options

OFM waits for a set time when write activity is quiet and the system is in a safe state to initiate the backup of open files. OFM listens continuously for a period of write inactivity until it is ready to initiate a synchronize for backup or until the preset synchronization timeout period has expired. This value is configurable in svconfigurator under the “SnapVault” tab; the parameter is referred to as the “OFM Write Inactivity Period (seconds)” and defaults to three seconds. The minimum value is one second; the maximum value is 60 seconds. OFM will wait this period of time for write inactivity; if it cannot find a write inactivity period and has tried for 60 seconds, it will fail. This type of failure can result in a blank backup.

This default 60-second timer is known as the “Maximum time to wait for OFM synchronization (seconds)” and can be found under the “SnapVault” tab in `svconfigurator`. Again, the range is one to 60 seconds.

Specific drives can be excluded from open file backup using OFM if required. There is an option within `svconfigurator` that allows this exclusion (“SnapVault” tab).

NOTE 1: OFM supports one active Snapshot copy per disk volume letter. For example, if you try to use OFM to back up two directories on the same disk volume, one of the two directory backups will fail. If an active Snapshot copy is encountered during an attempted backup, OFM fails.

NOTE 2: Approximately 15% free space is required on the drive being copied using OFM. If you are running OSSV 2.2 or greater, you can use the Free Space Estimator to determine if there is sufficient free space to run OFM.

VSS Configuration Options

Certain conditions must be met before the OSSV agent can acquire a VSS snapshot. You can set the amount of time (snapshot timeout) that the agent waits until it retries a VSS snapshot if the conditions are not right at the time. Setting this parameter avoids unacceptably long waiting periods. The default is 180 seconds (the maximum value). The minimum value is one second.

Like OFM, the VSS values can be modified using `svconfigurator` under the “SnapVault” tab.

NOTE: When using OFM/VSS with OSSV, the snapshots are placed in the following “pseudo” file systems:

OFM: \\?\Z:
VSS: \\?\GLOBALROOT\Device\Harddisk\VolumeShadowCopy42

However, the snapshots in these locations are discarded after the transfer, and OSSV can’t be used to back up the snapshot locations. OSSV handles all interaction with OFM/VSS.

Checkpoint Restart

Checkpoint restarts allow a user to restart a failed baseline transfer (OSSV 2.0) and a failed incremental transfer (OSSV 2.1). Checkpoints are taken every five minutes. The restart may occur at these five-minute intervals, although they may span multiple five-minute intervals and go back further into the transfer than expected. The reason for the unexpected restart of longer than five minutes is due to the fact that the restarts occur at file boundaries. If a large file spans multiple five-minute intervals, the restart will go back further.

Checkpoint restarts only occur during phase II (data send phase) of the OSSV transfer.

NOTE: This option is not a replacement for resyncing relationships in a disaster recovery scenario.

Backup Exclusion Lists

This feature was introduced in OSSV 2.1 and allows the user to exclude files and paths from backup. There are two files that are installed once the OSSV 2.1 agent is in place on the primary system. The `file-exclude.txt file` contains file exclusion configuration information with wildcard capabilities. The `path-exclude.txt` file contains path exclusion information where full directories and their contents can be excluded. Both files are located under the `$INSTALL_DIR\etc` directory by default.

A file or directory is excluded if the filename or any path element matches a file exclusion entry in the list in one of these two files.

On Windows systems, exclusion list files are Unicode text files. On UNIX systems, exclusion list files are multibyte text files. Each entry is on its own line. Wildcard characters are supported. Use an asterisk (*) to specify any number of characters within a single path element. Use a question mark (?) to specify one character within a single path element. Use an exclamation mark (!) to remove the special meaning from * or ?. Use a pound sign (#) at the beginning of a line for a comment.

NOTE: See the latest OSSV 2.2 release notes for more details (available on NOW), or open these files for various notes and comments.

System State Backup and Restore

Various components, including the REGISTRY and system files, can now be backed up and restored using the OSSV 2.1 agent. The components included in the system state backup will vary depending on the operating system, installed applications, and configuration. These components may include:

- REGISTRY
- System files and settings, including the boot files
- System files that are under Windows file protection
- Certificate services database
- IIS metadirectory
- Various performance counters
- Active Directory and SYSVOL data (domain controllers)

Essentially, replacing the primary file system path with the keyword “SystemState” will initiate a system state backup of the primary:

```
snapvault start -S ossv_prim:SystemState sv_secondary:/vol/sec_vol/sec_qtree
```

Boot files and system files are backed up even when they are on different volumes. Subsequent backups use block-level incremental backups.

NOTE: On Windows 2000 systems, before starting system state data backups, make sure the system has recent Windows service packs installed. You can have problems with restored systems if the backed-up system was running an older version of the operating system.

Restores also use the keyword “SystemState” in place of the file system path:

```
snapvault restore -S sv_secondary:/vol/sec_vol/sec_qtree SystemState
```

In certain Active Directory environments, there are other options for restoring system state data. The keyword may change to `SystemStatePrimary` in instances when restoring system data from a backup and marking it primary.

For more information on system state restores, please refer to the latest OSSV release notes, available on NOW.

You can also use system state backups as part of a disaster recovery plan. To create a backup for use as part of a disaster recovery plan, you would essentially back up the entire system drive and any other relevant partitions or drives and back up the system state. Be aware that when recovering from a disaster using a complete system backup, we do not support “bare metal restore.” The user would need to ensure that the base operating system is installed on an identical hardware configuration with identical service packs, names, drive letter mappings, file system types, etc. For more details on backing up and restoring a complete system for disaster recovery purposes, please refer to the latest OSSV release notes, available on NOW.

OSSV Database Backup

With older versions of the OSSV Client database, the database backup is a manual process. There are two options for backing up the database in older versions. The user can simply include the `$INSTALL_DR/db` path in the backup stream. In addition, the `svdb` command can be used, which will create a file under the `$INSTALL_DR/db` directory.

With OSSV 2.2 the OSSV Database backup is now performed automatically with every OSSV transfer. There are three configuration levels for the client database. “BLI” backs up the history file and its corresponding BLI checksum file, “DB only” backs up only the history file, and “None” disables the automatic database backup. The options can be set from the `svconfigurator` utility, under the SnapVault tab. The database is created under the qtree root as `.OSSV_Database` on the secondary. To restore this database, issue a `snapvault restore` command on the primary. When performing the restore, specify any destination, such as \temp\database, and OSSV will automatically detect the database, decode it, and place in the appropriate directory.

Checker/Fixer

Checker/Fixer is a new feature for OSSV, which is executed on the secondary. Checker/Fixer is a diagnostic command in Data ONTAP. In order to run Checker/Fixer, it must be enabled in the `snapvault.cfg` file on the primary by changing the key `QSM:GenerateVerifyChecksums` to `TRUE`.

Checker is used to check all the data in the last backup for a given relationship against the data set on the secondary and generates the differences. If a discrepancy is found between the primary and secondary, Fixer can then be used to make the image on the secondary in sync with the image on the primary.

NOTE: Checker/Fixer will be available in Data ONTAP versions greater than 7.1; please see NOW for the latest available version of Data ONTAP.

Free Space Estimator

Free Space Estimator is a command on the primary (`svestimator`) in OSSV 2.2 that enables you to determine if there is sufficient disk space available on the primary to perform an OSSV transfer. Free Space Estimator can be run on a primary with or without OSSV already installed. It can help determine if a specified drive has the appropriate space available to install OSSV on a new client.

There are two modes for the Free Space Estimator, built-in mode and standalone mode. In built-in mode, Free Space Estimator runs in the background, at the start of every OSSV transfer, and reports whether there is sufficient space to back up based on the current OSSV configuration. The results are recorded in the `$INSTALL_DIR/etc` directory. If sufficient space isn't found, the operation is not aborted by default. However, this can be changed by modifying the `snapvault.cfg` file. In addition, it is possible to disable Free Space Estimator from running automatically by disabling it in the `svconfigurator` utility under the SnapVault tab. In standalone mode, Free Space Estimator is installed as a standalone application on a system that may or may not have OSSV installed. This utility can be useful if you are planning on deploying OSSV and need to know how much free space is available and which directories can handle the OSSV installation.

NOTE: Free Space Estimator is set to run prior to any OSSV transfer. If the file system to be backed up is large, Free Space Estimator may take an extended period of time to estimate space requirements, causing the secondary to time out before the backup can start. In this case, it's recommended that Free Space Estimator be disabled prior to the OSSV transfer.

Unattended Install

Unattended install enables you to install or upgrade OSSV software on a primary storage system with minimal user intervention. Although this new feature does assist in creating an install image, it is not 100% hands off. This feature is useful if deploying over a large number of primary systems. Unattended install allows you to set the installation variables noninteractively, and in most cases, a reboot is not required after the install. In order to perform an unattended install, an installation script and other supporting files are required. To gather these files, use the utility `svconfigpackager,` available with OSSV. This utility saves the current configuration settings to file when run on a primary. In addition, it creates an installation script that in conjunction with the configuration settings file and other files can be used to perform unattended installs or upgrades.

When creating an unattended install package, the installation script and other files created by the svconfigpackager utility on an operating system cannot be used for running an unattended installation on a different operating system. For example, an unattended install created on a Windows 2000 server cannot be used on a Windows 2003 server. In addition, there are separate installation scripts for Solaris and HP-UX. For more information, please see the "Unattended Install" example in section 7, labeled "Creating an Unattended Install Image."

NOTE: Unattended install is only supported on systems running OSSV 2.x. OSSV 1.x agents cannot be upgraded using this technique.

Resync after Restore/Break

This feature is available with OSSV 2.2. Resync after Restore/Break allows the user to resynchronize a relationship without requiring a new baseline transfer. Previous to OSSV 2.2, if a relationship got out of sync, a new baseline would be required. A system is considered synchronized as long as a common snapshot exists between the primary and secondary. Once this is lost, the incremental backups begin to fail. There are three ways an OSSV relationship can get out of sync:

1. An older version of the OSSV database is restored to the primary.
2. Data is restored using the `snapvault restore` command.
3. The state of the destination qtree in an OSSV relationship is changed to read-writable (even if the contents in the qtree were not modified). **NOTE**: If the contents of the qtree were modified, all data written to the qtree will be lost on the resync.

To resync a relationship, use the `snapvault start -r` command from the secondary.

NOTE: The resync after restore/break will resynchronize the relationship between the primary and secondary. This will not propagate changes back to the primary; it is not similar to the resync in SnapMirror®.

NOTE: Resync After Restore/Break is not available until versions greater than Data ONTAP 7.1; please see NOW for the latest version of Data ONTAP.

LREP (Logical Replication)

In the OSSV 2.2 install bundle, LREP is now included. Offices that are small enough to have name, print, and file service requirements met by a single server typically have limited WAN bandwidth. OSSV is a strong solution for remote office backup, but the initial transfer can be crippling. LREP can be used to seed the baseline on the secondary.

Use the LREP utility to write the initial transfer (the baseline) to a portable drive (Zip drive, etc,). The portable hardware could also be a FAS250 shared between offices. Ship the portable media to the data center and locally write to the secondary system. No network bandwidth is used, only a manual process of moving the media from remote site to data center. Once the data is on the secondary system, modify the OSSV relationship to reflect the real primary → secondary relationship.

In addition, LREP can be used in conjunction with encryption and compression utilities and sent over the network. Once it is transferred, the data can then be decrypted and uncompressed at the destination. You can then update the OSSV relationships to reflect the correct primary → secondary relationship.

There are two utilities required for the entire LREP process: the lrep_reader and lrep_writer. lrep_writer is used at the data center or location of the secondary system, lrep_reader at the remote office.

There is an example of how to use LREP in the appendix.

NOTE: If you use LREP to do a restore to the primary, you cannot use the “resync” capability of OSSV 2.2 to reestablish the relationship.

4) Management Options

Open Systems SnapVault can be managed from a variety of NDMP-based applications. Once the primary agent has been installed, the schedules and retention policies have been determined, the network is in place including all firewall settings if any, and the directory to qtree mappings has been laid out, a management system can be introduced acting as a third party in a primary-to-secondary relationship. The management system communicates over an available TCP/IP network interconnecting the primary, secondary, and management system. In all cases, NDMP is used as a transport for SnapVault messages and commands. All scheduling, baselines, relationship creations, retention policies, backup control, and monitoring can be centrally configured on the management system—all utilizing NDMP over the IP network.

The current supported management applications are:

- Network Appliance DataFabric Manager 2.2 and higher (OSSV 2.2 and later requires DFM 3.2R1 or later)
- BakBone NetVault 7.x and higher (OSSV 2.2 and later require NetVault 7.4 or later)
- Syncsort Backup Express 2.2 and higher
- CommVault QiNetix 5.9 (with OSSV 2.2)

In addition to these GUI-based applications, OSSV can be managed via the Data ONTAP CLI. For the purposes of this document, we will use the Data ONTAP CLI.

DataFabric Manager

Unlike homegrown scripts and competitive products, only DataFabric Manager takes full advantage of the Network Appliance APIs and industry standards to deliver a full suite of storage management capabilities for enterprise storage and content delivery infrastructures.

In order to enable DFM management of SnapVault and Open Systems SnapVault, the business continuance option much be purchased and added to the DFM installation. DFM utilizes NDMP to access the primary and secondary systems. TCP port 10000 must be open if firewalls exist. If multiple interfaces exist on the DFM server, NDMP-preferred interfaces can be utilized.

DFM 3.2 system requirements:

Windows 2000 (SP2 or greater) or Windows 2003

- PC based on Intel® with single 2-GHz CPU (Xeon or Pentium® 4)
- At least 4GB free disk space (8GB recommended)
- At least 512MB memory

Solaris 8 (Patch 108827 or higher) or 9 for SPARC™

- Single UltraSparc IIIi processor at 1 GHz (such as Sun™ Fire V120 or V240)
- At least 4GB free disk space (8GB recommended)
- At least 1GB memory

Linux workstation or server

- PC based on Intel with single 2-GHz CPU (Xeon or Pentium 4)
- At least 4GB free space (8GB recommended)
- At least 512MB memory

When baselines and relationships preexist, DFM allows the user to import these relationships for management, avoiding a costly full baseline transfer again. The DFM backup manager will recognize the relationship exists once it is selected for backup.

All scheduling can be performed in DFM. Multiple schedules can exist and retention policies placed on these. If a schedule preexists from the Data ONTAP command line and DFM is introduced later, limit the scheduling mechanism to only one of the two options. Two separate scheduling mechanisms can cause confusion and may interfere with other backups. The preexisting Data ONTAP schedules cannot be imported, similar to the relationships.

When creating a new relationship in DFM, there is no need to create a qtree name. DFM will create a unique qtree name. Please be aware of this when specific naming schemes are in place.

DFM is an excellent tool for data restoration. The ability to browse DFM-created backups on the secondary for single files, subdirectories, and directories makes restoration much simpler.

NOTE 1: In some situations OSSV primaries do not pass enough error messages back to the backup manager when a restore operation fails. You can look at the error messages logged on the secondary storage system (/etc/log/snapmirror) and OSSV primary ($INSTALL_DIR/etc/snapvault).

NOTE 2: You cannot limit the bandwidth used by OSSV transfers that are triggered by DFM.

Beginning with DFM 3.2, you have the functionality to create pre- and postscripts. The scripts must be PERL scripts, and PERL 5.6 will need to be installed on the DFM Server. The scripts are installed using a zip file which contains:

- The script
- An XML file named package.xml which includes:
    - Packaging information (filename, script version…)
    - Privileges needed to run the script

By default, the scripts are installed in “script-plugins” in the root of the DFM install. The scripts can be run manually via DFM or by a schedule. These scripts can be useful to put a database into hot backup prior to an OSSV transfer, and then releasing it from hot backup mode upon completion.

For more details on DFM and DFM pre- and postscripts, please refer to the DFM documentation on NOW.

Command Line Interface

To manage OSSV relationships via the Data ONTAP CLI, you utilized the `snapvault` commands. The commands are the same as SnapVault, with the exception that you identify the OSSV Primary as the

source. All backup schedules and relationships are configured on the secondary. In addition, for any restores, you will use the CLI that is installed on the Primary.

Syncsort

Syncsort's Backup Express has been certified for NetApp Data ONTAP and is currently in collaborative development on Data ONTAP 7.0. Fully integrated Open Systems SnapVault management is available with Backup Express 2.2. Backup Express includes complete support for NetApp SnapVault, including OSSV management for Windows, Linux, and UNIX. In addition to Backup Express, Syncsort also provides its own OSSV agent. Backup Express can be used to manage both the NetApp and Syncsort OSSV agent. The Syncsort OSSV agent for Windows also includes integrated application support for Exchange and SQL server. In addition, Syncsort also provides a bare metal recovery option.

NOTE: While Syncsort does provide its own OSSV agent, it does not perform exactly the same as the NetApp OSSV agent. Syncsort creates images that it transfers, which can then be mounted as a snapshot-based LUN to select individual files or database objects for restore.

CommVault

The CommVault QiNetix suite, based on CommVault's Common Technology Engine, provides data protection by managing data throughout its lifecycle via integrated backup/recovery, migration, archiving, replication, and storage management. By adding in CommVault QiNetix QuickRecovery, you can enable backup and recovery of Exchange, SQL, and Oracle® with the NetApp OSSV agent. For more information, visit CommVault's Web page.

NOTE: Unlike Syncsort, CommVault does not have its own OSSV agent; it simply uses QiNetix Add-ons to manage OSSV.

5) Best Practices and Recommendations

The following lists of items are recommendations and common best practices. Many of these have been uncovered from field testing, others from lab testing. These items are not necessary for OSSV to completely function, but will definitely have an impact in terms of sizing, scalability, manageability, availability, configuration, and overall architecture. Please consider these items in all OSSV deployments. Most are well-tested issues and are highly recommended for production installations of Open Systems SnapVault.

Take Stock of Your Data

Identify all directories that will be backed up. Obtain estimates of how much data is contained in these directories. How frequently does this data need to be backed up? What is the rate of change of data? How large is the backup window? Can I kick off multiple backups simultaneously? How many volumes are available on my secondary? What is the size of the baseline data set? Does the data set have non-ASCII filenames, e.g., filenames containing Japanese characters? Repeat this for every machine to be backed up.

Caution: A very large number of small files (greater than 1 million) can have an adverse impact on performance and also result in a significantly large amount of overhead data being transferred during backups. SnapVault introduces ~16kB of overhead for every file being backed up.

Secondary Considerations

Use of Data ONTAP 6.5.1 or *later* is recommended, however; if running OSSV 2.2, you must use Data ONTAP 6.5.4. If you plan on using the block-level incremental feature of OSSV, it is recommended that

a minimum of OSSV 2.1 2 and Data ONTAP 6.5.2R1 be installed. At a minimum you will need an *sv_ontap_sec* license and a primary license. For Windows 2000 and Windows NT, an OFM license is recommended. Create secondary volumes and pay close attention to the size of volumes. They must have enough space to hold backups and Snapshot copies, sometimes many (up to 250). Check the "maxfiles" value of volume. If the backup contains or may contain non-ASCII filenames, the storage system's volume language should be changed to contain .UTF-8. Plan your backups—e.g., what primary directory will go into what secondary volume. Assign easy-to-read qtree names, consistent qtree names, and names that correspond to source file systems and hosts.

It is highly recommended to have all relationships that take roughly the *same transfer time* go into the *same volume*. For example, you may save weekly backups of all long-running transfers to one volume and nightly backups of short-running transfers to another volume. Mixing long and short transfers will result in the longer transfers holding up the shorter ones from being visible. For faster transfers, output of "quiescing" will appear in those situations where fast and slow transfers are mixed on the same volume. Those faster transfers will remain in a "quiescing" state until the slowest transfer completes; this can be viewed as holding up or slowing down the faster transfers.

Remember that schedules for SnapVault are on a per volume basis. One volume with many relationships will potentially cause a flood of backup traffic each time the scheduled backup time is reached. OSSV will attempt to back up all relationships in that volume based on that one schedule. Creating more than one volume will result in multiple schedules; each volume could then have its own schedule; and the schedules could be staggered.

While defining backup schedules, keep in mind SnapVault transfer limits. If all relationships are in one volume, which maintains one schedule, all transfers will occur concurrently. The R200 platform supports a maximum of 128 concurrent SnapVault transfers; with small numbers of volumes and large numbers of relationships you run the risk of exceeding the 128 limit. If the limit is exceeded, the remaining transfers are queued up and started as and when others complete. Other smaller platforms only support between four and 16 concurrent transfers; those platforms are limited especially when dealing with large numbers of SnapVault relationships.

NearStore Personality

In order to enable customers to utilize FAS storage systems as secondary storage, a new software license option called NearStore Personality (**nearstore_option)** has been introduced. This license option can be installed only on the FAS3020/3050 systems. This option is supported on Data ONTAP 7.1 and later versions. The goal of this license option is to provide increased concurrent streams when FAS3020/3050 storage systems are used as destinations for SnapMirror and SnapVault transfers and to enable SnapVault for NetBackup™. This license option should not be installed on these storage systems if they intend to handle primary application workloads.

Concurrent Replication Limits

The default Data ONTAP behavior without the **nearstore_option** license is to maintain a fixed upper limit for concurrent SnapMirror and SnapVault transfers based on the type of disks the storage system has attached. Without the license installed, the total concurrent SnapMirror and SnapVault transfer limits for the FAS3020/3050 are:

FAS3020:	16 concurrent streams
FAS3050:	16 concurrent streams

FAS3020 with ATA drives: 8 concurrent streams

FAS3050 with ATA drives: 8 concurrent streams

Note that the values listed above are the combined total of all source *and* destination transfers that can be concurrently run on the given platform. For example, this means that on a system with FC drives, if you have two QSM sources, six QSM destinations, and three VSM sources concurrently running, you can only start up five more SnapVault destinations. Please note that a replication operation within the same storage system is considered two concurrent streams.

In a clustered configuration, the maximum stream count shown above is **per controller**. So in a clustered FAS3020 configuration with FC drives where both controllers are active, each controller has a maximum limit of 16 streams. Once a controller is taken over by the other controller, there will only be 16 maximum streams available by the two controllers. During takeover/giveback, all transfers running on either the controller being taken over or the controller being given back will be aborted.

Once the **nearstore_option** license is installed, the storage system switches to NearStore Personality. When the storage systems take on NearStore Personality, Data ONTAP limits the maximum concurrent transfers based on the type of the replication operation. NearStore Personality also removes the above restrictions of eight concurrent streams for ATA drives.

The following table describes the maximum streams for each SnapVault operation.

Operation	**Maximum Streams FAS3020**	**Maximum Streams FAS3050**
SnapVault Source	16	16
SnapVault or OSSV Destination	32	64

Figure 1) Maximum streams with NearStore Personality

If you are using the FAS3050 with NearStore Personality as a QSM and SnapVault destination alone, you can have up to 64 concurrent streams. Without the **nearstore_option** license on the same FAS3050, you are limited to 16 concurrent streams with FC drives and eight with ATA drives. This is precisely a scenario where a FAS3050 with NearStore Personality would be beneficial to customers.

In a clustered configuration, all the values of maximum stream count shown above are **per controller**. So in a clustered FAS3020 configuration where both controllers are active, each controller has the maximum limit of 32 streams for QSM destination. Once a controller is taken over by the other controller, there will only be 32 maximum streams available by the two controllers. During takeover/giveback, all transfers running on either the controller being taken over or the controller being given back will be aborted.

Space Requirements

The OSSV agent requires space on the primary system for various components. If running a version earlier than OSSV 2.1, it is important to pay close attention to free space in the drive where OSSV is installed. All default OSSV subdirectories, including the `trace` and `db` subdirectories, are installed here. Open Systems SnapVault needs disk space for its built-in database. The database disk space requirements depend on the number of files and average files size and number of directories.

If block-level incremental backup is set to *OFF*:
Size of database = number of files and directories in backup data set times 96 bytes

If block-level incremental backup is set to *LOW* or *HIGH*:
Checksums require 16 bytes per 4K bytes of data

Size of database = (number of files and directories in backup data set x 96 bytes) + [total size of all files in the backup data set / (4K bytes x 16 bytes)]

Temporary space requirements are approximately twice the size of the database above.

If you are running OSSV 2.2 or greater, a simple way to determine your space requirements would be to use the Free Space Estimator, `svestimator`, to make sure you have enough space. It may be a good idea to run this on a scheduled basis (weekly, monthly, quarterly) to avoid any incidents.

NOTE: Please refer to the latest OSSV 2.x release notes, available on NOW, for the latest space requirements for the OSSV primary database.

In addition to built-in database space requirements, if you are using Open File Manager to manage open file backups by using Snapshot technology for a particular drive, OFM will need extra space for the drive for which it is creating a point-in-time copy. The recommendation is a minimum of 15% additional free space on those file systems that are being backed up. If free space is not available, disable OFM for those drives.

SnapVault Overhead
When using OSSV, there will always be some sort of overhead to be transferred for files in the OSSV relationship that have been modified. The OSSV Primary will send one 4kB header for every file/directory that exists in the relationship. In addition, for files/directories that are larger than 2MB, an additional 4kB header is transferred for every 2MB.

6) Other Considerations Prior to Deployment

Multiple Concurrent OSSV Transfers from the Same OSSV Primary

You should plan your backup schedules so that 16 or fewer transfers occur at the same time from the same primary system. To do this, create multiple schedules on multiple volumes on the secondary system or simply eliminate (by consolidation) the number of file systems being backed up from the primary system.

Reliable Disk I/0

Disk I/O errors could potentially cause problems with the OSSV primary database. Keep track of the system event log for logged disk errors. Many disk subsystem manufacturers also provide vendor-specific disk diagnostic utilities.

Low-Bandwidth Network Links

Customers that deploy OSSV in environments where a low-bandwidth network link separates the primary from the secondary have an aversion to repeating baseline transfers, for obvious reasons. These customers should perform frequent backups of the primary OSSV database to minimize the possibility of OSSV database corruption requiring a baseline transfer to be repeated. Since resync after restore break isn't available prior to OSSV 2.2, be sure to use `svdb` and backup of the local $INSTALL_DIR/db. If a baseline transfer takes an extremely long time, consider backing up the OSSV

database after every incremental. Ensure older database backups are deleted from the primary due to free space considerations. This isn't as much of an issue if you are running OSSV 2.2 due to the resync after restore/break functionality.

Source Data Considerations

A large number of small files may degrade performance and also result in a large amount of overhead data sent over the network. `snapvault status` output will reveal higher-than-expected data transfers once such a backup is complete.

If a large number of files are not likely to be modified, consider changing the BLI level to LOW. This will keep the OSSV database size in check.

Database Backups

An application database must be unmounted (shut down) before OSSV backups are initiated. OSSV performs a file-level backup in most database environments, since file modification times are constantly changing. OSSV is not integrated with any database backup APIs or prescripts or postscripts. The database files need to be dismounted (brought to a logically consistent state and closed) prior to using OSSV to back them up. If users want to use the "hot backup mode" method, they will need to script it and test it themselves and ensure that the procedure works reliably in their environment. It is crucial that an exact procedure is followed in the script.

In addition to scripts that reside on the hosts, if running DFM 3.2 or later, the pre- and postscripting capabilities can be utilized to manipulate the database before and after the OSSV transfer.

OSSV Primary Database Growth

OSSV database space consumption that is continuously increasing and is significantly greater than the database size that is *documented in the release notes* should be reported to technical support immediately. Upgrading to OSSV 2.1 is a first step to avoiding this issue. In addition, use of the Free Space Estimator included in OSSV 2.2 will help alleviate this issue.

7) Installation and Configuration

Primary System Platforms

Solaris

- 2.6, 7, 8, 9 on Sparc with native Solaris UFS file systems
    - Symantec® VxFS versions 3.5 and 4.0 is supported on Solaris 8 and 9

Windows

- Windows 2000 server and advanced server (32-bit only)
- Windows NT 4 standard server edition and enterprise edition with SP6
- Windows 2003 server standard edition and enterprise edition (32-bit only)

NOTE: OSSV 2.2 for Windows Server 2000 and 2003 has been tested and is supported when used on 64-bit Intel EMT64 and AMD Opteron processor based systems.

HP-UX

- HP-UX 11, or 11i (HP-UX 10.2 with OSSV 2.1)

AIX

- AIX 5.1, 5.2 or 5.3 (AIX 4.3.3 with OSSV 2.1)

Linux

- RedHat 7.1, 7.2, 7.3, 8.0 and 9; or AS 2.1; RH Enterprise Linux AS 3.0, ES 3.0, and WS 3.0

IRIX

- 6.5.x

Follow all recommendations in previous sections for calculating free space on the primary system. Do not forget about OSSV primary database requirements and Open File Manager space considerations (in a Windows environment). For specific memory, disk space, and other requirements, *please refer to the latest OSSV release notes, available on NOW.*

Licensing

All licensing is configured on the secondary system. For every open systems platform being backed up, a matching OSSV primary license will need to be purchased and installed on the secondary system. Various license pack sizes are available:

- Windows: `sv_windows_pri` (takes care of each supported Windows OS)
- UNIX: `sv_unix_pri` (takes care of each supported UNIX OS, including IRIX and AIX)
- Linux: `sv_linux_pri` (takes care of each supported Linux OS)

If you are utilizing Open File Manager (OFM) in a Windows 2000 environment, a separate license will need to be installed (in addition to the base Windows license).

- OFM: `sv_windows_ofm_pri`

The secondary system(s) will also need to be licensed as secondaries.

- SnapVault Secondary: `sv_ontap_sec`

In addition, if you are using the NearStore Personality on your secondary system(s):

- NearStore Option: `nearstore_option`

For more information, *please refer to the latest OSSV release notes, available on NOW.*

Secondary System Requirements

The secondary system should be running a minimum of Data ONTAP 6.5.. For more information on the latest version of Data ONTAP, *please refer to the latest OSSV release notes, available on NOW.*

Basic configurations:

```
options snapvault.enable on
```

`options snapvault.access all` *(or list of IP addresses/hostnames allowed to back up to this system)*

Firewall

If there is a port filter or firewall between the OSSV primary and secondary, ensure that TCP port 10566 (the SVListener port) is open. If you are utilizing DFM or another NDMP-based management tool to manage the OSSV relationship, ensure that TCP port 10000 (the default NDMP port) is open as well.

Port 10566 will need to be open in both directions during restore operations.

Running the Free Space Estimator

In this example, we will be running the Free Space Estimator on a Windows 2003 system (the process is the same for other platforms). There are three options for the `svestimator` command:

- '-o' if OSSV is already installed, if so it will use the current OSSV setting configured in the `svconfigurator` utility
- '-i' will include the OSSV installation package size in its calculation (this is the option to use when determining if OSSV can be installed on a new server)
- '-d' to output debug trace to a directory that will be created in the current directory

In order to run, Free Space Estimator requires two files to properly estimate free disk space: the path and file exclusion lists (located in `$INSTALL_DIR/etc`) and a configuration file named `estimator.cfg`.

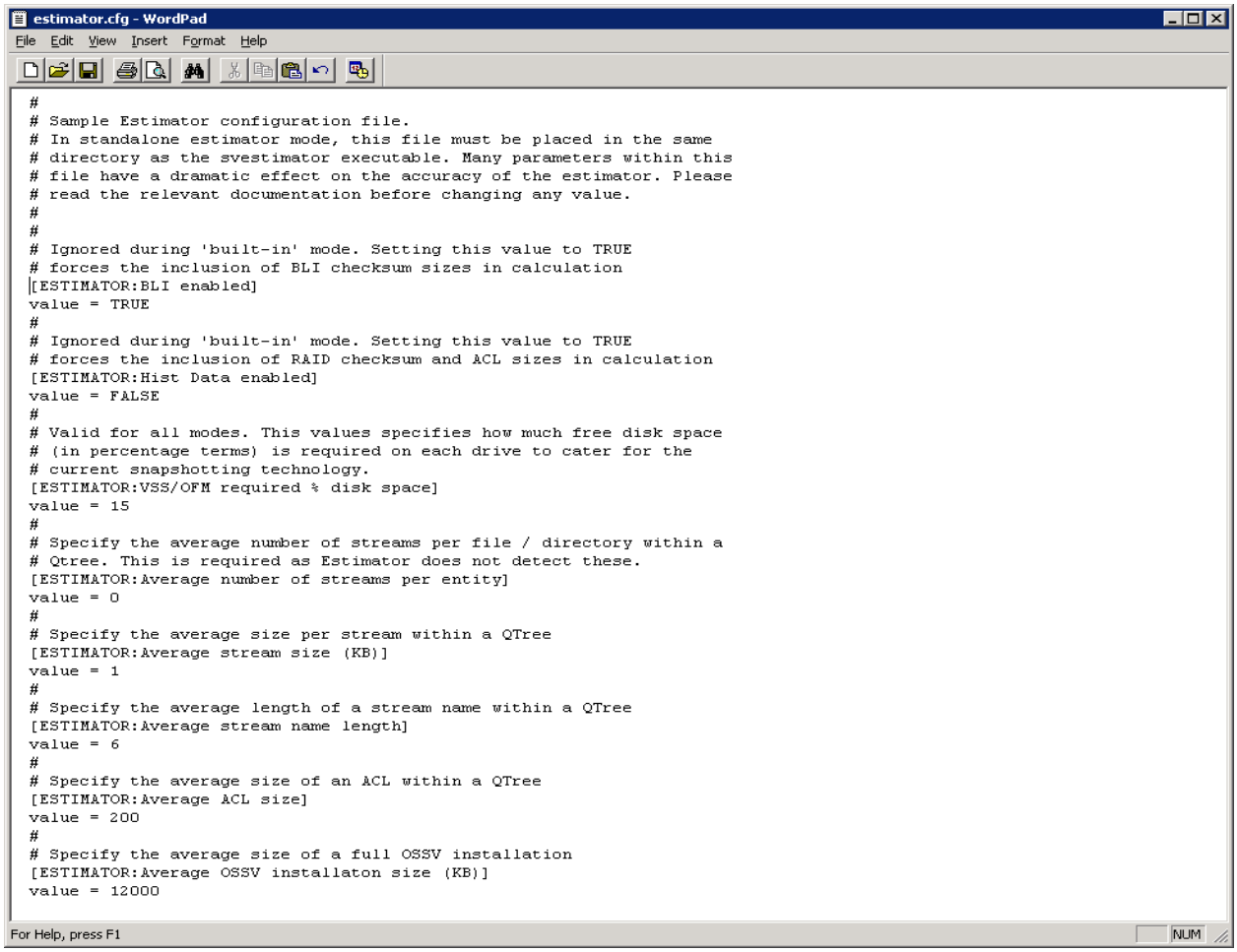

Figure 2) estimator.cfg file

The `estimator.cfg` file contains user-defined options that are taken into consideration when estimating free space (located in `$INSTALL_DIR/config`). On a standalone you must create this file and the path and file exclusion files in the directory in which the standalone space estimator is run.

```
C:\Program Files\NetApp\snapvault\bin>svestimator -i c:\

Scanning system volumes...
Volume 'A:\' type Removable  Free Space 0%
Volume 'C:\' type Normal NTFS Free Space 90%
Volume 'D:\' type CDRom  Free Space 0%
Volume 'X:\' type Remote  Free Space 0%
Volume 'Y:\' type Remote  Free Space 46%
Volume 'Z:\' type Remote  Free Space 12%

Examining 'c:\'...

Estimated space requirements so far:
Installation: 12.00 MB
Database: 35.00 MB
Temp: 60.00 MB

Analyzing space requirements...
'C:\' is suitable for 'Installation requirements'
'C:\' is suitable for 'Database requirements'
'C:\' is suitable for 'Temporary space requirements'
Estimator has found sufficient space for backup
```

Figure 3) svestimator output (standalone)

Here is an example of running the free space estimator on a Windows 2000 system that doesn't have OSSV installed; for this we use the `-i` option. Notice that it shows we have 90% free space on the C:\ drive. Once it scans all the system volumes, it then determines how much space is required and if there is enough on the specified path. The last section of the output tells us that we do have enough space for the installation files, database requirement, and the temporary space requirements.

```
C:\Program Files\NetApp\snapvault\bin>svestimator -o c:\

Scanning system volumes...
Volume 'A:\' type Removable  Free Space 0%
Volume 'C:\' type Normal NTFS Free Space 90%
Volume 'D:\' type CDRom  Free Space 0%
Volume 'X:\' type Remote  Free Space 0%
Volume 'Y:\' type Remote  Free Space 46%
Volume 'Z:\' type Remote  Free Space 0%

Examining 'c:\'...

Estimated space requirements so far:
Database: 34.00 MB
Temp: 58.00 MB

Analyzing space requirements...
Estimator has found sufficient space for backup
```

Figure 4) svestimator output (built-in)

Now we will run it on a client that already has OSSV installed, using the -o option in the command.

Notice that it shows the current free space and the current requirements for the client database. Once it determines the space requirements, it verifies there is sufficient space.

Installing the OSSV 2.2 Agent on a Windows 2003 System

In this example we will be installing OSSV 2.2 on a Windows 2003 system called “i386-rtp01”.

Download the Windows 2003 agent from NOW.

Unzip the file in the folder of your choice.

Open an Explorer window in the appropriate folder (`ossv` folder will be unzipped) and simply double-click the `setup.exe` icon.

Figure 5) OSSV 2.2 Setup Wizard

Once the OSSV Setup Wizard appears, click Next and follow the directions on the subsequent screens.

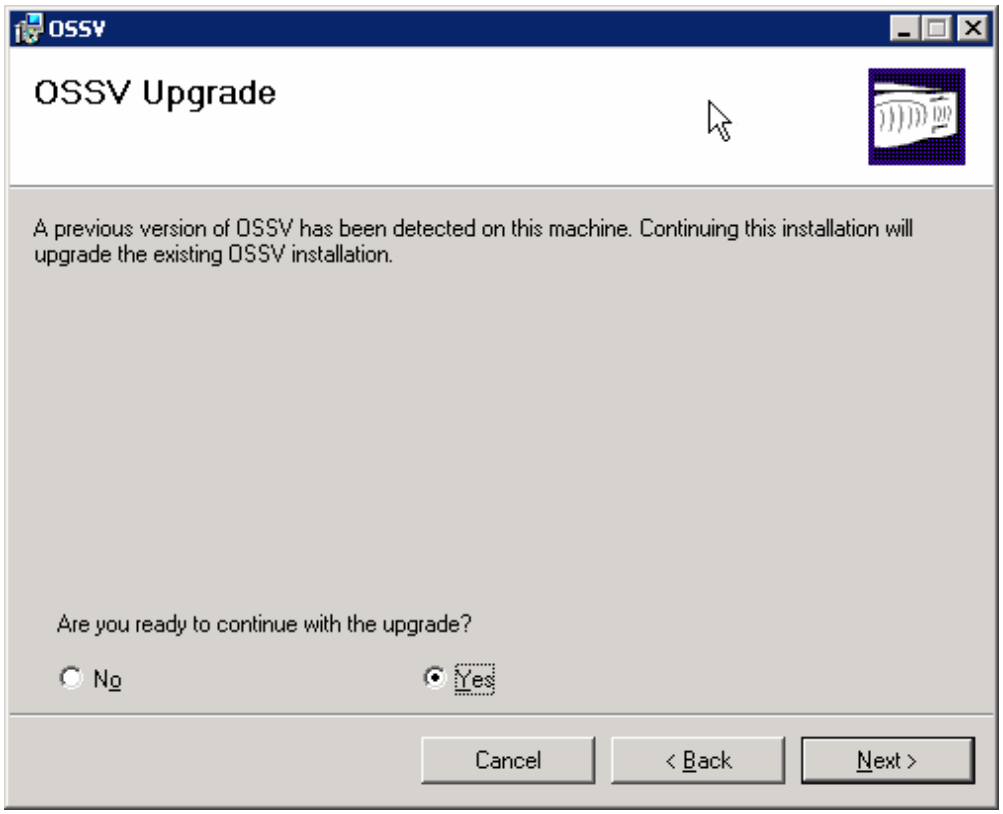

Figure 6) OSSV Upgrade Wizard

If the installation is an upgrade, you will encounter a screen similar to the above screen. In either case (upgrade or new), simply follow directions on the screens to walk through the installation. The installation should be fairly brief and is simple to achieve if all prerequisites have been met (see OSSV release notes on NOW).

NOTE: During the install, you must choose the NDMP listening port. By default this is TCP port 10000. If there is another application (e.g., VERITAS®) listening on this port on that particular primary system, the user may encounter problems once the automated `HealthCheck Utility` runs at the end of the installation routine. This utility attempts NDMP authentication as part of its sanity check; if NDMP is improperly configured or not listening, the installation routine will fail.

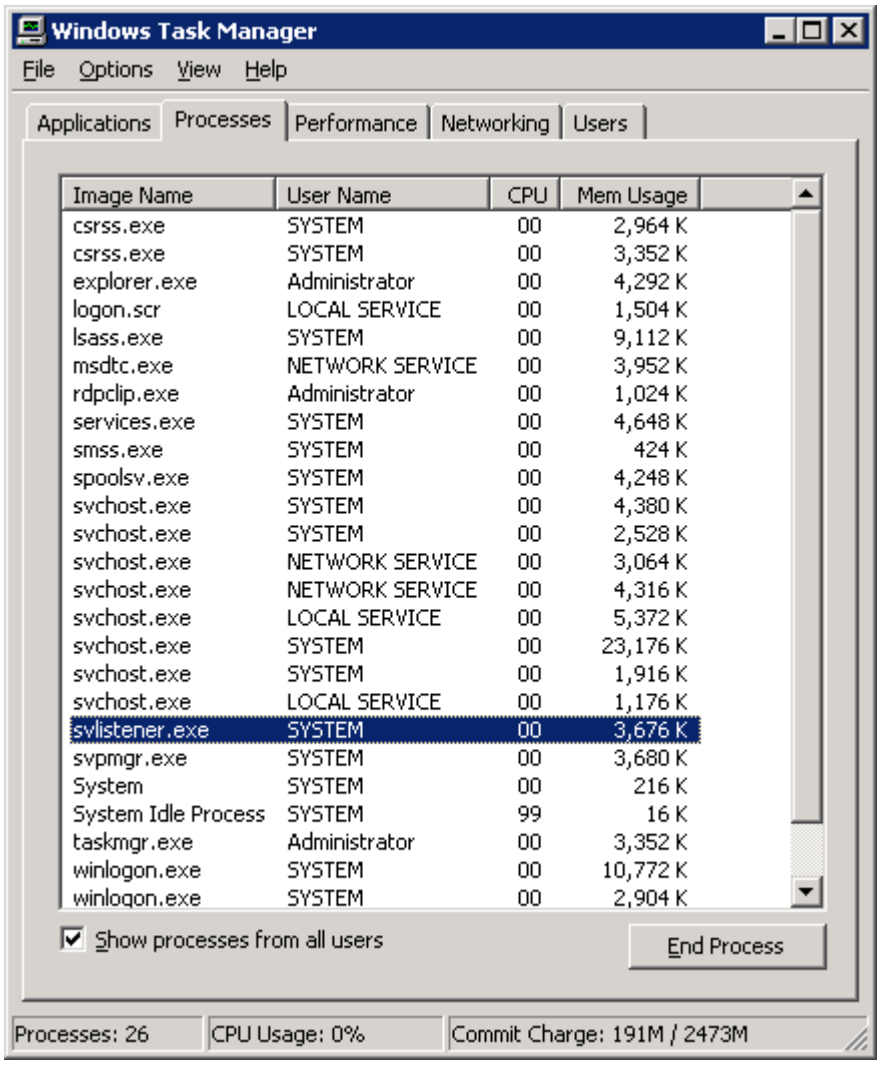

Figure 7) Task Manager - OSSV Processes

Once the installation is complete, you should see the `svlistener` process and the `svpmgr` process running on the primary system.

Installing the OSSV 2.2 Agent on a Solaris 9 System

Choose a system to be your OSSV primary. In this example we will be installing OSSV 2.2 on a Solaris 9 system .

Download the latest OSSV 2.2 agent (including the release notes) for Solaris from NOW.

Uncompress the file and extract it in a temporary directory (`uncompress *.tar.Z; tar -xvf *.tar`).

Once the file is extracted and uncompressed, you can simply use `pkgadd` to perform the installation. The extraction will create a directory called `ossv`. Perform the following on `ossv` and select the package for installation (only one option!), so enter 1 when asked to select the package:

```
# pkgadd -d ossv

The following packages are available:
  1  ossv     OSSV
              (sparc) core-id-7030

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
```

Next, answer questions related to the license agreement:

```
Processing package instance <ossv> from </ossv>

OSSV
(sparc) core-id-7030
NetApp

OSSV
2_2_2005DEC08

Have you read and agreed to the terms of the license?
(y = yes, n = no, d = display license) (y n d) [d] : y
```

Either accept the default path or add a new path where space is not an issue and most likely won't become an issue in the future. This is a good time to evaluate free space in /usr. If /usr is highly utilized, consider changing the default installation directory. If you wish to accept the default path, click Return; otherwise, type in the full path.

```
Please enter the path where you would like
the SnapVault directory to be created [/usr/snapvault] :
```

Next, enter NDMP information. Please *note* the information that is entered here. This information can be modified in the future from the command line and `svconfigurator (SnapVault tab)`. The NDMP listening port (default of 10000) can be modified during the install or using `svconfigurator` as well. This information is vital if using an NDMP management tool such as DataFabric Manager. By default the port is 10000. If you have *another backup application* (e.g., VERITAS NetBackup) installed on this system, you may need to use another port (e.g., 11000).

NOTE: If you use anything other than port 10000 for the NDMP listening port on your OSSV primary, you will need to change your NDMP management tool (DFM, for instance) to communicate on the new port as well.

NOTE: The postinstall health check utility will run by default after you complete your installation; if NDMP information is not present, the installation will fail.

```
Enter the User Name to connect to this machine
via the NDMP protocol : root

Please enter the password to connect to this machine
```

```
via the NDMP protocol :
Confirm password:

Enter the NDMP listen port [10000] :
```

Enter the name or IP address of the secondary system to which you are backing up. Enter multiple names/addresses here if multiple secondaries exist. This information can be modified using `svconfigurator` at a later time if necessary.

```
Enter the hostname or IP address of the SnapVault secondary
storage system(s) allowed to backup this machine.
Multiple hostnames or IP addresses must be comma separated.
> : r100-rtp01
```

After you enter the secondary system information, the following will appear, and you will be prompted to continue with the installation:

```
## Executing checkinstall script.
checkinstall running
CHOSEN_CLASSES=ossvcore
PKG_BASE=/usr/snapvault
Using </var/tmp> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <ossv> [y,n,?] y
```

The install script will now begin, and the software will be installed on the system. Near the end of the installation, services will be started on the system. Once the services have started, a health check will run. If it is successful, the following information will appear after the output from the health check utility:

```
Check Succeeded

Installation appears valid

Installation of <ossv> was successful.
```

To verify a successful install (outside of using the health check utility), ensure that `svlistener` is running:

```
# ps -ef | grep svlist
    root   754   751  0 10:39:09 ?        0:00 svlistener 50
```

NOTE: For the remainder of this document we commonly refer to a Windows environment (`svconfigurator`, etc.). All items referred to in the following sections apply to UNIX as well, with only slight variations.

SVCONFIGURATOR

Figure 8) `svconfigurator`

Once you have properly installed the OSSV agent on your primary system, complete basic configuration by modifying any outstanding parameters using the svconfigurator tool (accessible from the Start menu or from the command line, or `$INSTALL_DIR\bin\svconfigurator.exe` (`$INSTALL_DIR/bin/svconfigurator` in UNIX).

Once the `svconfigurator` GUI appears, browse the various tabs to view information about your particular installation.

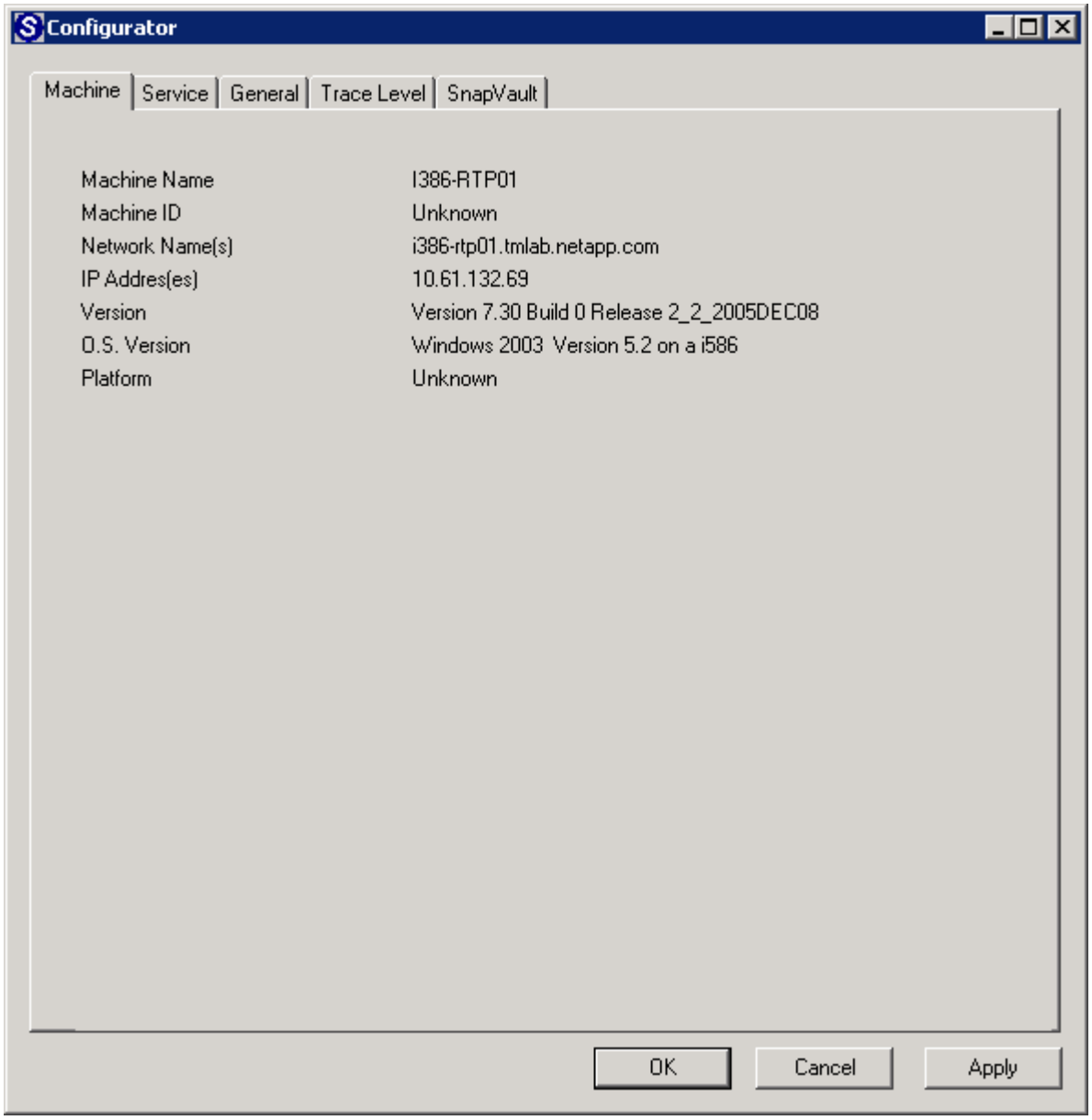

Figure 9) `svconfigurator` - Machine Tab

The **`Machine`** tab displays information about the version of OSSV and the primary system machine information (OS, hardware, etc.).

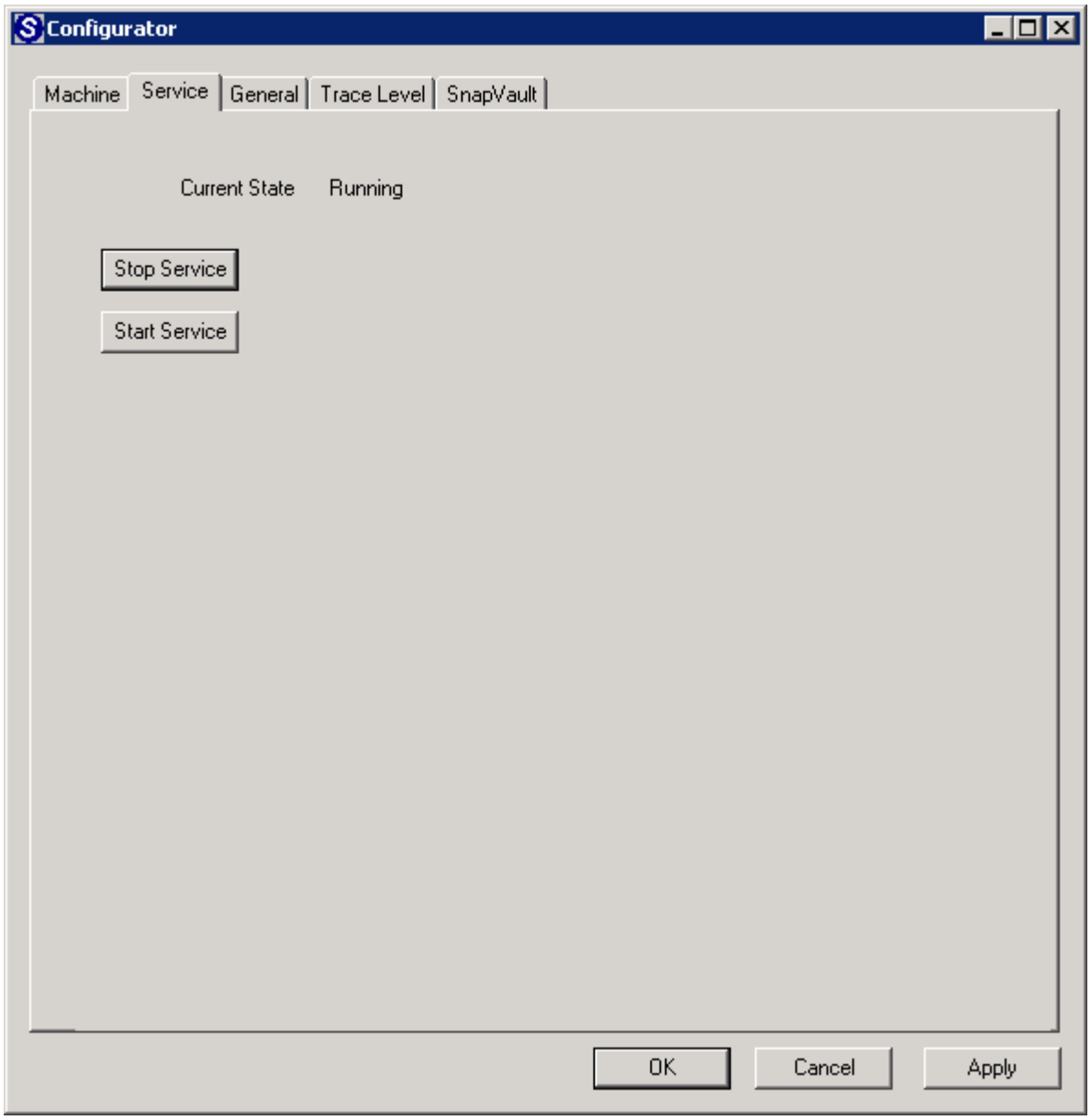

Figure 10) `svconfigurator` - Service Tab

The **`Service`** tab allows the user to stop and start the OSSV services.

NOTE: Use this tab when stopping and starting the OSSV service. This will ensure that all pertinent services for OSSV are stopped and then restarted. It is NOT recommended that you stop and start OSSV from Windows Services.

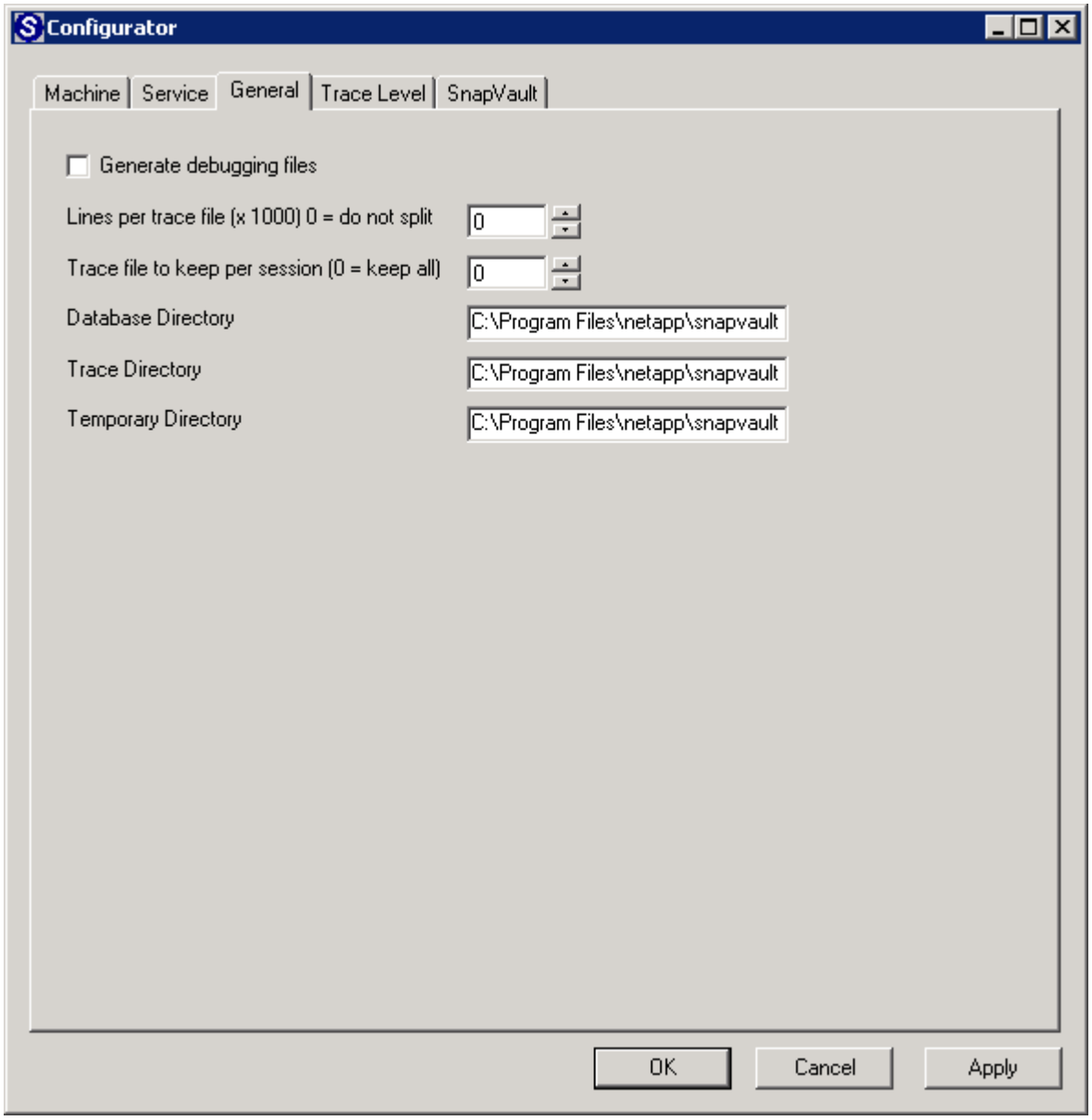

Figure 11) `svconfigurator` - General Tab

The **`General`** tab allows for generation of debug files (first click the button here, then move to the **`Trace Level`** tab to modify default process log output settings). This tab also allows default directory locations to be modified. This might be necessary in situations where these default file systems are approaching maximum capacity. Be very clear on the amount of free space in the `$INSTALL_DIR\db` directory. **NOTE**: If you generate debug logs, especially in `Verbose` mode, you must set the level back to `Normal` once all relevant data is gathered. Otherwise, the `$INSTALL_DIR\trace` directory will rapidly reach its limit.

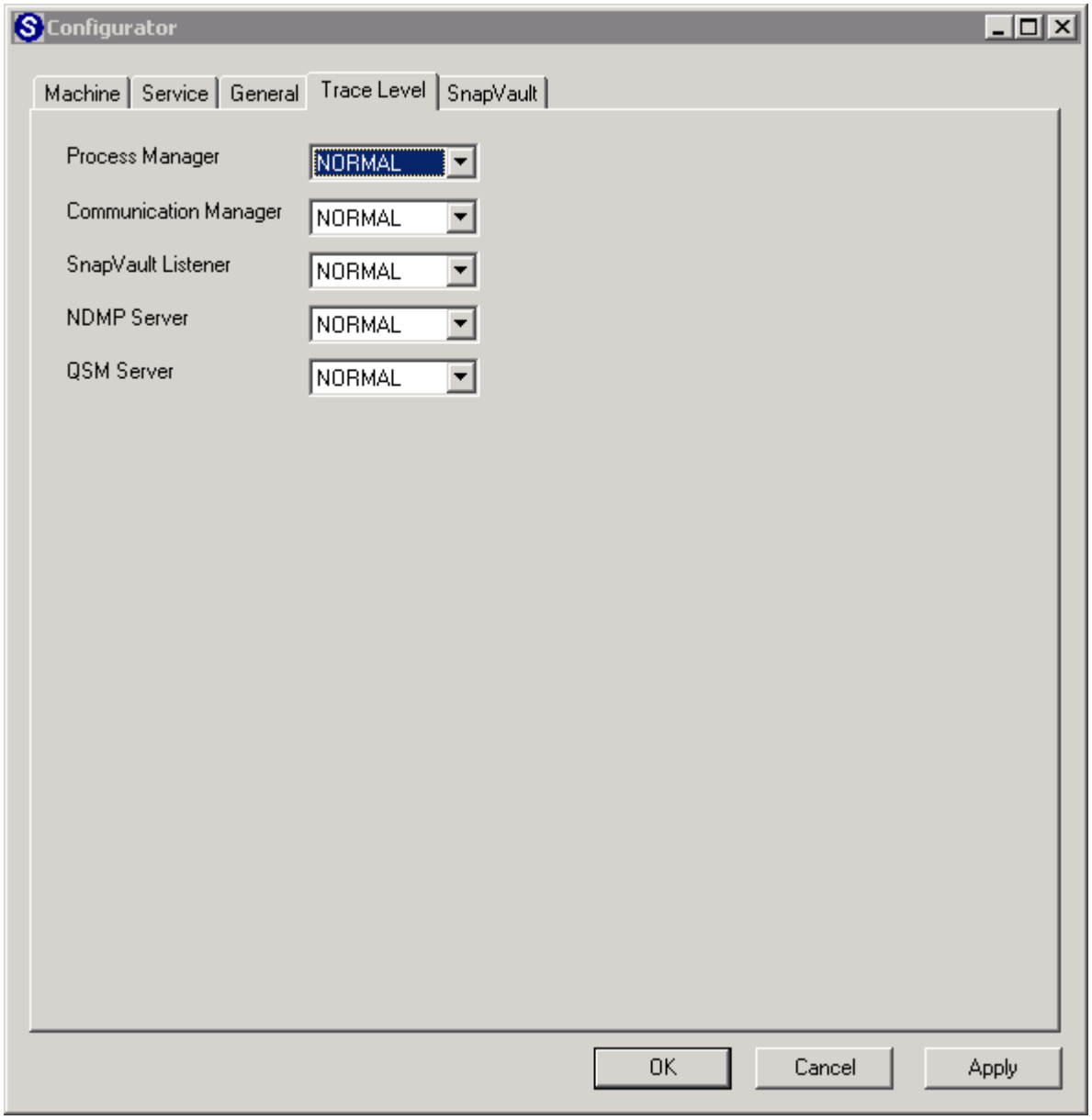

Figure 12) `svconfigurator` - Trace Level Tab

The **`Trace Level`** tab is available to allow modification of default logging output for the OSSV processes. These settings will be modified when `Generate debugging files` is selected from the **`General`** tab.

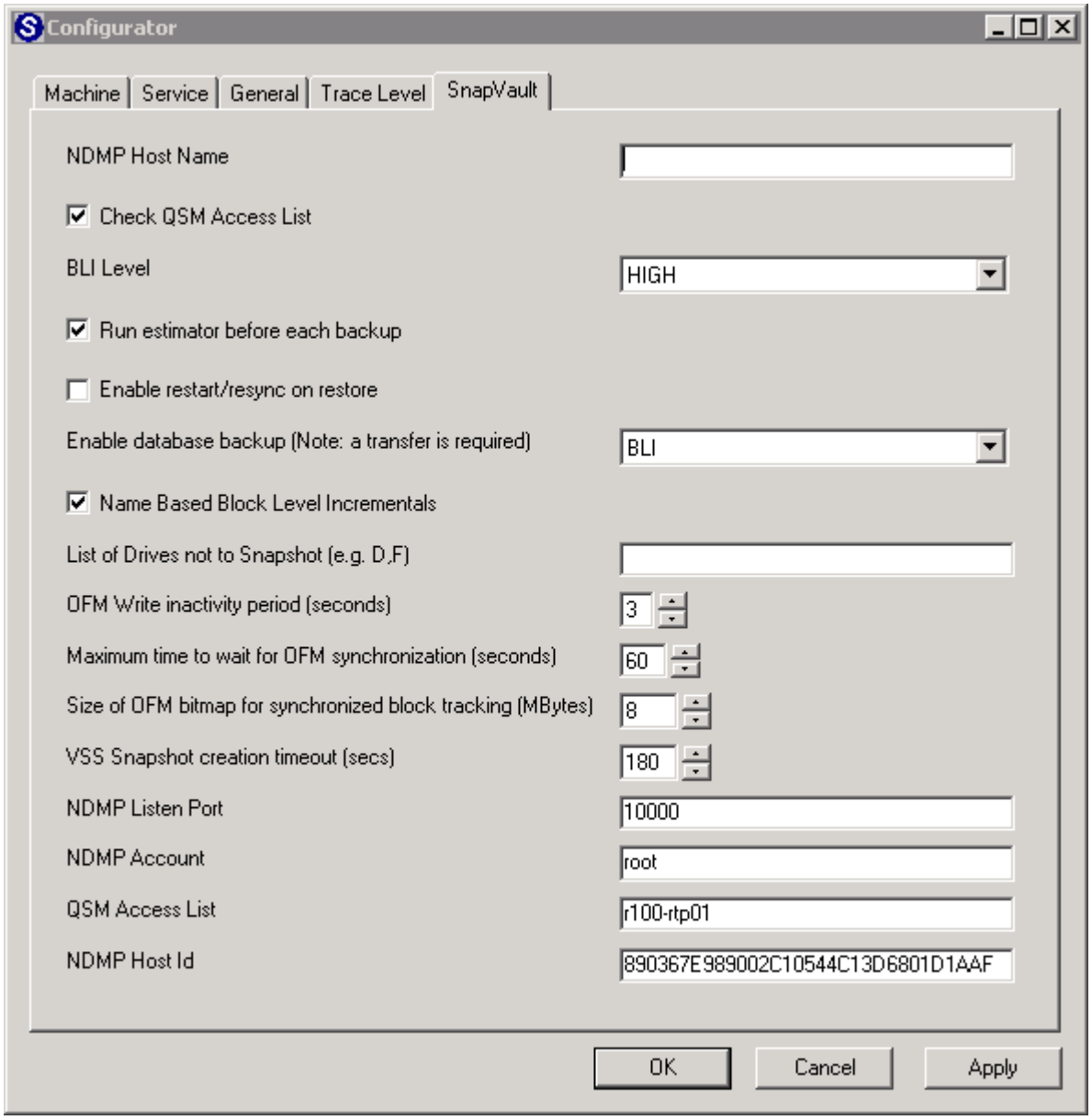

Figure 13) `svconfigurator` - SnapVault Tab

The **`SnapVault`** tab is available to allow multiple OSSV configurations to be modified. From this location, you can modify BLI level, OFM default parameters, NDMP parameters, VSS parameters, and security settings (`Check QSM Access List`). Once you select `Check QSM Access List,` you must specify systems (name or IP) to which you are allowing this particular primary system to back up.

NOTE: Any time a parameter is modified, click Apply to apply settings. It is also often necessary to stop/start the OSSV service after making a change using the “Service” tab. For more information on these settings and their exact meanings, *refer to the latest OSSV release notes, located on the NOW site.*

Binaries

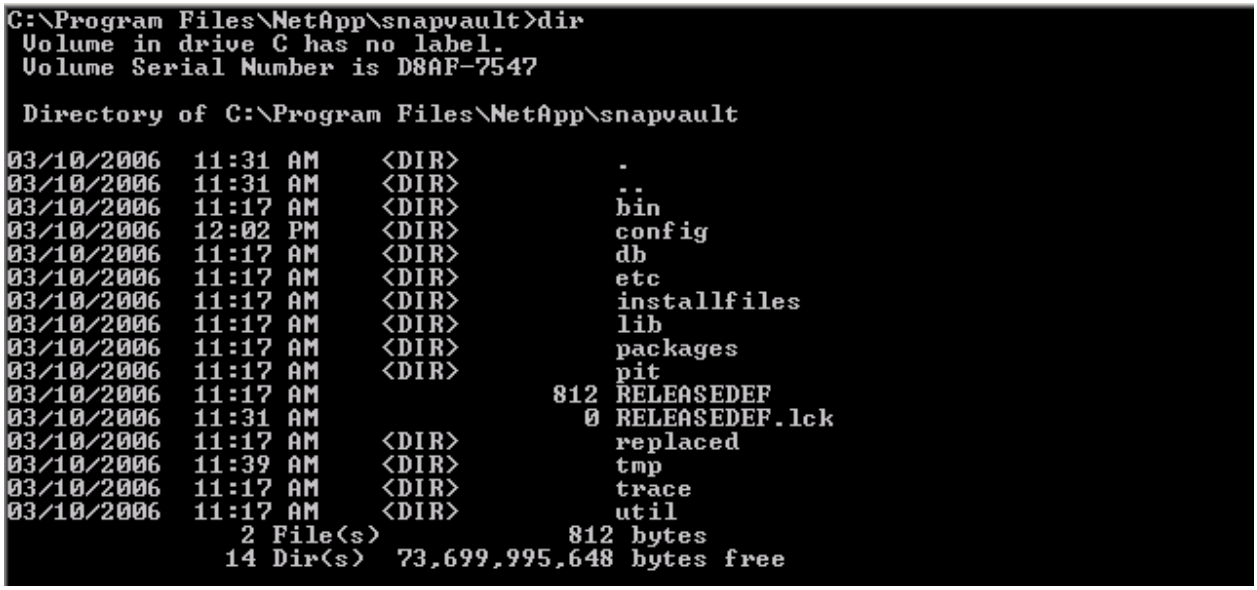

Figure 14) OSSV Binaries

A set of binaries is installed with the primary OSSV agent. These binaries are available to perform various tasks to maintain a successful OSSV environment. Tasks include modifications of core settings, NDMP password changes, checking health of installation, OSSV database backup, stopping and starting services, updating the parameters typically modified in `svconfigurator` at the command line, etc. Most of these binaries can also be executed using `svconfigurator`. Some environments do not allow for `x` sessions or GUI management (typical UNIX shops), and command line would be necessary. Becoming familiar with these commands is often necessary.

```
C:\Program Files\NetApp\snapvault\bin>svinstallcheck
SnapVault home directory: 'C:/Program Files/netapp/snapvault'
SnapVault database directory: 'C:\Program Files\netapp\snapvault\db'
SnapVault temporary directory: 'C:\Program Files\netapp\snapvault/tmp'
SnapVault Database and Temporary directories have 92% space left (70285Mb)
SnapVault service is running
SnapVault listener is running
NDMP Server, on port 10000, details:
    Vendor      Netapp
    Product     SnapVault
    Version     02.02
    Host        i386-rtp01
    Host Id     890367E989002C10544C13D6801D1AAF
    OS Type     Windows 2003
    OS Version 5.2
SnapVault QSM Server is responding correctly
Validating filesystems:
   Drive 'A:\' is removable, unsuitable for backup
   Drive 'C:\' is suitable for backup
   Drive 'D:\' is a CDROM, unsuitable for backup
   Drive 'X:\' is removable, unsuitable for backup
   Drive 'Y:\' is removable, unsuitable for backup
   Drive 'Z:\' is removable, unsuitable for backup

Check Succeeded
```

Figure 15) svinstallcheck

An important executable is `svinstallcheck`, otherwise known as the `HealthCheck Utility`. This tool will perform a quick sanity check of system details, file systems suitable for backup, NDMP authentication, and database and temporary space available. This file runs automatically at the end of a new installation or upgrade. It is recommended that this utility be run manually or via script on a regular basis to track the `SnapVault Database and Temporary directories` remaining space.

```
C:\Program Files\NetApp\snapvault\bin>snapvault
The following commands are available; for more information
type "snapvault help <command>"
abort            destinations     help            release
restore          status
```

Figure 16) snapvault command

In order to monitor `status` of an OSSV backup or restore, use the `snapvault` command from the primary system. This command also allows the user to `release` relationships that have been stopped on the secondary systems (freeing up the primary directory for backup to a different location). This command also provides the user with the important function of `restore`. Please follow syntax rules displayed using `snapvault <command> help` on the primary.

```
C:\Program Files\NetApp\snapvault\bin>svpassword
Password:
Password changed
```

Figure 17) svpassword

In order to modify the NDMP password, use the `svpassword` executable.
NOTE: There is no confirmation of password, so type slowly.

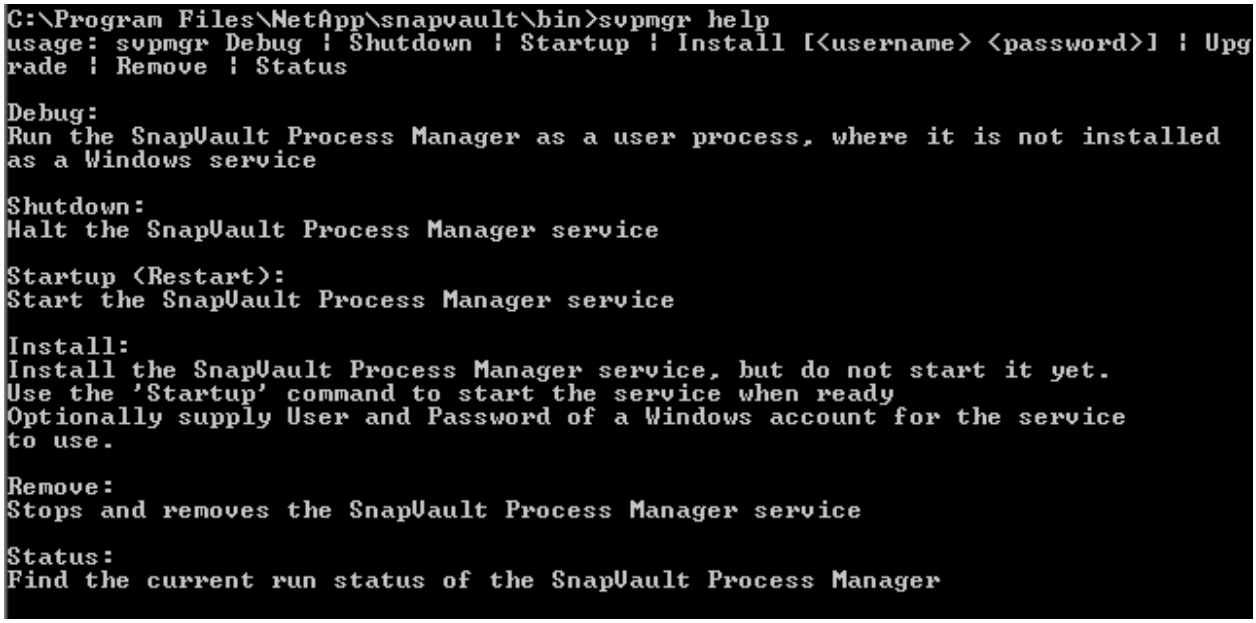

Figure 18) svpmgr utility

To quickly stop and start OSSV services, use the `svpmgr` utility. This command will shut down OSSV-related services and start them up if necessary.

ETC and TRACE Directories

The $INSTALL_DIR\etc and $INSTALL_DIR\trace directories are also important locations for various files and logs.

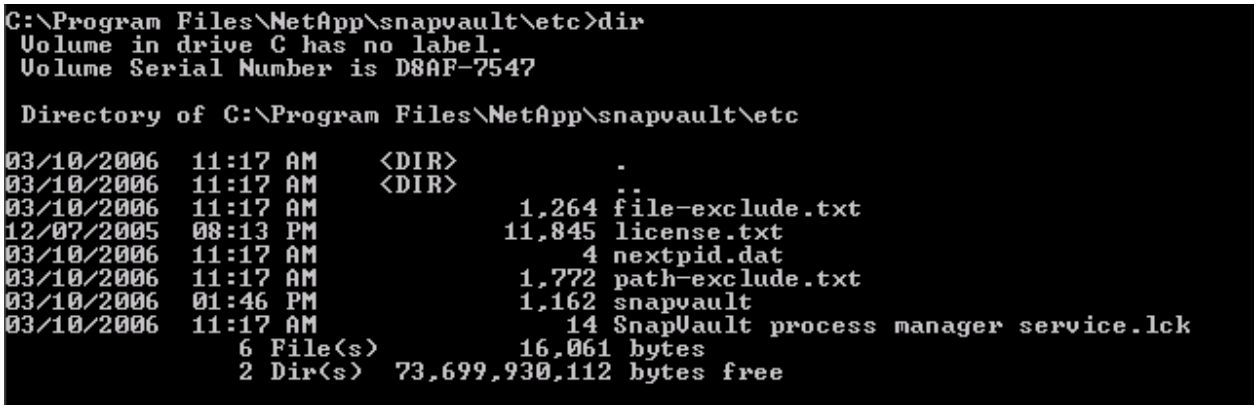

Figure 19) etc directory contents

To find the log files associated with a primary, look in the `etc` directory for the `snapvault` log file. This file contains all backup information, restore information, and error information for the primary system; `VSS`, `OFM` information is noted, as well as time stamps, file systems being backed up, etc. The log file is

automatically archived in this location also and are created during the first transfer of the following day. Log files for previous days are named snapvault.yyyymmdd.

In addition to the log file, you will find the exclude list files (`path-exclude.txt` and `file-exclude.txt`) in this directory. As mentioned in previous sections, these files allow for exclusion of specific files, ranges of files, and full directories.

```
C:\Program Files\NetApp\snapvault>cd trace

C:\Program Files\NetApp\snapvault\trace>dir
 Volume in drive C has no label.
 Volume Serial Number is D8AF-7547

 Directory of C:\Program Files\NetApp\snapvault\trace

03/10/2006  02:34 PM    <DIR>          .
03/10/2006  02:34 PM    <DIR>          ..
03/10/2006  02:34 PM               871 qsmserver.log
03/10/2006  02:32 PM           110,128 qsmserver107.log
03/10/2006  02:33 PM        34,620,187 qsmserver108.log
03/10/2006  02:34 PM        34,620,187 qsmserver109.log
03/10/2006  02:34 PM             8,465 svconfigurator.log
               5 File(s)     69,359,838 bytes
               2 Dir(s)  73,303,617,536 bytes free
```

Figure 20) trace directory

The trace directory includes log files that were generated during debug and troubleshooting situations. Do not enable debugging without the assistance of support. If debugging is enabled, please set the debug level back to its default of `Normal` when finished with the troubleshooting. Multiple files can exist in this location: multiple files for one process or multiple files for many processes. These files can grow rapidly, and space consumption should be monitored.

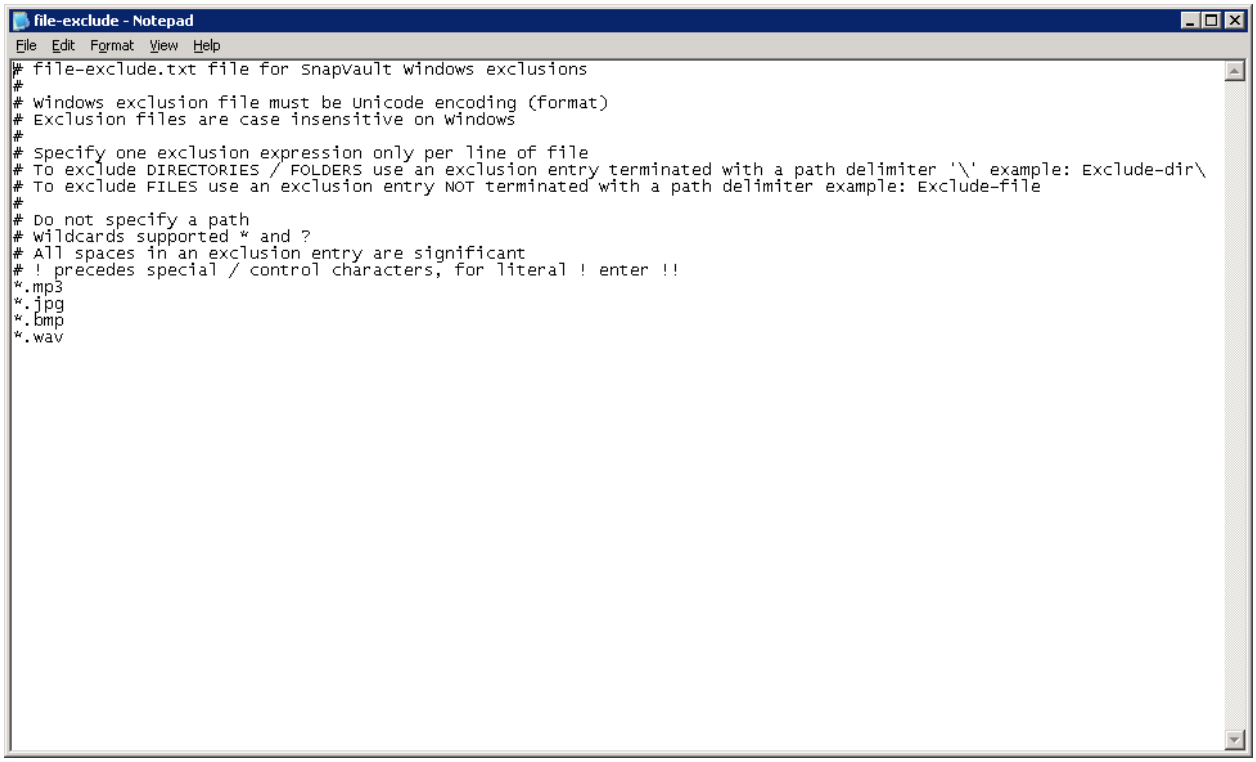

Figure 21) `file-exclude.txt`

The exclude files contain various comments on syntax, wildcards, and allowed characters. In addition to the `file-exclude.txt` file displayed above, there is a similar file, `path-exclude.txt`, providing the user with the ability to exclude full paths and directories. Once an update is made to one or both of these files, the exclusion will occur on the next update. Please pay close attention to these parameters and refer to *the latest OSSV release notes, available on NOW,* for specific information.

Creating an Unattended Install Image

In order the help deploy OSSV over a large number of clients, OSSV 2.2 introduced Unattended Install. Below we will go over how to create an unattended install on Windows 2003 system.

NOTE: Once we create an Unattended Install image on a Windows 2003 system, we can then only deploy that to other Windows 2003 systems. If the environment also contains Windows 2000 systems, we will need to create a separate Windows 2000 Unattended Install image.

In order to create an Unattended Install image, you must go to a system that has OSSV installed and stop OSSV services via the `svconfigurator` utility.

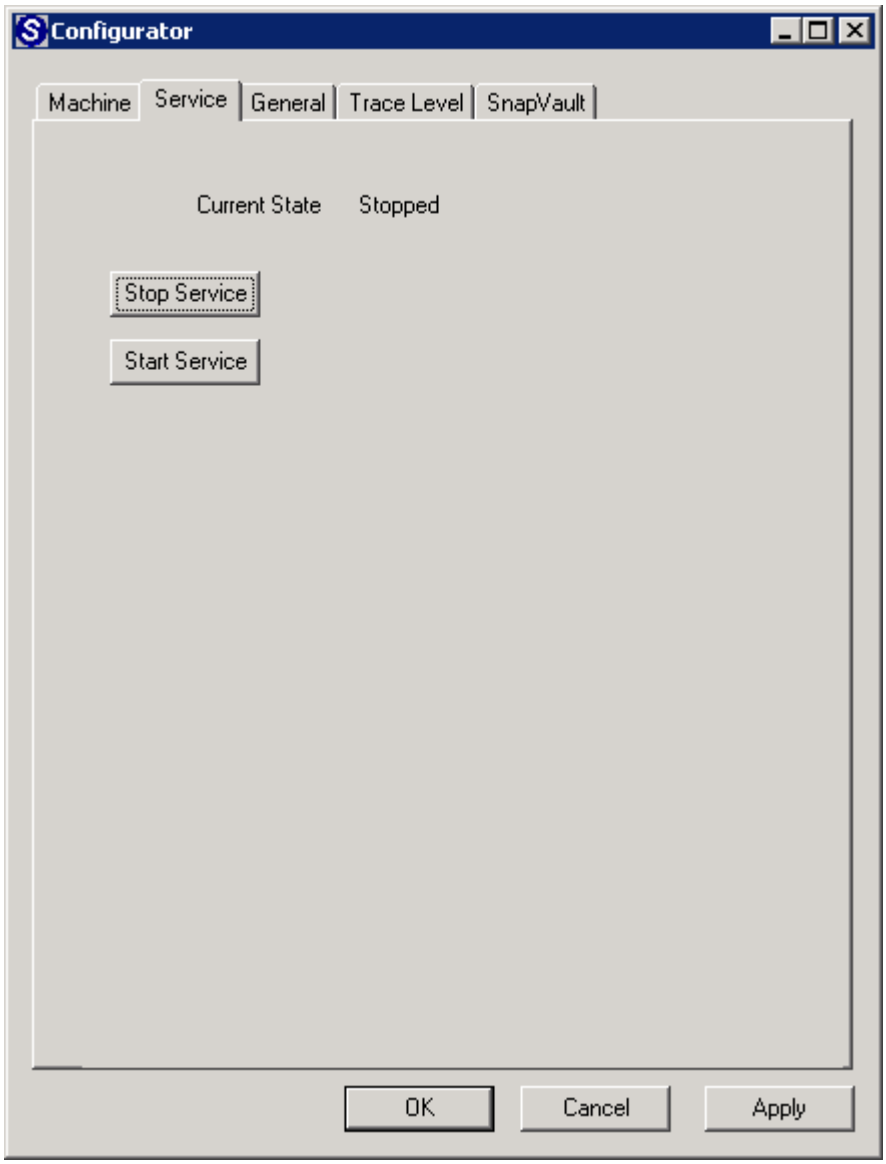

Figure 22) `svconfigurator` - Stopped Services

Once the services have been stopped, configure all the parameters you want your new install to have using the `svconfigurator` utility. After setting your parameters, close the configurator utility.

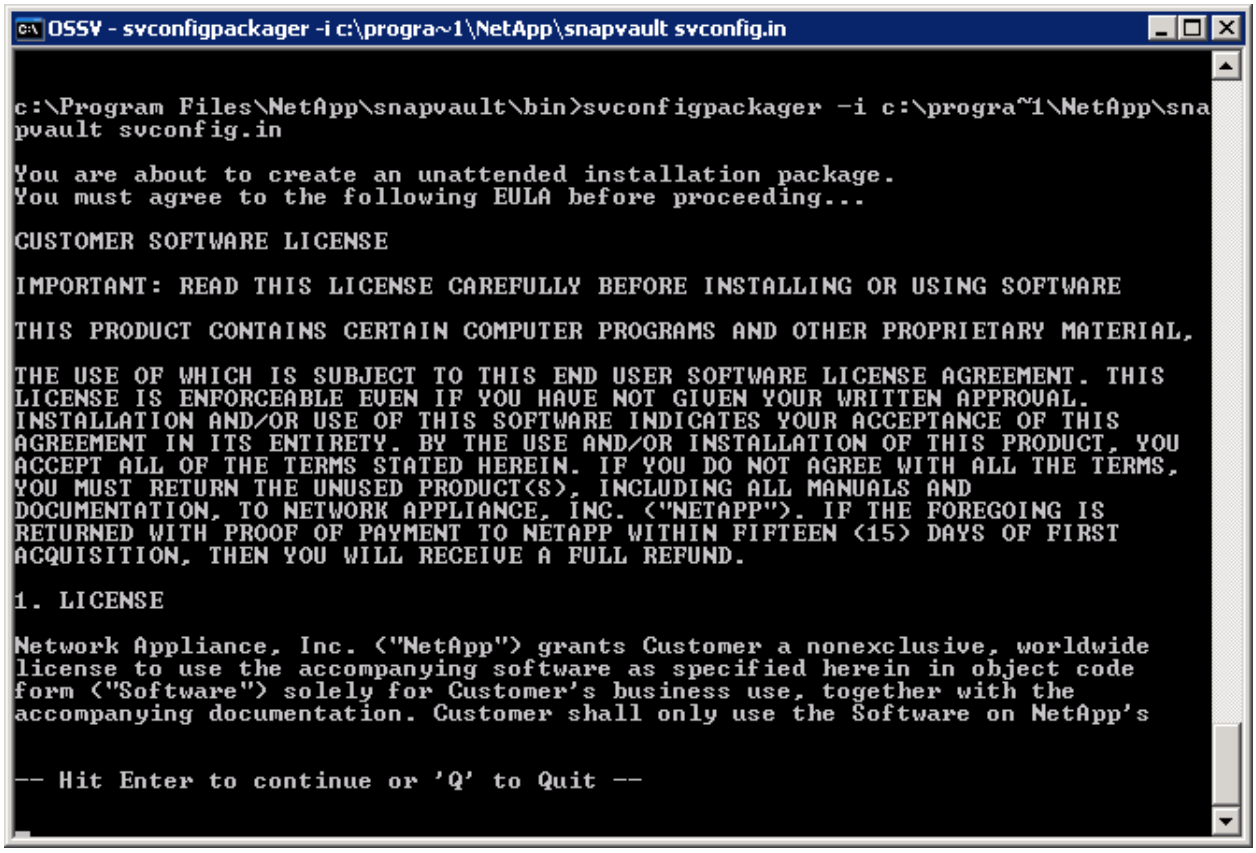

Figure 23) `svconfigpackager`

Open a CLI window and change to the `$INSTALL_DIR/bin` directory. To create an installation script and save the configuration settings to a file, use:

```
svconfigpackager –i path_name filename
```

Where `path_name` is the location of the destination for the rollout.

NOTE: If this is an upgrade, be sure to add the –h option, which tells Unattended Install to honor the existing configuration parameters.

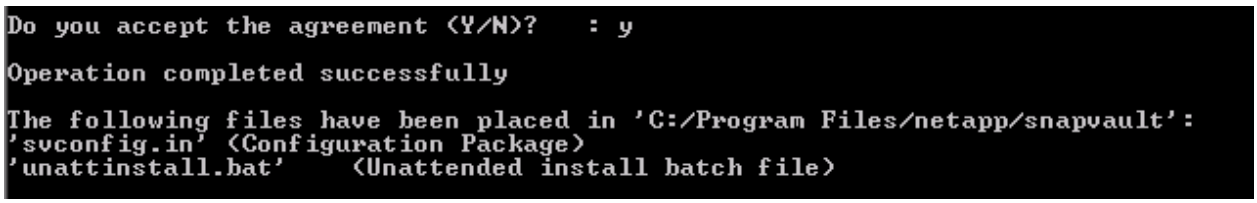

Figure 24) `svconfigpackager` - completed

Once you accept the terms to the license agreement, you will see 2 files that were created, `svconfig.in` and `unattinstall.bat`. These are the files that will need to be placed on each server, along with the OSSV install files.

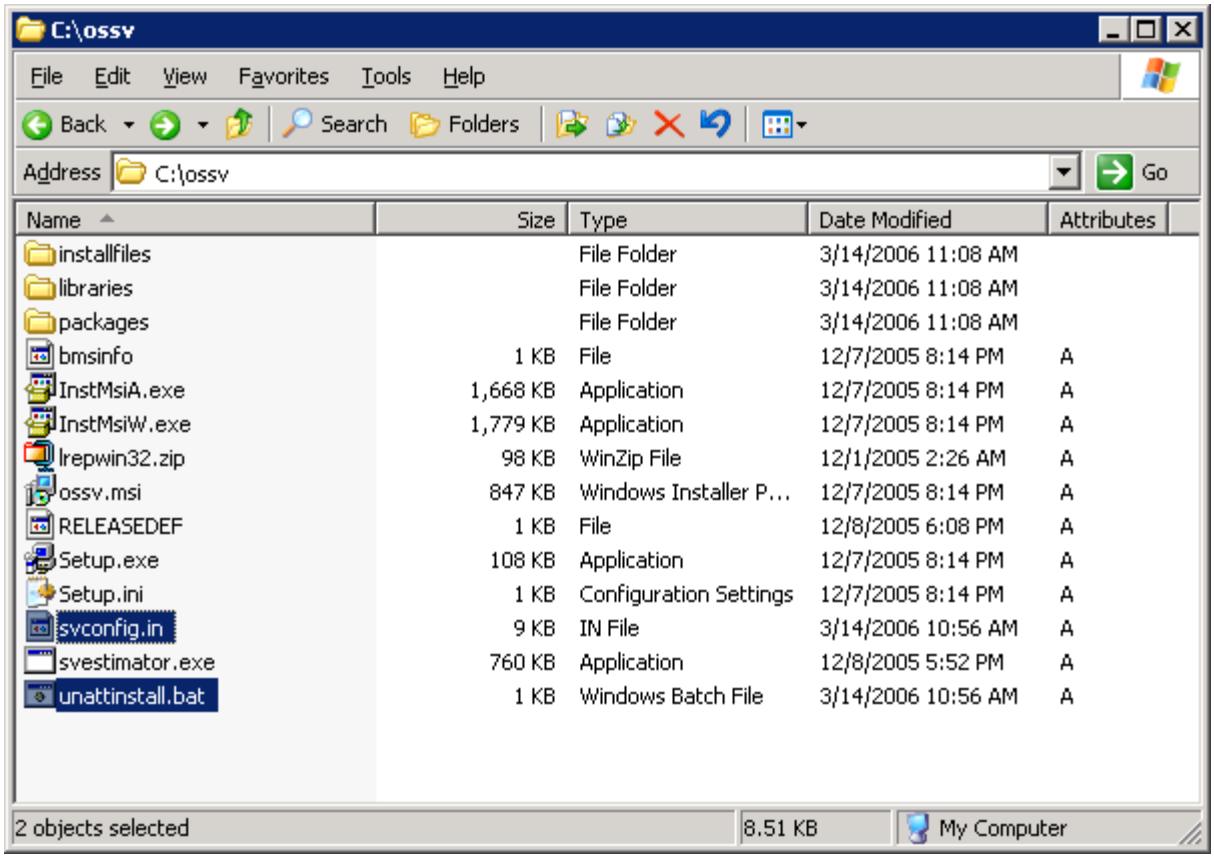

Figure 25) `svconfigpackager` files

Here we have unzipped the OSSV package and placed the `svconfig.in` and `unattinstall.bat` files in `c:\ossv` on the server we wish to perform the unattended install on.

```
C:\ossv>unattinstall.bat

C:\ossv>msiexec /i ossv.msi /qn targetdir="c:\progra~1\NetApp\snapvault" db_dir=
"C:\Program Files\netapp\snapvault\db" trace_dir="C:\Program Files\netapp\snapva
ult/trace" tmp_dir="C:\Program Files\netapp\snapvault/tmp" reboot=ReallySuppress
 UNATTENDED_INSTALL=1 HONOR_EXISTING_CONFIG=0 CONFIG_FILE=svconfig.in

C:\ossv>
```

Figure 26) `unattinstall.bat`

Once all the files have been copied, open a command window and issue the `unattinstall.bat` command. This will read the configuration file (`svconfig.in`) and install OSSV on the server without any user interaction. At the end of the installation, the script will automatically run `svinstallcheck` to

verify that the installation was a success. If the installation fails, log files are generated and are logged in `%SystemRoot%\Documents and Settings\Current User\Local Settings\Temp`.

Configuring the Secondary System

Once the secondary system is upgraded to the appropriate release of Data ONTAP (refer to the latest OSSV release notes, available on NOW), various settings will need to be configured.

```
      sv_linux_pri ABCEFGH for 500 nodes
      sv_ontap_pri not licensed
      sv_ontap_sec site
       sv_unix_pri ABCEFGH for 500 nodes
sv_windows_ofm_pri not licensed
    sv_windows_pri ABCEFGH for 500 nodes
```

Figure 27) Secondary licenses

Licensing should be in place; appropriate licenses may need to be purchased, depending on the environment. The secondary system itself will need to be licensed as the SnapVault secondary. These licenses are configured using the `license add` command.

```
r100-rtp01> options snapvault
snapvault.access              all
snapvault.enable              on
```

Figure 28) `options snapvault`

Once licensing is in place, the `snapvault.access` and `snapvault.enable` options will need to be configured. The `snapvault.access` option refers to machines (via name or IP; ranges are legal) that are allowed to back up to this particular secondary system. See the `man` pages for details on wildcards and ranges for this command.

```
r100-rtp01> snapvault
The following commands are available; for more information
type "snapvault help <command>"
abort               modify              start               stop
destinations        release             status              update
help                snap
```

Figure 29) `snapvault` - Secondary

The `snapvault` command is available to enable SnapVault baselines (relationship creation, level-0, achieved via `snapvault start`), allow users to monitor `status`, delete relationships and subsequently qtrees (`snapvault stop`), manually kick off incremental backups outside regularly scheduled backups (`update`), view `destinations`, `modify` relationships, create SnapVault schedules (`snap`), and abort transfers. See the `man` pages for more detailed information about each command, including syntax.

Creating a Baseline Relationship

Once all the previous sections have been successfully completed and reviewed, please review the OSSV release notes again to ensure no settings or recommendations were missed. We can now create a relationship and initiate the baseline transfer, which will allow the user to begin scheduled block-level or file-level updates or incrementals moving forward.

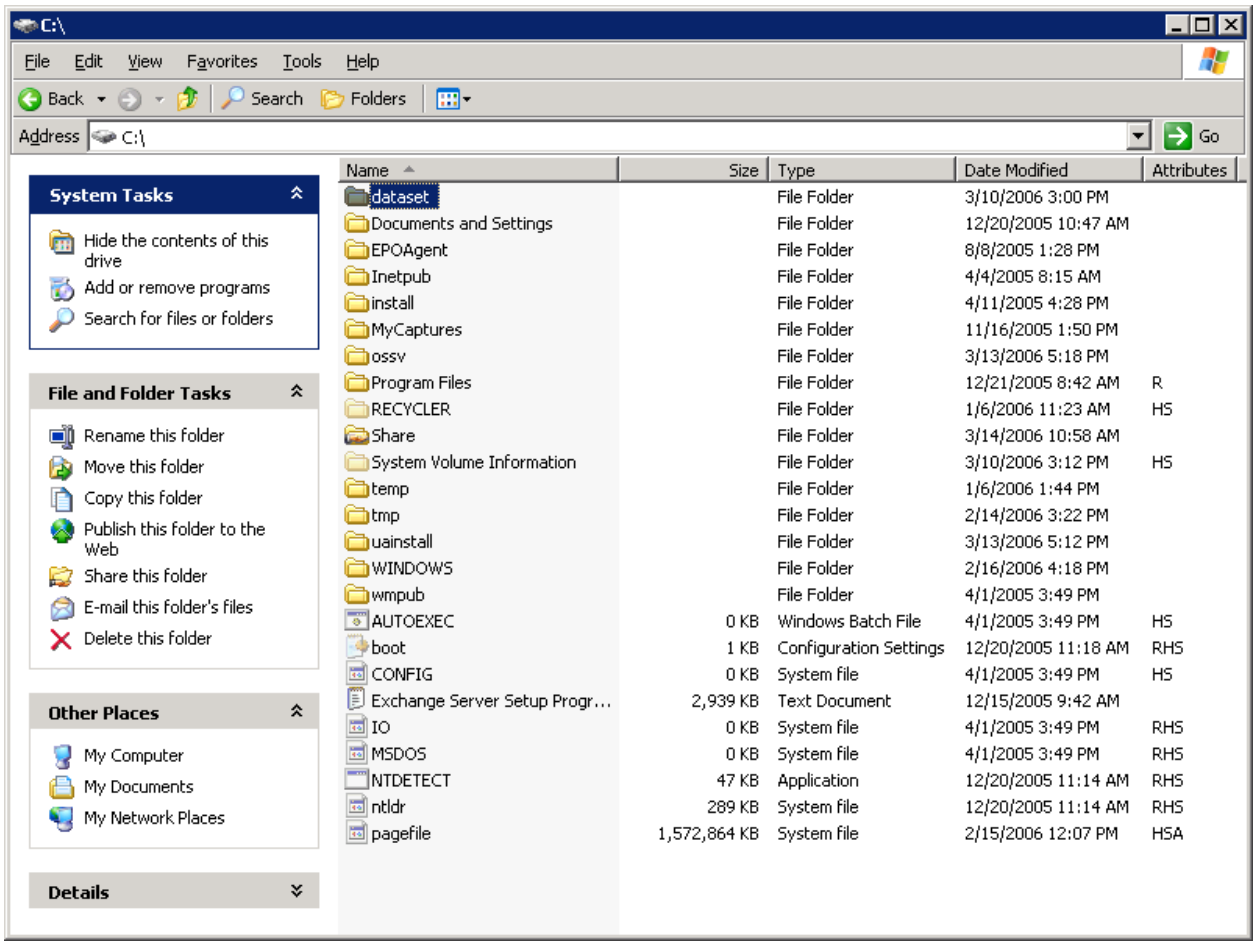

Figure 30) dataset to be backed up

First, select the data set to back up. Once the proper data set has been chosen, create a meaningful name for the secondary qtree. In this example, we are backing up `C:\dataset`, which happens to be a 1.5GB file system.

```
r100-rtp01> snapvault start -S 10.61.132.69:c:\dataset /vol/ossv_flex/dataset
Snapvault configuration for the qtree has been set.
Transfer started.
Monitor progress with 'snapvault status' or the snapmirror log.
r100-rtp01>
```

Figure 31) Running `snapvault start` from Secondary

Now log in to the secondary system and issue the baseline creation command, which is `snapvault start` (see man pages for detailed syntax and options). As you can see above, our relationship is backing up `C:\dataset` from host 10.61.132.69 to the `/vol/ossv_flex/C_dataset` qtree located on r100-rtp01.

```
r100-rtp01> snapvault status -l /vol/ossv_flex/dataset
Snapvault secondary is ON.

Source:                  10.61.132.69:c:\dataset
Destination:             r100-rtp01:/vol/ossv_flex/dataset
Status:                  Transferring
Progress:                488456 KB
State:                   Uninitialized
Lag:                     -
Mirror Timestamp:        -
Base Snapshot:           -
Current Transfer Type:   Initialize
Current Transfer Error:  -
Contents:                Transitioning
Last Transfer Type:      -
Last Transfer Size:      -
Last Transfer Duration:  -
Last Transfer From:      -
r100-rtp01>
```

Figure 32) `snapvault status -l`

Monitoring the transfer with the `snapvault status –l` command (long listing) is recommended. From this output, you can view bytes transferred, time, type of transfer (baseline, update), age of backup (lag), etc. Key items to take away from this output are `Progress, Current Transfer Type, Status,` and `Lag time`.

`Progress` is the amount of data (`kB`) that has been transferred at the time the `snapvault status –l` was issued. This output will generally be higher than the actual amount of data residing on the primary system due to overhead associated with SnapVault (approximately 16kB for every file updated).

`Current Transfer Type` displays whether this is a baseline, (`Initialize`) transfer, incremental (`Replica`) transfer, or `Restore` transfer. As always, refer to the Data ONTAP `man` pages for more detail.

`Lag` is the amount of time that has expired since the last successful backup. Basically, a session with a `Lag = 23` hours means that the session displayed is 23 hours old, or 23 hours behind the current time. This is completely normal if nightly backups are being performed; however, if hourly backups are being performed, this reveals that the backups are not occurring every hour and haven't occurred for the last 23 hours.

```
r100-rtp01> snap list ossv_flex
Volume ossv_flex
working...

  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  Mar 10 15:04  r100-rtp01(0033604316)_ossv_flex-base.0 (busy,snapvault)
r100-rtp01>
```

Figure 33) `snap list`

Once the transfer is complete, the secondary will create a baseline Snapshot copy, which is a *softlocked* Snapshot copy. All subsequent updates will be named according to the `snapvault snap sched` input. Snapshot copies will accumulate until the retention number specified in the `snapvault snap sched` is reached.

Scheduling OSSV Backups via the Secondary System

A good rule of thumb is to specify all primaries with the same class of data for protection, to be directed to the same volume on the secondary server. So, for example, data on primaries that need to be backed up every night might go to `/vol/backup_nightly`. Data on primaries that need to be backed up twice a day might go to `/vol/backup_twice_daily`.

The `snapvault snap sched` command is used for scheduling with Data ONTAP (you can choose to schedule using a supported NDMP tool as well). `snapvault snap sched` is available on the secondary system. This command sets, changes, or lists Snapshot schedules. If no schedule argument is given, the command lists currently configured Snapshot schedules.

There are two options for scheduling: `-x` is the transfer schedule; `-c` is the create schedule and is the default. The `-x` option tells OSSV to *transfer* new data from all primary qtrees residing in that particular volume prior to creating the Snapshot copy. If `-x` is not specified, only local Snapshot copies on the secondary will be created without any communication with the primary. To move changed data from the primary on a schedule (incrementals), the `-x` option needs to be added to the `snapvault snap sched` command input.

```
R100> snapvault snap sched -x backup sv_daily 5@Mon-Fri@23
```

Where the schedule itself is:

```
cnt[@day_list][@hour_list]
OR cnt[@hour_list][@day_list]
```

The command above schedules incremental backups every weekday at 2300 hours on the volume named `backup`. When Snapshot copies are created and begin to accumulate (toward the specified `cnt` or retention policy) on the secondary for a particular volume, they are numbered from oldest to newest, from 0 to `cnt`-1. When creating a new Snapshot copy, the SnapVault process on the secondary system will delete the oldest Snapshot copies, increment by one the number on the remaining Snapshot copies, and finally create a new “0” Snapshot copy.

Further information about this command may be obtained by consulting the *Online Backup and Recovery Guide,* SnapVault section. `man` pages within Data ONTAP can also be referenced.

Recovering OSSV Data Using the Command Line

If it is possible to NFS mount or CIFS map the secondary volume on the OSSV client, recoveries may be performed by simply copying SnapVault data from the secondary volume. Permissions on SnapVault data are the same as permissions on user data, so the same authentication rules apply.

If it is not possible to access to mount or map the secondary volume, you may use `snapvault restore`, which is part of the OSSV distribution.

Restoring Data on an OSSV Primary Running Windows 2000

All command-line actions to restore data to a primary are initiated from the primary. Log on to the primary to begin restoring data.

```
C:\>cd Program Files\netapp\snapvault\bin

C:\Program Files\netapp\snapvault\bin>snapvault restore –s sv_daily –S r100-
rtp01:/vol/ossv/my_documents C:\Temp\restored_my_documents
```

The text following “-s” is the Snapshot copy name. The text following “-S” is the name of the secondary server, followed by the location of the backed-up data on the secondary volume. The final argument is the location on the primary where the file is to be restored. Note that the file may be restored to a different location on the primary under a different name. The command line is case sensitive. When files are located multiple subdirectories deep on a file system, please be careful with this command, since it is sensitive. Single file restores can be issued using `snapvault restore`.

NOTE: The root primary file system name that was being backed up is essentially replaced with the qtree name when performing a restore using this command.

Uninstalling the OSSV Primary Agent

In some cases, it will be necessary to uninstall the OSSV client agent due to NDMP port unavailability, corrupted registry settings, etc. Before uninstalling the client agent, first ensure that all backups and restores have completed. Next, stop the OSSV processes on the primary by either using the `svconfigurator` tool or running the `svpmgr` command under the `bin` directory.

```
C:\Program Files\netapp\snapvault\bin> svpmgr shutdown
```

Verify that `svlistener` is not running by utilizing task manager for Windows. This process was described in an earlier section.

Use the typical Add/Remove Programs tool included with Windows to remove OSSV.

8) Troubleshooting

OSSVINFO

If/when opening a case with technical support, a common set of files and output is required. In addition, the executable `OSSVINFO.exe or OSSVINFO.sh,` located on the internal engineering pages, is available. This executable will automatically execute the following commands and obtain the appropriate log files. `OSSVINFO.exe` is supported in Windows 2000, Windows 2003 and `OSSVINFO.sh` is supported on Solaris, Linux, IRIX, HP-UX, and AIX systems that have the OSSV agent installed.

`OSSVINFO` syntax is as follows:

Windows

```
OSSVINFO.exe [ –s secondary ] [ -l username:password ] outfile.txt
```

UNIX

```
OSSVINFO.sh [-s filer [-l user[:passwd]]]
```

The output is saved to a file in the form `ossvinfo-%Y%m%d%k%M%S.log`

This executable can be run from a Windows command shell or UNIX command prompt. When reporting problems seen in Data ONTAP, always use the -s option to collect information specific to Data ONTAP.

OSSVINFO can be used to gather OSSV and system information at customer sites when reporting problems. OSSVINFO currently collects the following data.

1. REGISTRY
2. HOSTNAME
3. HW_INFO
4. OS_VERSION
5. MEMORY
6. VOLUME
7. NET_USE
8. SERVICE_INFO
9. SNAPVAULT_STATUS
10. SNAPVAULT_DESTINATIONS
11. SNAPVAULT_STATUS_L
12. SVINSTALLCHECK
13. RELEASEDEF
14. SNAPVAULT
15. SNAPVAULT_CFG
16. CONFIGURE_CFG
17. PATH_EXCLUDE
18. FILE_EXCLUDE
19. SECONDARY_HOSTNAME
20. SECONDARY_VERSION
21. SECONDARY_SNAPMIRROR
22. SECONDARY_SNAPMIRROR_0
23. SECONDARY_SNAPVAULT_STATUS_L
24. SECONDARY_SNAPVAULT_STATUS_C
25. SECONDARY_QTREE_STATUS
26. SECONDARY_SNAPVAULT_SNAP_SCHED
27. SECONDARY_SNAP_LIST
28. SECONDARY_CIFS_SHARES

29. SECONDARY_SYSCONFIG

30. DATE

Manually Obtaining Data

It is possible to manually obtain data available from the secondary system. The following commands need to be executed prior to opening a case (place all data in the new case before submitting the case):

- `options snapvault`
- `snapvault status`
- `snapvault status -l`
- `snapvault status -c`
- `snapvault status -s`
- `snapvault snap sched`
- `priv set advanced ; snap status`
- `qtree status`
- `snap list`
- `snap list -q`
- `version`
- `license`

Secondary System Logs

Obtain all the relevant log files from both the OSSV secondary systems.

- `/etc/log/snapmirror`
- `/etc/messages`

Primary System Logs and Data

Obtain all the relevant log files and system information from the OSSV primary systems.

- `$INSTALL_DIR\etc\snapvault` or `$INSTALL_DIR/etc/snapvault`
- Properties of the file system/drive being backed up
- Properties of the file system/drive containing the OSSV agent installation

  On Windows:
  "`My Computer`" -> right-click drive icon -> "`Properties`" -> “`General`”.

On UNIX:
`df -k <directory>`

- Output from /etc/{v}fstab
- Error messages seen on console, if any

Generating Debug Information

While troubleshooting OSSV cases, the user may be asked to generate debug information pertaining to specific OSSV processes running on primary systems. The following steps would be executed to perform this.

Setup Debug Information

Prior to performing the following procedure, ensure no updates or transfers are occurring or will occur while debug is enabled.

1. Locate and open the `svconfigurator (Start → Run` OR command line`)`.
2. Go to the `General` tab.
3. Select the box marked `Generate debugging files`.
4. Go to the `Trace Level` tab.
5. Using the pull-down list, set `SnapVault Listener` to `VERBOSE`.
6. Select `Apply`.
7. Go to the `Service` tab.
8. Select `Stop Service`.
9. Wait for `Current State` to display `Stopped`.
10. Go to the `$INSTALL_DIR\trace` directory (if present). If there are any files here, delete all of them; these are old debug files.
    NOTE: Deleting old files will free space; these debug files can grow large and grow quickly.
11. Select `Start Service`.
12. Wait for `Current State` to display `Running`.
13. Dismiss the `svconfigurator` GUI by selecting `OK`.

Collect Debug Files

1. Open up a command prompt (Windows) or a shell (UNIX).
2. Change directory to the `$INSTALL_DIR\bin` directory.
3. Perform the command `snapvault status` to generate the necessary debug files.
4. Open `svconfigurator`.
5. Go to the `Service` tab.
6. Select `Stop Service`.
7. Wait for `Current State` to display `Stopped`.

Inspect Debug Files

First, change the current working directory to the `$INSTALL_DIR\trace` directory.

With a text editor, (e.g., WordPad, `vi`), open the latest debug file named `svlistener`*`xxxxxxxxxx`*`.log`, where *`xxxxxxxxxx`* is a set of numbers pertaining to the date and `svlistener` process number. The output will be similar to the following (output will vary depending on process chosen for debug output):

```
0 TRACE   :2336    1   0 165726 ** SnapVault Release: 'DEVELOPMENT' Built 'Wed Dec  7 20:09:10 GMTST 2005' **
0 TRACE   :2336    2   0 165726 ** Built on 'CYGWIN_NT-5.0 bubbles 1.5.5(0.94/3/2) 2003-09-20 16:31 i686
unknown unknown Cygwin' **
0 TRACE   :2336    3   0 165726 ** Running on host 'i386-rtp01', Time now '11:57:26' on 'Tue 14 Mar 2006' **
0 TRACE   :2336    4   0 165726 ** Host OS 'Windows 2003' version '5.2' on 'i586' hardware **
1 MACHINFO:2336   20   0 165726 NVBuildlevel = '0'
0 TRACE   :2336    5   0 165726 ** Server 'TRUE' SVVersion 0 SVBuildLevel 0 Description '(null)' **
0 TRACE   :2336    6   0 165726 ** SV Process Id '50' SV System Name 'I386-RTP01'
0 TRACE   :2336    7   0 165726 ** sizeof(long) = '4' **
0 TRACE   :2336    9   0 165726 **
0 PROCESS :2336  202   0 165726 ProcInitializeTags(50, TRUE, FALSE, FALSE, FALSE, FALSE,
svlistener0603141157.log, 3)
4 CFGUTIL :2336   34   0 165726 CfgUtilDoRemote, configfile: configure.cfg, alt path : (null).
4 CFGUTIL :2336   31   0 165726 CfgUtilGetNamesFromProcContext
4 CFGUTIL :2336   36   0 165726 CfgUtilDoRemote returns FALSE, with client (null) and server (null)
3 PATH    :2336   10   0 165726 PathHomeBuild returning "C:/Program
Files/netapp/snapvault/config/configure.cfg"
4 CFGLOCAL:2336   55   0 165726 CfgLocalGetKey
4 FILE    :2336   15   0 165726 FileStreamOpen(006278A8, C:/Program
Files/netapp/snapvault/config/configure.cfg, 0x101)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(00628F48, C:/Program
Files/netapp/snapvault/config/configure.cfg.lck, 6)
4 FILE    :2336    9   0 165726 TRUE = FileLockSet(000000BC, 0x7b)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(0012F9CC, C:/Program
Files/netapp/snapvault/config/configure.cfg, 101)
4 FILE    :2336   14   0 165726 TRUE = FileStreamClose(0x6278a8)
4 PROCESS :2336    3   0 165726 692 = ProcGetRealId()
4 PROCESS :2336  204   0 165726 SigKillEvent 000000BC
4 CFGUTIL :2336   34   0 165726 CfgUtilDoRemote, configfile: configure.cfg, alt path : (null).
4 CFGUTIL :2336   31   0 165726 CfgUtilGetNamesFromProcContext
4 CFGUTIL :2336   36   0 165726 CfgUtilDoRemote returns FALSE, with client (null) and server (null)
3 PATH    :2336   10   0 165726 PathHomeBuild returning "C:/Program
Files/netapp/snapvault/config/configure.cfg"
4 CFGLOCAL:2336   55   0 165726 CfgLocalGetKey
4 FILE    :2336   15   0 165726 FileStreamOpen(006278A8, C:/Program
Files/netapp/snapvault/config/configure.cfg, 0x101)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(00628F48, C:/Program
Files/netapp/snapvault/config/configure.cfg.lck, 6)
4 FILE    :2336    9   0 165726 TRUE = FileLockSet(000000B8, 0x7b)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(0012F8C4, C:/Program
Files/netapp/snapvault/config/configure.cfg, 101)
4 FILE    :2336   14   0 165726 TRUE = FileStreamClose(0x6278a8)
3 CFGLOCAL:2336    9   0 165726 Trying to load key 'TcpPort' from stanza 'Machine'
4 FILE    :2336   15   0 165726 FileStreamOpen(006278A8, C:/Program
Files/netapp/snapvault/config/configure.cfg, 0x101)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(00628F48, C:/Program
Files/netapp/snapvault/config/configure.cfg.lck, 6)
4 FILE    :2336    9   0 165726 TRUE = FileLockSet(000000B8, 0x7b)
4 FILESYS :2336   32   0 165726 TRUE = FileSysOpen(0012F8C4, C:/Program
Files/netapp/snapvault/config/configure.cfg, 101)

4 FILE    :2336   14   0 165726 TRUE = FileStreamClose(0x6278a8)
```

Delete Debug Files and Disable Debug

Delete debug files as soon as enough information has been collected. Turn debug off by reversing the above procedure. This is important, since the default location for the trace directory is under the default `$INSTALL_DIR`. A file system can quickly fill up if debug is left `on`.

9) Relevant Documentation

Reading the following documents is highly recommended in order to gain a better understanding of OSSV and SnapVault.

- Tech reports
  - TR-3234: Leveraging Network Appliance SnapVault for Heterogeneous Environments
  - TR-3240: SnapVault Deployment and Configuration
  - TR-3252: Enhancing Heterogeneous Backup Environments with SnapVault
- Manuals
  - Data ONTAP 6.4/6.5/7.x Online Backup and Recovery Guides
  - DataFabric Manager 3.x Administration Guide
  - OSSV 2.x release notes, available on NOW
  - OSSV 2.2 Installation and Administration Guide

APPENDICES

Appendix A: Logical Replication (LREP) for Seeding Baselines

LREP Demo

This example customer has a secondary system named `r200` in its data center based in Raleigh, North Carolina. `vol1` will be the secondary volume. There is a small Windows server named `nt1` in the company's Smithfield, North Carolina, office. A second Windows machine named `nt2` lives at the data center and will function as the lrep *writer*. A Zip drive, which will be drive letter `E,` will be moved between `nt1` and `nt2`.

At Remote Office

First, install the OSSV client and `lrep_reader` on the remote server, nt1. Navigate to the directory that contains the `lrep` executable and enter the following command:

```
C:\>lrep_reader -p snapvault_start -f r200 -q /vol/vol1/backup -o E:\test@0
nt1:D:\dataset
```

Examining one argument at a time:

`-p snapvault_start` = use SnapVault protocol

`-O = disable OSSV`

`-f  r200 = r200 is the final destination`

`-q /vol/vol1/backup = the full path on the FINAL destination`

```
-o E:\test@0  = the portable drive, a name for the file that will be created,
@number of 2GB files(0=infinite) * number of 2GB files created. This feature
can allow you to span multiple drives.

nt1:D:\dataset = the source you want to mirror
```

If your portable drive is small, say, 8GB, and your data is 12GB, and you have the option of connecting two portable drives at E:\ and F:\, then you could use the following:

```
-o E:\test@4 -o F:\test@0
```

NOTE: `E:\test@4` means create a maximum of four 2 GB files. So it will direct `lrep_reader` to store first 8GB in `E:\.`
NOTE: `F:\test@0` (0 means unlimited here) means create all files until the end of the stream in F:\.

To write to the final destination: `lrep_writer -p snapvault_start E:\test F:\test.`

At Data Center

Now move the Zip disk to the data center and `lrep_writer` host, `nt2,` and start the `lrep_writer`:

```
C:\lrep_writer -p snapvault_start -O E:\test
```

NOTE: OSSV cannot be installed on the `lrep_writer` machine due to contention for TCP port 10566.

Now start the transfer from `r200` (secondary system):

```
r200> snapvault start -S nt2:E:\test /vol/vol1/backup
```

Now modify the relationship on the secondary to reflect the true relationship settings:

```
r200> snapvault modify -S nt1:D:\dataset /vol/vol1/backup
```

Now you can simply manually force an incremental backup of the volume (for testing):

```
r200> snapvault update /vol/vol1/backup
```

Check status:

```
r200> snapvault status
```

Appendix B: Modifying Data of an OSSV Destination

Since the Open Systems SnapVault destination is a read-only, there are two methods that can be used to modify the data on the secondary system. The first method is to utilize the SnapVault/SnapMirror bundle, making the OSSV destination a SnapMirror destination. This method will allow the data to reside in the existing qtree and no extra volumes will need to be created. The second method is to utilize the FlexClone™ technology.

Using the SnapMirror/SnapVault Bundle

SnapVault does not currently have the ability to create a writable destination on the secondary system. However, you can use SnapMirror to convert the Open Systems SnapVault destination to a SnapMirror destination, making it a typical SnapMirror destination that can be quiesced and broken. In order to utilize this bundle, you must be running a version of Data ONTAP that supports the resync after break or restore feature first introduced in OSSV 2.2. In addition, the secondary system will also need to have a SnapMirror license enabled.

Converting and Making Secondary Read/Write

The following steps need to be performed to convert an OSSV or SnapVault secondary backup destination to a usable/writable destination (typically for DR situations):

1. Secondary: Turn SnapMirror and SnapVault off.
2. Secondary: Switch to privileged mode (`priv set diag`).
3. Secondary: Convert SnapVault qtree to SnapMirror qtree (`snapmirror convert <sec_qtree_path>`).
4. Secondary: Turn SnapMirror on.
5. Secondary: Quiesce the qtree.
6. Secondary: Break the mirror, making it writable.
7. Secondary: Turn SnapVault on.

Reestablishing the Relationship

With OSSV, there currently isn't a mechanism to propagate any changes made while the qtree is in a Read/Write state. If it is determined the same destination qtree needs to be used, then all changes to the qtree will be lost. The only other option is to leave the qtree as is and perform a new baseline of the source volume.

1. Secondary: Resync the secondary qtree

```
snapvault start -r -S <pri_system>:<pri_path> <sec_qtree_path>
```

NOTE: Resync with OSSV 2.2 (`snapvault start -r` command) isn't yet supported. Please see KB12138 for the latest versions of ONTAP that are supported.

Using FlexClones

FlexClone volumes are a point-in-time, writable copy of the parent volume (OSSV destination). Changes made to the parent volume after the FlexClone volume is created are not reflected in the FlexClone Volume. In order to create a FlexClone volume, the OSSV destination must be a flexible volume and FlexClone license installed and the SnapMirror license to convert the destination to a writable destination.

Creating a FlexClone Volume

The following step needs to be performed to create a FlexClone:

1. Secondary: `vol clone create clone_vol -b parent_volume [parent_snap]`

If no parent snapshot is named, a new base snapshot will be created by Data ONTAP.

NOTE: The base snapshot can not be deleted as long as a clone of the parent volume exists.

Once the FlexClone has been created, you can use CIFS/NFS to mount the FlexClone to the host with the data in a Read/Writable format.

NOTE: Any changes made to the FlexClone volume can NOT be propagated back to the OSSV Primary. The only way to get the changes made in the FlexClone volume is to manually copy the files that were created or modified.

NOTE: The FlexClone volume will be a read-only volume, SnapMirror must still be used in order to make these writable. Using this method will eliminate the need to perform a resync on the OSSV relationship.

Revision History

Date	Name	Description
08/03/2006	Jeremy Merrill	FlexClone
05/23/2006	Jeremy Merrill	Appendix B
05/04/2006	Jeremy Merrill	Update
09/26/2004	Darrin Chapman	Creation