



NxVTape



Virtual Tape Backup Server Installation Guide And Operator's Manual

**Version 1.0
July 18, 2006**

Table of Contents

Table of Contents	ii
1 Overview	4
1.1 Topological Diagram	5
2 Installation	6
2.1 Requirements	6
2.1.1 Hardware	6
2.1.2 Software	6
2.2 Installing the NxVTape Host Component	6
2.3 Installing VTapeService	7
2.3.1 Installing the Tandem NxVTSERV Component (Optional)	7
2.3.2 Installing the PC VTapeService Component (Optional)	7
3 Using NxVTape	9
3.1 Starting the Virtual Tape Service	9
3.1.1 NxVTSERV Tandem Service	9
3.1.2 VTapeService PC Service	9
3.2 Starting a Job from a TACL Prompt	10
3.2.1 Performing Backups	10
3.2.2 Performing Restores	11
4 Settings and Configuration	12
4.1 Diagnostics	12
4.1.1 Diagnostic Parameters	12
4.2 Logging	13
4.2.1 Logging Parameters	13
4.3 Service Parameters	14
4.3.1 NxVTape Parameters	14
4.3.2 NxVTserv Parameters	14
4.4 Licensing Parameters	14
4.5 A Sample GO File	15
5 Glossary	17



Nexsion, NxLIB, and NxVTape are trademarks of KLD, Inc.

Microsoft, and Microsoft Windows are registered trademarks of Microsoft, Inc.

ACI, BASE24, BASE24-atm, BASE24-pos, BASE24-teller, and NET24-XPNET are trademarks or registered trademarks of ACI Worldwide Inc., Transaction Systems Architects, Inc., or their subsidiaries.

1 Overview

NxVTape is a software tape backup server that can serve the same function as existing physical tape backup systems while simultaneously eliminating the need for a physical presence at the data site. It accomplishes this through the use of two components, the NxVTape host component (a Tandem process) and a PC component which runs as a Microsoft Windows ® component on the server slated as the backup repository.

- Uses same host software and procedures as existing tape backup schemes, minus the tape.
- Backups may be scoped to individual files, subvolumes, or entire drives.
- Fully multithreaded implementation on both host and PC permits many simultaneous backups and restores between multiple hosts and backup server PCs.
- Server comes in two variants: Windows and Guardian.
- Large TCP/IP block size (up to 52kb/block) permits very quick transfers.
- Simple, robust, no-nonsense TACL prompt operation ensures reliability – very little GUI interaction required.
- Easy-to-read logs.
- Easy installation.
- Conserves resources by providing the ability to backup encrypted data to a remote site in just minutes, compared to hours or days to send physical tapes via truck.
- Fastest and most inexpensive software-only Tandem backup system on the market.
- Optional Huffman data compression reduces virtual tape file size dramatically – 1/3 original size or better in some cases.
- Optional DES/3DES encryption protects privacy with an established and industry standard cryptographic algorithm.

1.1 Topological Diagram

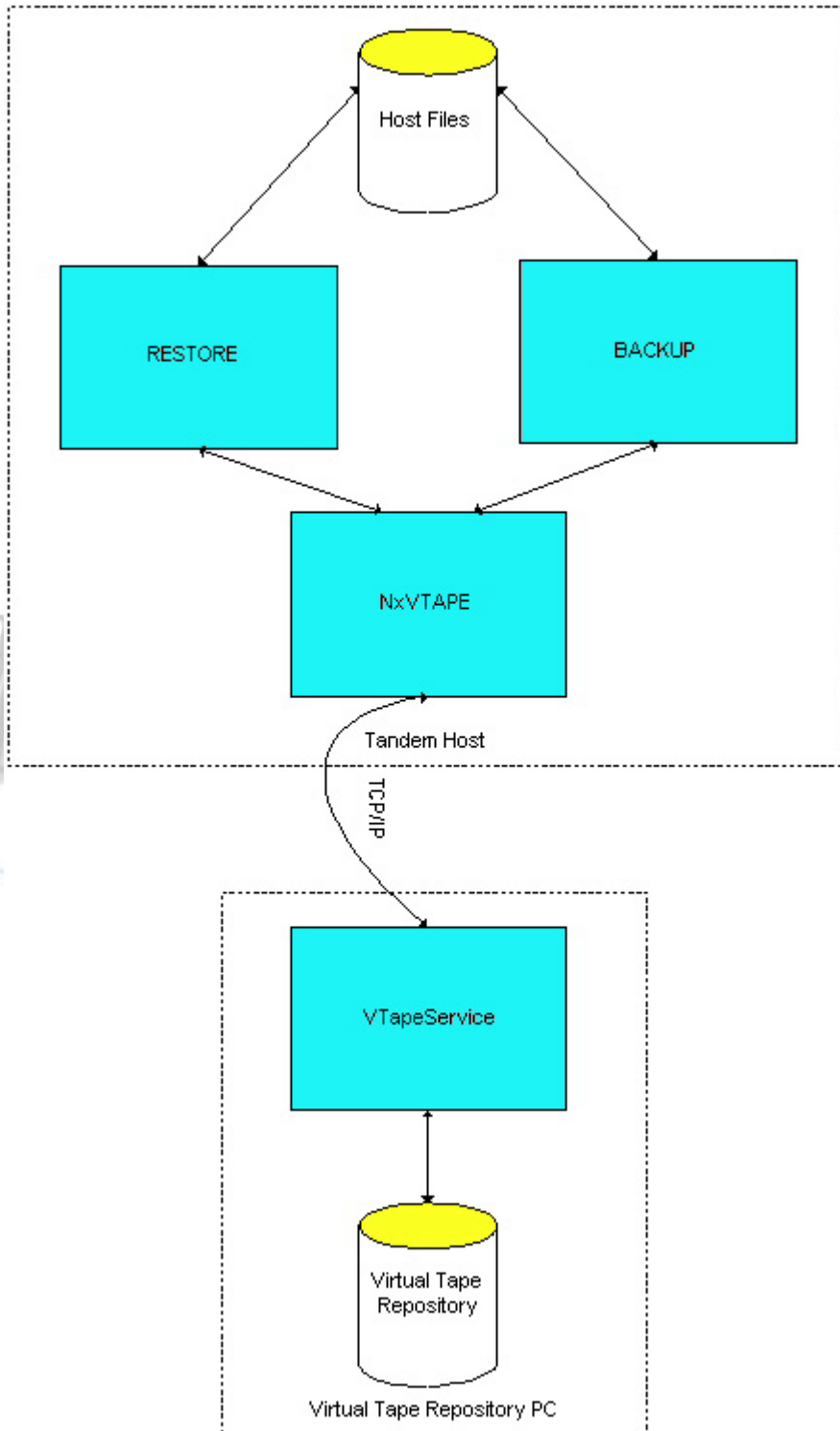


Figure 1 - This diagram shows the processes and data path of a typical backup arrangement.

Note that the physical block labeled Virtual Tape Repository PC may also be a Tandem running the NxVTServ process. TCP/IP allows a degree of system agnosticism that allows each organization to deploy in a manner that best serves its particular needs.

2 Installation

The following section will walk the IS administrator through requirements, installation, and deployment planning.

Broadly, NxVTape installation components can be categorized thus:

1. NxVTape host executable
2. Host configuration/go file
3. VTapeService.exe Windows service
4. PC Configuration file

2.1 Requirements

2.1.1 Hardware

Host

- Any K- or S- series Tandem/HP Nonstop Server system may run NxVTape.
- Any K- or S- series Tandem may likewise run NxVTServ, the Tandem version of the virtual tape server.

PC

- Any networked machine with ample drive space for backups and capable of running Microsoft Windows 2000, Windows 2003 Server, or Windows XP Professional operating system.

2.1.2 Software

Host

- Guardian D48 or newer

PC

- Microsoft Windows 2000, Windows 2003 Server, or Windows XP Professional operating system.

2.2 Installing the NxVTape Host Component

This subsection offers a step-by-step guide for configuring NxVTape to run. [Section 3](#) offers a more comprehensive delineation of the various runtime parameters.

The NxVTape software is delivered as a standard PAK file. The PAK file should be transferred to the HP/NonStop system using a standard file transfer utility such as FTP or IXF. Within the client transfer utility, the file type to use is “Binary”.

1. Once the file has been transferred, it must be unpacked into its final destination. For example, to unpack to \$data1.nxvtape, issue the below command from the TACL prompt:

```
$DATA1.TEMP> unpack NxVTape, *.*.*, vol $data1.nxvtape
```
2. Modify the included GO file.
 - a. Point the BACKUP-SERVER param to point to the file server and port. This will be either to a NxVTServ Tandem process, for Tandem-to-Tandem backups, or to a PC running VTapeService.
 - b. Specify log-file and log output destination. NxLIB allows for unique routing of each severity level of log message (see [Logging](#)).

- c. If the optional encryption support was purchased, specify one, two, or three DES keys to use for encryption/un-encryption of files. A single key will provide DES encryption; two or three will offer stronger Triple-DES (3DES) protection.
- d. Give the process a system-unique name in the `run` command at the end of the file, as in this example:


```
run nxsrv /name $ETAP, nowait, cpu 0/
```

 - i. Specify a CPU for which the product was licensed (e.g. 0/1, 2/3) as the process' CPU.
(The process name \$ETAP is purely informational – in this case, it might be the tape process for encrypted backups for instance.)

3. Obey the go file to start the NxVTape process.

The process should be running at this point. In your INFO and CONFIG logs (see [Logging](#) section for more information on configuring the logs), you should see something similar to the following:

```
07/11 11:14:49.97 $ETAPE 9307 NXVTAPE CONFIG NxVTAPE Version 1.2
07/11 11:14:49.98 $ETAPE 9308 NXVTAPE CONFIG Copyright (c) 2006 by KLD, Inc.
07/11 11:14:49.98 $ETAPE 9309 NXVTAPE CONFIG WWW.NEXSION.COM
```

2.3 Installing VTapeService

VTapeService comes in two flavors, Windows and Tandem. This section describes the installation procedure for each. Each component is optional, but one or the other (or both) must be present to serve access to the virtual tape repository.

2.3.1 Installing the Tandem NxVTSERV Component (Optional)

The Tandem tape server will be unpacked along with the [NxVTape Host Component](#).

Once it is installed, follow these steps:

1. To configure it, edit the GOVTSERV file.
 - a. Specify BIND-SOCKET parameter as the address to accept (leave as "*" to accept all address) and the port on which to listen.
 - b. Specify the VOLUME parameter. This is the drive to be used for the virtual tape files. The subvolume is determined by a command line parameter to BACKUP (see the section entitled [Starting a Job from a TACL Prompt](#)).
 - c. Set up the parameters already described in [Section 2.2](#).
2. Obey the GO file to start the process.

2.3.2 Installing the PC VTapeService Component (Optional)

To install the PC variant of the server, VTapeService, simply run VTapeInstall.exe.

The installer creates the appropriate directories, generates the configuration file, and extracts the VTapeService executable files to the appropriate install folder.

A brief description of the settings follows:

- Program location – path where the executable program, support files, and configuration file will be stored.

- Repository location – path where the log file and all backups will be kept. This is referred to as a repository.
- Service Port – the TCP/IP port to use for inbound connections from the hosts performing backups.
- Automatically start – Specifies whether the service starts on user login.

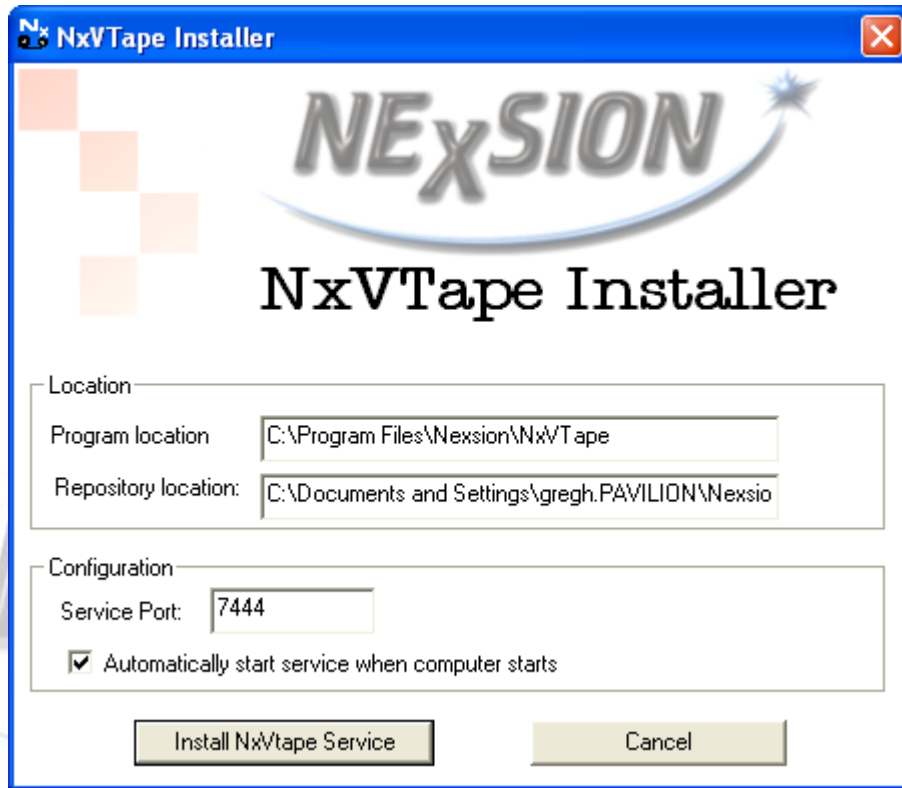


Figure 2 – NxVTape_Install program

3 Using NxVTape

This section describes the usage of NxVTape once all of its parts are installed.

3.1 Starting the Virtual Tape Service

3.1.1 NxVTSERV Tandem Service

Once all of the installation procedures have been followed for the optional NxVTSERV, it only needs to be run; obey the GO file. See [Section 2.3.1](#) for details.

3.1.2 VTapeService PC Service

All operations are initiated from the host. The optional PC component, VTapeService, plays the role of server. Hence, all UI on the PC side is purely informational, and consists of a status screen (see Figure 3) and a log file in the repository directory.

VTapeService is an executable which, upon install, will start automatically each time the user logs into Windows.

Once running, the VTapeService icon will appear in the system tray: 

Right-clicking this icon will display two options. “Properties” permits a viewing of the present state of VTapeService, including number of backup jobs active, number of restore jobs active, and the cumulative number of bytes transferred for each. “Exit” will terminate the process and any currently active jobs.

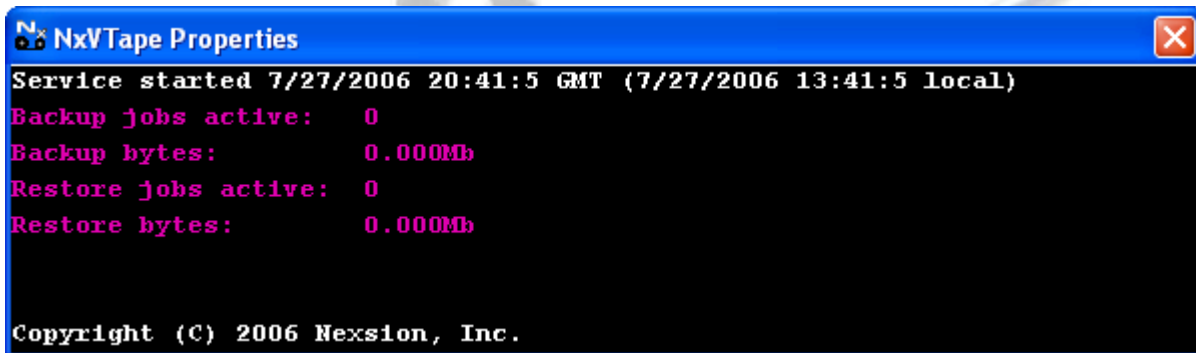


Figure 3 - VTapeService properties

The information in properties conveys a snapshot of the present status of VTapeService. More detailed information can be found in the log, located in the repository directory.

The Backup Bytes (Restore Bytes) count up as bytes are received. Note that these are TCP/IP payload bytes, not just file data. This is especially true if compression is used. Therefore, the # of bytes reported by Backup Bytes may differ slightly from that in Restore Bytes due to differences in the headers for these respective operations.

Up to 64 jobs may be active at any one time. Naturally, performance will be affected by CPU and network throughput limitations.

3.2 Starting a Job from a TACL Prompt

Following is command-line syntax for performing backups and restores. Please note that Hewlett Packard provides the BACKUP and RESTORE tools standard on all Tandem Systems and are therefore mentioned here only insofar as they provide the entry point for NxVTape functionality. We will only touch on details salient to NxVTape.

The reader is referred to the Guardian documentation on BACKUP and RESTORE, available from Hewlett Packard, for a more thorough treatment of these powerful tools. This can be found in the [Guardian Disk and Tape Utilities Reference Manual](#).

3.2.1 Performing Backups

NxVTape masquerades as a tape drive process to the existing BACKUP and RESTORE utilities.

The precise syntax for Backup, as seen in *Guardian Disk and Tape Utilities Reference Manual -- 3. BACKUP*, is as follows:

```
BACKUP [\node.][$volume.][subvolume.]
[ / run-option [, run-option ] ... / ]
{ tape-device-name
{ ( tape-device-name1, tape-device-name2, ... ) }
, backup-files
[ , file-mode-backup-option ] ...
```

To backup a drive, you would use the BACKUP tool as follows:

```
BACKUP $ETAP.#MAINNAME.SUBNAME,$DRIVE.*.* ,listall,blocksize 52
```

If the NxVTape process \$ETAP is configured to communicate with VTapeService, this command will backup all the files on \$DRIVE to a PC file called MAINNAME . SUBNAME in the PC's repository directory.

If \$ETAP is talking to a Tandem NxVTSERV process it will backup to a file called e.SUBNAME in subvolume MAINNAME on the volume specified by the VOLUME runtime parameter for NxVTServ.

It is possible to back up only files matching a certain pattern – for example, source code files ending in “C” or “S” – or an entire system. For example, to backup a group of files – say, all files ending with “S” – use the following syntax:

```
BACKUP $ETAPE.#MAINNAME.SUBNAME,$DRIVE.*.*S,listall, blocksize 52
```

Obviously, as evidenced by the optional \NODE prefix, it is possible to backup from another system via an Expand network.

The listall parameter specifies progress output.

The final parameter, blocksize 52, specifies usage of 52-kilobyte blocks. The parameter is not mandatory but can help performance a bit for transfer of large files.

3.2.2 Performing Restores

As for BACKUP, the reader is referred to Hewlett Packard's documentation for RESTORE, found at *Guardian Disk and Tape Utilities Reference Manual -- 7. RESTORE* .

The essential syntax for RESTORE matches that of BACKUP, but lacks the `blocksize` parameter.

Virtual tape images taken from one system can indeed be restored to a separate system, making NxVTape a fast and viable tool for very high-volume file transfers.

Performance-wise, a restore of an individual file at the end of the tape file can take from a few seconds to a few minutes. For example, to restore a 2-megabyte file at the very end of a 396MB archive to a S72000 will take approximately 20 seconds, depending on network conditions, number of files in the archive, etc. Thus, accidental deletions can be rectified with relative ease compared to a conventional tape drive which may require several hours for the same operation



4 Settings and Configuration

This section offers a description of the various configurable parameters and their respective functions. Understanding these parameters and their function will give the operator a fairly solid grasp of the product and its operation. All runtime parameters are specified in the GO file and therefore must be specified prior to running a given NxSSLSRV process. Future releases may feature a real-time configuration and monitoring utility.

Taxonomically, the parameters are divided into the following major groups, discussed in the following subsections:

1. Diagnostic parameters
2. Logging parameters
3. Service parameters
4. Licensing parameters

Each of the parameters in the GO file is described following this convention:

Description
param *name* [*values*]

The [*values*] field shows literals in “double-quotes” or else a brief, plain-language description of the value.

4.1 Diagnostics

NxVTape is built on top of the powerful NxLIB library and its proprietary multitasker. This underlying library allows for a certain degree of high-level control over task-level monitoring and heap management. Under most circumstances, the default values in the GO file (excepting those noted in [Section 2.2](#)) shipped from Nexsion require no changes.

4.1.1 Diagnostic Parameters

Poolsize determines the maximum size the program's heap can grow, in megabytes.

param poolsize [integer, default 8]

Task-cputime determines if tasks track their CPU time (time spent in processor).

param task-cputime ["enable" | "disable", default "enable"]

The task-walltime switch determines if tasks track total elapsed time for each task's life.

param task-walltime ["enable" | "disable", default "enable"]

Determines whether to generate one-line task info on termination. This goes to log, INFO-level ([see Logging Parameters](#)).

param task-report ["enable" | "disable", default "enable"]

Task-stackcheck Turns on internal stack-probes (for diagnostics, high watermarks).

param task-stackcheck ["enable" | "disable", default "enable"]

Task-Verify determines whether to step through pool on each pool operation (memory allocation or free).

param task-verify ["enable" | "disable", default "disable"]

On exception, trap handler will allow task manager to remove thread and continue. Otherwise, the exception is handled by the operating system and the entire process will be killed on exception.

param trap handler ["enable" | "disable", default "enable"]

Looptimer applies a one-minute loop-timer, triggering trap if procedure loops for over a minute.

param looptimer ["enable" | "disable", default "enable"]

4.2 Logging

NxLIB's logging capabilities allow for tight granular control over both the level of detail in each log message, and the routing of messages to specific files as triaged by message severity. This allows operators to look at only the messages requiring specific operator actions in one file, or the EMS log, and if necessary review lower-severity messages that may aid in diagnosing specific problems.

4.2.1 Logging Parameters

Logger simply turns logging on or off.

param logger ["enable" | "disable", default "enable"]

Log-file specifies the default file to which to log. Any severity-class of log messages destined for "FILE" will end up in the file specified by this parameter.

param log-file [Guardian filename]

Log-app provides the name of the application, which will appear in log message headers.

param log-app [name of application]

Log-civiltime should be enabled for local system time, and disabled for GMT.

param log-civiltime ["enable" | "disable", default "enable"]

Log-hdr specifies the level of detail per log record.

0: Log message only

1: Log hh:mm:ss, process name, abbreviated severity, and message

2: Log mm/dd, hh:mm:ss.ss, process name, message code, log-app name, expanded severity, and message.

3: Log yyyy/mm/dd, hh:mm:ss.sss, process name, message code, log-app name, expanded severity, and message.

param log-hdr [0-3]

All of the following parameters allow for triaging of log messages to logs by severity. Each message severity may go to any log, or to multiple, comma-delimited logs. All parameters are in double-quotes in the GO file.

param log-level-abend ["FILE[, EMS[, PAGE[, DEBUG]]]]"

param log-level-stop ["FILE[, EMS[, PAGE[, DEBUG]]]]"

param log-level-critical ["FILE[, EMS[, PAGE[, DEBUG]]]]"

```
param log-level-alert      [ "\"FILE[ , EMS[ , PAGE[ , DEBUG]]\" " ]
param log-level-warning   [ "\"FILE[ , EMS[ , PAGE[ , DEBUG]]\" " ]
param log-level-info      [ "\"FILE[ , EMS[ , PAGE[ , DEBUG]]\" " ]
param log-level-config    [ "\"FILE[ , EMS[ , PAGE[ , DEBUG]]\" " ]
param log-level-dev       [ "\"FILE[ , EMS[ , PAGE[ , DEBUG]]\" " ]
```

4.3 Service Parameters

This class of parameters determines the runtime behavior of the NxVTape and optional NxVTServ host processes.

4.3.1 NxVTape Parameters

tape-server – IP address/port pair for target tape repository. This will be the PC running VTapeService OR the Tandem system running NxVTServ.

```
param tape-server      n[n[n]] . n[n[n]] . n[n[n]] . n[n[n]] : n[n[n[n[n]]]]
```

des-key-*n* – The DES keys, where *n* is “1”, “2”, or “3”.

```
param des-key-1      [ 16-character hexadecimal number ]
param des-key-2      [ 16-character hexadecimal number ]
param des-key-3      [ 16-character hexadecimal number ]
```

4.3.2 NxVTServ Parameters

bind-socket – address portion is an asterisk to accept connections from all, followed by the port on which to listen:

```
param bind-socket    * | [ n[n[n]] . n[n[n]] . n[n[n]] . n[n[n]] ] : n[n[n[n[n]]]]
```

volume – the drive to which to write virtual tape images.

4.4 Licensing Parameters

There is but one licensing parameter. This is the 32-bit license key, obtained from Nexsion, which allows the product to run.

```
param license        [ 32-character hexadecimal number ]
```

Within this key are embedded feature availability flags. In the case of NxVTape, this includes compression, and encryption – depending on organizational needs.

4.5 A Sample GO File

The following is a sample obey GO file to start the NxVTape process. Most of the parameter apply to all Nexsion GO files.

```
== Stop any currently active process with the same name - presumably NxVTAPE.
stop $etap
```

```
== Clear the runtime params
clear all
```

```
=====
== DIAGNOSTIC PARAMS
=====
```

```
== Maximum size program's heap can grow, in MB.
param poolsize          8
```

```
== Switch determines if tasks track their CPU time (time spent in processor).
param task-cputime      enable
```

```
== Switch determines if tasks track total elapsed time for each task's life.
param task-walltime    enable
```

```
== Switch determines whether to generate 1-line task info on termination.
== Goes to log (INFO-level)
param task-report      enable
```

```
== Turns on internal stack-probes (for diagnostics, high watermarks).
param task-stackcheck  enable
```

```
== Determines whether to step through pool on each pool operation (memory
== allocation or free). For diagnostics.
param task-verify      disable
```

```
== On exception, trap handler will allow task manager to remove thread and
== continue. Otherwise, the exception is handled by the operating system
== and process will be killed on exception.
param trap handler     enable
```

```
== Applies 1-minute loop-timer, triggering trap if procedure loops for
== over a minute.
param looptimer        enable
```

```
=====
== LOG PARAMS
=====
```

```
== The log params allow configuration of data logged by the NxSSLSrv app.
== log-levels are broken down by severity, from abend (most serious event)
== to dev (developer debugging, most verbose).
```

```
== The logs themselves are broken into FILE, EMS, PAGE, and DEBUG
== Configuration allows:
```

```
== - Into which log each message will go by severity.
param logger           enable
param log-file         $ZTN1.#PTR6ZB1
```

```
param log-app                NxSRV

== log-civiltime - enable for local system time; disable for GMT
param log-civiltime         enable

== log-hdr - level of detail (0-3)
param log-hdr               2

== This allows triaging of log messages to logs by severity
param log-level-abend       "FILE, EMS, PAGE, DEBUG"
param log-level-stop        "FILE, EMS"
param log-level-critical    "FILE, EMS"
param log-level-alert       "FILE, EMS"
param log-level-warning     "FILE, EMS"
param log-level-info        "FILE"
param log-level-config      "FILE"
param log-level-dev         "FILE"

=====
== NxVTAPE PARAMS
=====

== tape-server points to the TCP/IP address and port of the virtual tape
== service. This may be either NxVTSERV on the Tandem, or VTapeService on a
== Windows box.
param tape-server           150.150.150.150:7444

== Compression - Turns compression on or off
param compression          on

== DES Keys - these are the keys for the optional encryption capability.
== Supply one for regular DES, or two or three for Triple-DES.
== IMPORTANT: KEEP TRACK OF THESE; YOU WILL NEED THEM TO RESTORE TO A
== DIFFERENT SYSTEM. THE KEYS FOR A RESTORE MUST MATCH THOSE USED FOR THE
== CORRESPONDING BACKUP!
param des-key-3             0123456789abcdef
param des-key-1             6543210fedcba987

== License Key contains all the information necessary for licensing the
== product for legal use, and also contains product-specific information.
param license               2D05DA67AFB8A37DB266C120482F70EE

== Run the program
run NxVTAPE /name $etap, nowait, cpu 0/
```

5 Getting the Most from NxVTape

5.1 Backup Performance

NxVTape can be run as multiple process instances on multiple CPU's to improve throughput. Because of this, it is possible to improve performance by running multiple jobs in parallel. This gives NxVTape a significant advantage over other solutions, especially traditional tape.

For example, it is possible to back up four roughly equivalent production drives – containing, say, 16-24Gb of data each – using four CPU-differentiated instances of NxVTape all connected to a common repository machine running VTapeService but going through different control ports. Since the latter is primarily bounded by network limitations, if its Ethernet interface is powerful enough the backup should require about the same amount of time required to back up just one of the four drives.

Generally, the faster the system, the more compression will improve performance. The bounding factor then becomes network bandwidth rather than CPU. At that point, performance is benefited through the process parallelization described above.

5.2 Restore Performance

In most use cases, backups are performed on a wide swath of the system at once, while restores are used less frequently and on a very narrow subset of the archive. A common scenario is an accidentally-deleted file. Anyone who has experienced this is well aware of the time and hassle required to load the tape, invoke the command, and wait perhaps many hours as the tape spools by, the drive scanning forward to the needed file.

For this reason, VTapeService's restore function has been optimized to scan quickly without transferring the full payload of the archive over the network. An index scheme is not needed; that would almost certainly compromise security for encrypted files by moving key-derivations off the host.

Restoring the last file to a S72000 from 380MB compressed (700MB uncompressed) archive requires about 20 seconds using a 10-Megabit Ethernet controller. Faster response could be expected with faster hardware.

5.3 Labeling/Naming Conventions

Virtual tapes can be named/labeled however one wishes within Guardian's limitations. It is recommended that all operators agree upon and follow a common naming convention for the virtual tapes, since each file must bear a unique name within the repository.

One suggestion is to use the first segment for the system name and (if backing up drive individually) drive, with the suffix segment containing encryption status, compression status, and date, as in the following example.

```
BACKUP $ETAPE.#DTRAPROD.EC080106,$PROD.*.*S,listall, blocksize 52
```

Such a scheme maintains system, drive, compression/encryption and date uniqueness.

6 Glossary

Authentication - The process of verifying identity so that one entity can be sure that another entity is who it claims to be.
ASN – Abstract Syntax Notation
BER - Basic Encoding Rules.
CA - Certification Authority.
CBC - Cyber Block Chaining, a protocol variant wherein a portion of the previously encrypted cipher text is used in the encryption of the current block
Cipher - An algorithm for encrypting and decrypting data.
CRL - Certificate Revocation List.
Cryptography – The study of using cryptological algorithms to build secure systems and protocols.
Cryptology – The theory of designing algorithms used to provide security.
CTS – Common Transport System. A open interface communication protocol by which XPNET / Base24 components communicate.
DER - Distinguished Encoding Rules.
DES - Data Encryption Standard.
DH - Diffie-Hellman, a secure key-exchange mechanism.
DN - Distinguished Name.
DSA - Digital Signature Algorithm.
GMT – Greenwich Mean Time. An absolute time-of-day used in many systems, tied to the local time for Greenwich, England.
Huffman – A lossless compression algorithm named after the late MIT professor David Huffman.
IPRA - Internet Policy Registration Authority .
OID - Object Identifier
OSI - Open Systems Interconnect. In 1978, the International Standards Organization developed this model as the first step in an attempt to standardize interconnection of diverse networks.
PCA - Policy Certification Authority. Obsolete under x509 v3.
Peer - The other machine in the conversation.
PEM - Privacy Enhanced Mail. PEM, published in 1993, includes specifications for a public key infrastructure based on X.509 v1 certificates.
PKI - Public Key Infrastructure.
Public key certificates - data structures that bind public key values to subjects.
RA- Registration Authority.
RSA - R. Rivest, A. Shamir, and L. M. Adleman.
SHA – Secure Hash Algorithm.
SPC - Software Publisher Certificate.
SSL - Secure Sockets Layer.
TLS - Transportation Layer Security.